GeoHazards

MDPI

Article

Time Series Foundation Models and Deep Learning Architectures for Earthquake Temporal and Spatial Nowcasting

Alireza Jafari [1], Geoffrey Fox [1,2,*], John B. Rundle [3], Andrea Donnellan [4] and Lisa Grant Ludwig [5]

[1] Computer Science Department, University of Virginia, Charlottesville, VA 22904, USA; jrp5td@virginia.edu
[2] Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA 22904, USA
[3] Physics and Astronomy and Geology, University of California, Davis, CA 95616, USA; jbrundle@ucdavis.edu
[4] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA; andrea.donnellan@jpl.nasa.gov
[5] Department of Population Health & Disease Prevention, University of California, Irvine, CA 92697, USA; lgrant@uci.edu
* Correspondence: vxj6mb@virginia.edu or gcfexchange@gmail.com

Abstract: Advancing the capabilities of earthquake nowcasting, the real-time forecasting of seismic activities, remains crucial for reducing casualties. This multifaceted challenge has recently gained attention within the deep learning domain, facilitated by the availability of extensive earthquake datasets. Despite significant advancements, the existing literature on earthquake nowcasting lacks comprehensive evaluations of pre-trained foundation models and modern deep learning architectures; each focuses on a different aspect of data, such as spatial relationships, temporal patterns, and multi-scale dependencies. This paper addresses the mentioned gap by analyzing different architectures and introducing two innovative approaches called Multi Foundation Quake and GNNCoder. We formulate earthquake nowcasting as a time series forecasting problem for the next 14 days within 0.1-degree spatial bins in Southern California. Earthquake time series are generated using the logarithm energy released by quakes, spanning 1986 to 2024. Our comprehensive evaluations demonstrate that our introduced models outperform other custom architectures by effectively capturing temporal-spatial relationships inherent in seismic data. The performance of existing foundation models varies significantly based on the pre-training datasets, emphasizing the need for careful dataset selection. However, we introduce a novel method, Multi Foundation Quake, that achieves the best overall performance by combining a bespoke pattern with Foundation model results handled as auxiliary streams.

Keywords: deep learning; earthquake nowcasting; foundation models; transformers; pre-trained models; graph neural networks; seismic data; Southern California

Citation: Jafari, A.; Fox, G.; Rundle, J.B.; Donnellan, A.; Ludwig, L.G. Time Series Foundation Models and Deep Learning Architectures for Earthquake Temporal and Spatial Nowcasting. *GeoHazards* **2024**, *5*, 1247–1274. https://doi.org/10.3390/geohazards5040059

Received: 12 October 2024
Revised: 12 November 2024
Accepted: 15 November 2024
Published: 21 November 2024

1. Introduction

Earthquake forecasting represents a dynamic and critical domain of research with profound implications for disaster risk reduction and public safety. The ability to accurately nowcast seismic events and mitigate their impacts is vital for preserving lives and minimizing damage. Forecasting earthquakes is inherently complex due to the absence of consistent and reliable indicators and the infrequent occurrence of major seismic events. This complexity is further compounded by multifaceted long-term and short-term interactions that can transfer between regions [1]. Earthquake nowcasting is a cutting-edge approach in seismology that focuses on real-time nowcasting of seismic activity to assess immediate risk [2]. By analyzing extensive historical and real-time seismic datasets, nowcasting identifies patterns that may signal an impending earthquake. Deep learning, with its robust capacity to process and derive insights from extensive datasets, offers a promising avenue for identifying patterns and potential predictive signals within these data. By leveraging advanced

algorithms and computational power, deep learning facilitates a deeper understanding of seismic phenomena, leading to more accurate and reliable earthquake forecasting [3].

Historically, the study of future earthquakes has relied heavily on statistical models, which operate by identifying patterns to forecast seismic events, as referenced in various studies [4,5]. However, these conventional methodologies frequently encounter limitations in capturing the complex temporal and spatial patterns inherent in seismic activities [6]. Such shortcomings manifest in challenges related to capturing subtle long-term patterns and understanding the interactions between different seismic activities and the diverse nature of seismic sources.

In recent years, deep learning techniques have emerged as a promising avenue to address these challenges by leveraging the power of neural networks and transformers to learn and extract patterns from vast amounts of seismic data [2,7,8]. A prevalent methodology for earthquake nowcasting involves the utilization of sophisticated neural network architectures, such as Convolutional Neural Networks (CNNs) or Long Short-Term Memory Networks (LSTMs) [2,9,10]. CNNs generally excel at capturing local patterns within features, while LSTMs are adept at modeling temporal dependencies and long-term trends.

Graph Neural Networks (GNNs) present a well-suited approach for earthquake forecasting due to their proven ability to capture and model complex relationships within data through graph structures. These models have demonstrated significant promise across various domains, such as stock markets [11,12] and large language models [13], characterized by intricate networked relationships. Surprisingly, despite their potential and success in several domains, the use of GNNs in earthquake forecasting remains relatively underexplored. Only a limited number of studies have adopted GNNs for this purpose, and the graph-based modeling of known effective components, such as fault lines, needs further investigation [14–18].

An emerging advancement in deep learning methodologies involves the application of transformer architectures. Transformer models, such as BERT [19] and GPT [20], initially popularized in the domain of natural language processing, have revolutionized how we approach problems by capturing long-range dependencies and understanding context [21]. However, despite their versatility and success in various applications, transformer models have not been commonly used for earthquake forecasting [22,23]. Given their ability to handle sequential data and capture complex temporal patterns, transformers have the potential to significantly enhance earthquake forecasting.

Similarly, pre-trained foundation models have shown powerful capabilities in transferring knowledge across different tasks and domains [24]. In particular, over the last three years, there have been over sixty projects and 200 references addressing time series foundation models [25]. Pre-trained foundation models have shown considerable promise in time series prediction, effectively capturing temporal dependencies and improving predictive accuracy [26]. However, despite their success and versatility, these models have not yet been applied to the specific challenge of earthquake forecasting.

In this study, we focus on earthquake time series, derived from the logarithmic scale of energy released by earthquakes within a 14-day window, across a 0.1-degree spatial bin in Southern California. We construct a time series for each spatial bin, aiming to accurately capture the spatial dependencies and patterns of seismic events. We develop time series foundation models and advanced deep learning architectures tailored to earthquake time series data. Crucially, we introduce novel models that address existing limitations in the literature, significantly advancing the state of earthquake nowcasting.

We adopt six large pre-trained foundation models for earthquake nowcasting: iTransformer [26], PatchTST [27], TimeGPT [28], Time-LLM [29], Chronos [30], and TSMixer [31]. The first five models—iTransforemer, PatchTST, TimeGPT, Time-LLM, and Chronos—are transformer-based and TSMixer is MLP-based. These models have demonstrated exceptional performance in various domains by effectively handling complex temporal dependencies [32]. TimeGPT is a generative time-series forecasting model leveraging GPT architecture, pre-trained on a large collection of time series with over 100 billion data points.

Chronos, on the other hand, has proven its ability to integrate temporal information by converting time series data into contextual insights, making it particularly useful for our purposes. We pre-train the remaining models on three different datasets to enhance the model's predictive capabilities.

We also modify memory-based models, including DilatedRNN [33], TFT [34], and LSTM, to evaluate their suitability for earthquake nowcasting. DilatedRNN has shown strong potential in modeling sequential dependencies over extended periods, while TFT excels in handling multiple time series with varied lengths. LSTM, a widely recognized model in time series forecasting, provides valuable insights into the temporal dynamics of seismic activities.

To explore the capabilities of convolutional layers, we employ models such as TimesNet [35] and TCN [36]. TimesNet effectively captures local temporal features, while TCN demonstrates robustness in handling long-range dependencies, making both models suitable for the intricacies of earthquake data. Additionally, we incorporate a powerful MLP-based model called TiDE [37], which showcases impressive performance due to its simplicity and efficiency in capturing non-linear relationships within the data.

We introduce a GNN architecture, called GNNCoder, for earthquake nowcasting that leverages geographical interactions to enhance model prediction accuracy. We create an earthquake graph using the epsilon nearest neighbor algorithm based on the proximity of spatial bins, identifying spatial clustering and patterns. Our model employs Graph Attention Networks (GATs) with an MLP-based encoder-decoder to focus on relevant connections within seismic data, offering a comprehensive framework for accurate and reliable earthquake forecasting.

Furthermore, we introduce the Multi Foundation Quake model, an innovative approach that aggregates multiple foundation models to enhance nowcasting performance. By leveraging the strengths of various pre-trained models, Multi Foundation Quake effectively captures both temporal and spatial dependencies, providing a more robust and accurate forecast of seismic activities. This model highlights the potential of combining diverse pre-trained models to improve earthquake nowcasting.

By advancing the state of the art in earthquake nowcasting, this research significantly contributes to both the deep learning and earthquake research domains. It highlights the critical aspects of earthquake information that must be considered in future forecasting studies. The improved accuracy and reliability of our models have the potential to enhance disaster response efforts, minimize economic losses, and save lives by providing timely and precise nowcasting of seismic events. This research is pivotal in bridging the gap between advanced deep-learning methodologies and practical applications in understanding the probability of earthquake occurrence and mitigation.

In the next section, we will talk about the current state-of-the-art. Section 3 explains the data used in this study, including the preprocessing steps and segmentation into spatial bins. The employed methodology and pre-training process are detailed in Section 4, covering model architectures and training procedures. Section 5 provides comprehensive information about our experimental setup, evaluation metrics, and baseline comparisons. Finally, we conclude the article in Section 6, summarizing key findings and suggesting future research directions.

2. Current State-of-the-Art

Unlike traditional forecasting methods, nowcasting allows for rapid updates and model predictions, which can significantly enhance preparedness and mitigate the impacts of earthquakes [38]. Furthermore, several physical principles that underlie earthquake time series nowcasting are discussed in the following references, but we do not delve into them here [1–3]. In a later study, we will report on studies of the ETAS-based earthquake simulator [39–41], which provides a powerful method to understand the role of different physical effects seen in earthquake time series.

Our approach in this work is similar to [2,42] whose nowcasting technique is shown in Figure 1 [43]. In this method, an exponential moving average (EMA) is applied to the time series of small earthquakes in Southern California, with a correction to account for the relatively poor detection of low-magnitude earthquakes in the time series for the pre-automated era prior to about 1995. Thus, a 2-parameter filter on the small earthquake time series was constructed and then optimized with machine learning, as shown in Figure 1. The optimization criterion was the receiver operating characteristic (ROC) skill, which is the area under the ROC curve. We show the value of these physics-motivated observables for foundation models later in this paper, in Section 5.2.4.

Building on this nowcasting method, we then developed a time-series transformer model (QuakeGPT) to predict future values of the time series beyond the observed test data, as shown in Figure 1. In this transformer model for time series, we use feature vectors consisting of 36 months of data to predict the next value. We identify the "keys" with 36 months of data in the training dataset, and the corresponding "values" as the value of the next data point. We identify the "queries" as the 36-month feature vectors in the validation dataset that are used to predict the subsequent value of the time series. By employing the transformer's attention mechanism, QuakeGPT effectively captures temporal dependencies and patterns in the data. This allows for more accurate predictions compared to models that do not account for such complex relationships. Furthermore, the model's architecture enables parallel processing of input sequences, enhancing computational efficiency.

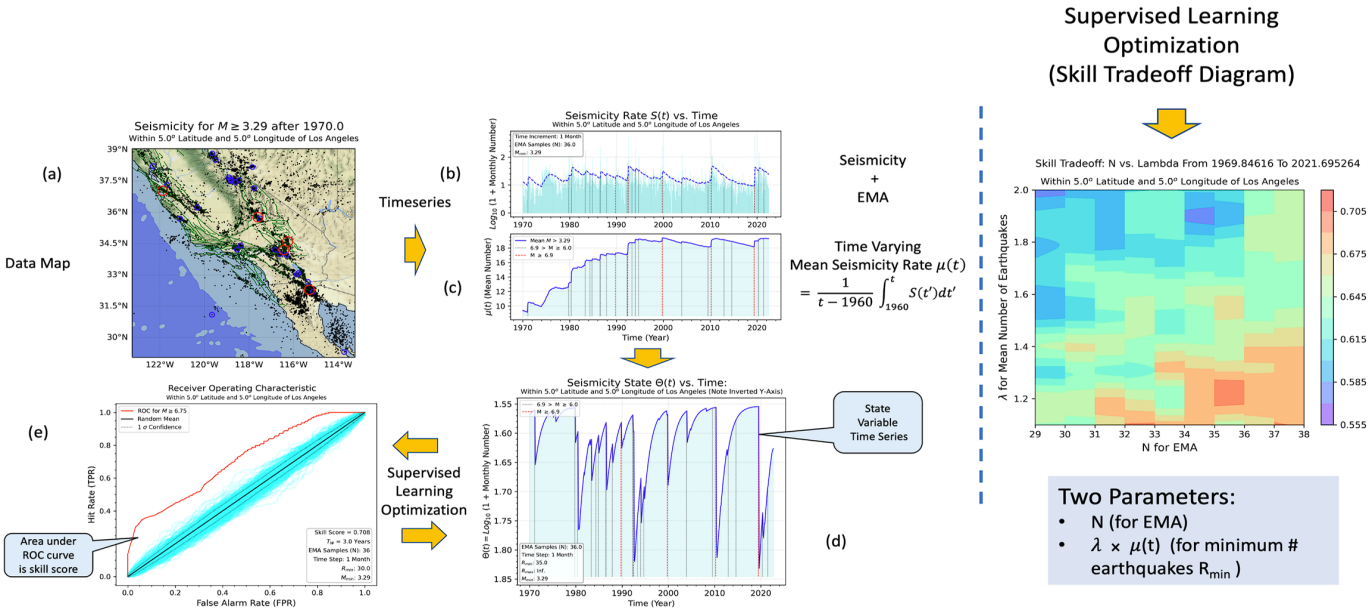

Figure 1. Illustration of the construction of a nowcast model for California. The nowcast is a 2-parameter filter on the small earthquake seismicity [42,43]. (**a**) Seismicity in the Los Angeles region since 1960, M > 3.29. (**b**) Monthly rate of small earthquakes as cyan vertical bars. The blue curve is the 36-month exponential moving average (EMA). (**c**) Mean rate of small earthquakes since 1970. (**d**) Nowcast curve that is the result of applying the optimized EMA and corrections for the time-varying small earthquake rate to the small earthquake seismicity. (**e**) Optimized receiver operating characteristic (ROC) curve (red line) used in the machine learning algorithm. Skill is the area under the ROC curve and is used in the optimization. Skill trade-off diagram shows the range of models used in the optimization.

The initial results of a feasibility study are for a similar region studied in this paper, using simulated test data, and are shown in Figure 2. The data here originate from new ERAS (Earthquake Rescaled Aftershock Seismicity) simulated data from [44]. These results are based on training the transformer model on 2021 years of ERAS data and then applying it to 53 years of independent test ERAS data (cyan region), a time span similar to the

observed data in California. Then, nowcasts were made for data values beyond the test data by feeding the previously predicted output values into the transformer as inputs to predict future values as described previously.

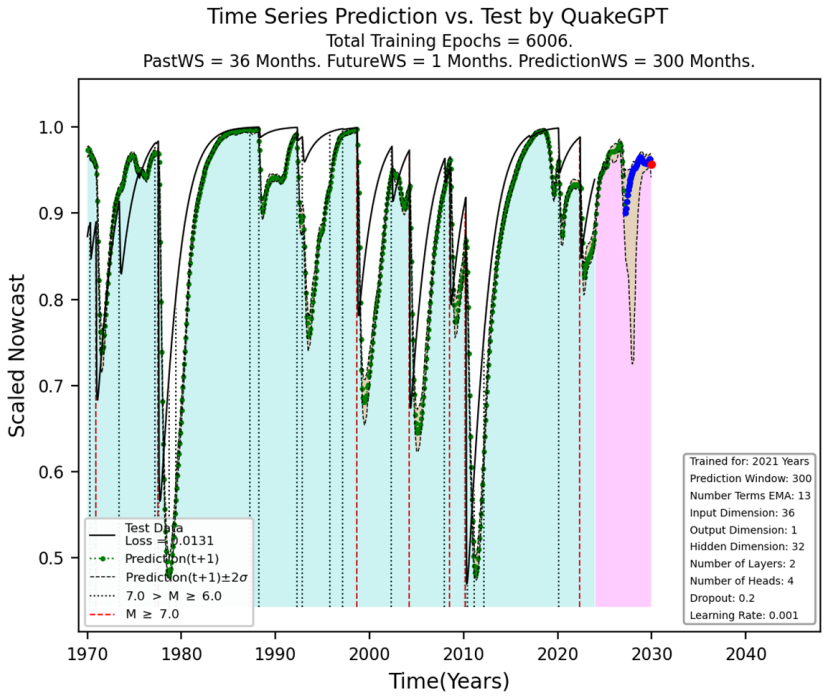

Figure 2. Image showing the application of the trained QuakeGPT transformer to an independent, scaled nowcast validation curve (green shading), followed by prediction of future values beyond the end of the nowcast curve (magenta shading). In this model, 36 previous values are used to predict the next value. Dots show the predictions and the solid line shows the nowcast curve whose values are to be predicted. Green dots show the predictions of the transformer up to the last 37 values. The 36 blue dots are predictions that were made and then fed back into the transformer to predict the final point (red dot). In this model, 50 members of an ensemble of runs were used to make the predictions. The dots represent the mean predictions. Brown areas represent the 1-sigma standard deviations to the mean values. In this model, 2021 years of simulation data were used to train the model.

3. Data

In this study, we focus on Southern California, a region renowned for its significant seismic activity and extensive fault lines. The earthquake data utilized were sourced from the US Geological Survey (USGS) online catalogs [45]. Our analysis encompasses a geographical area defined by a 4-degree latitude span (32° N to 36° N) and a 6-degree longitude range (−120° to −114°), as illustrated in Figure 3. The dataset spans from 1986 to 2024 and includes events recorded with their magnitude, epicenter, depth, and time.

The primary aim of this research is to forecast the cumulative released energy of earthquakes within a two-week horizon, thereby enhancing the understanding and nowcasting of significant earthquakes. To achieve this objective, the designated area of interest is segmented into discrete spatial bins, enabling localized earthquake nowcasting predictions within each bin. Note at this stage, we are exploring [2,42], both new model architectures, different choices in the quantity nowcast, and the metric of success. As our main interest is understanding different deep learning architectures, we fix both the nowcasted quantity and metrics across the different time series models investigated here.

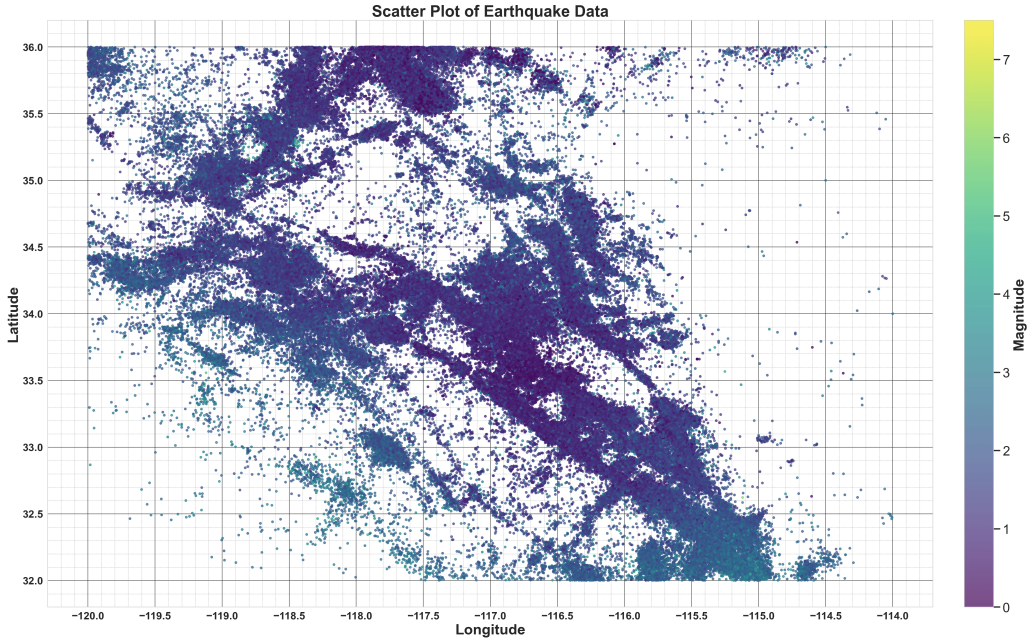

Figure 3. Distribution of earthquake epicenters in Southern California (32° N to 36° N, −120° to −114°) from USGS data (1986–2024). The scatter plot shows the spatial density of seismic events used to analyze and optimize spatial bins for earthquake nowcasting.

Following our previous study [2], we partition the mentioned area into 0.1 × 0.1 degree squares, each with a side length of approximately 11 km, resulting in a total of 2400 spatial bins. The decision to use a 0.1-degree grid is inspired by the RELM test initially performed by Ned Field at the USGS, where this specific discretization was applied [46].

The quantification of the energy released by earthquakes over a given time period is critical for understanding seismic activity patterns. Built on previous studies, we use the following formula to calculate the logarithm of the energy released by seismic events within a specified bin and time period [47]:

$$\mathrm{LogEn}_{(bin,time\_period)} = \log(\mathrm{energy}) = \frac{1}{1.5}\log_{10}\left(\sum_{\mathrm{quakes}} 10^{1.5 \cdot \mathrm{m}_{\mathrm{quake}}}\right) \quad (1)$$

In this formula, $\mathrm{LogEn}_{(bin,time\_period)}$ denotes the logarithm of the total energy released. The summation ($\sum_{\mathrm{quakes}}$) is carried out over all earthquake events occurring within the designated bin and time period. $\mathrm{m}_{\mathrm{quake}}$ is the magnitude of an earthquake. Each earthquake's magnitude is scaled by a factor of 1.5 and then used as the exponent for a base of 10, following the Gutenberg–Richter relationship, which relates earthquake magnitude to energy release [48]. The resulting values are summed to provide a cumulative measure of seismic energy.

To analyze the temporal distribution of seismic energy, we consider a time window of 14 days. This allows us to create biweekly time periods to have samples of seismic activity from 1986 to 2024. By dividing the data into these 14-day intervals, we can systematically assess changes in the total energy release over time. This approach provides a detailed and continuous record of seismic energy, facilitating a better understanding of long-term trends and patterns in earthquake activity.

As previously mentioned, we generate a time series of logarithmic energy for each spatial bin. However, as shown in Figure 3, some bins have experienced relatively few earthquakes over the past 40 years. Forecasting seismic activity in these sparsely active bins would not yield meaningful results. Consequently, we focus our analysis on the 500 most active and vulnerable bins out of the total 2400 within the study area, as illustrated in

Figure 4. This selection ensures that we concentrate on areas with a high risk of earthquake occurrence, thereby enhancing the reliability of our nowcasting system.

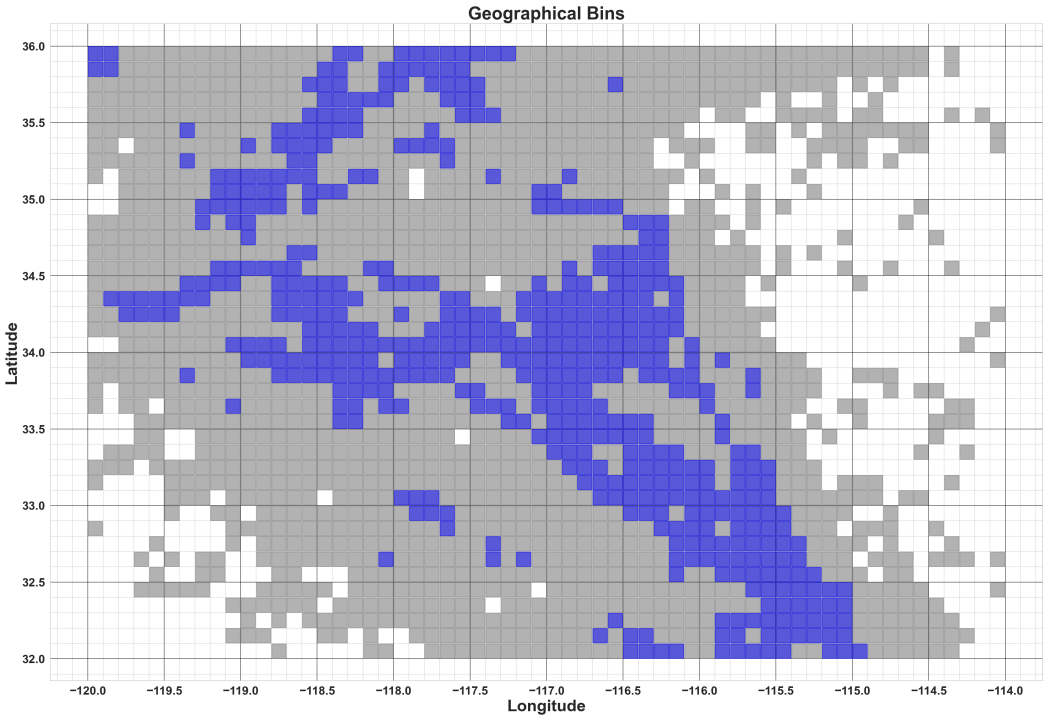

Figure 4. The 500 most active and vulnerable spatial bins, marked in blue, selected for analysis out of the total 2400, based on the frequency of earthquakes from 1986 to 2024. This selection focuses on high-risk areas.

Our dataset spans the years 1986 to 2024, encompassing a total of 1000 two-week samples. We utilize the first 80% of these samples as training data, while the remaining 20% are designated as test data. Figure 5 presents six time series from six randomly selected spatial bins, with earthquakes of magnitude greater than 5 prominently marked in each time series. This plot highlights the temporal distribution and intensity of significant seismic events across different regions.

Note that we must utilize a single feature; however, some of our models can incorporate multiple time series from various spatial bins. To ensure a fair comparison between models, we restrict our input to only time series values, as some models are limited to handling a single feature. However, we generate sequences in each sample consisting of 52 or 130 values to forecast the subsequent value. Additionally, all time series are normalized to have a maximum absolute value of 1 across all spatial and temporal data points. This normalization facilitates consistent and unbiased model evaluation. Although we primarily focus on time series values, we also explore certain scientifically identified features that enhance earthquake nowcasting. In our sub-experiments, we evaluate the performance of models that can accept multiple features using these advanced scientific features [2,43].

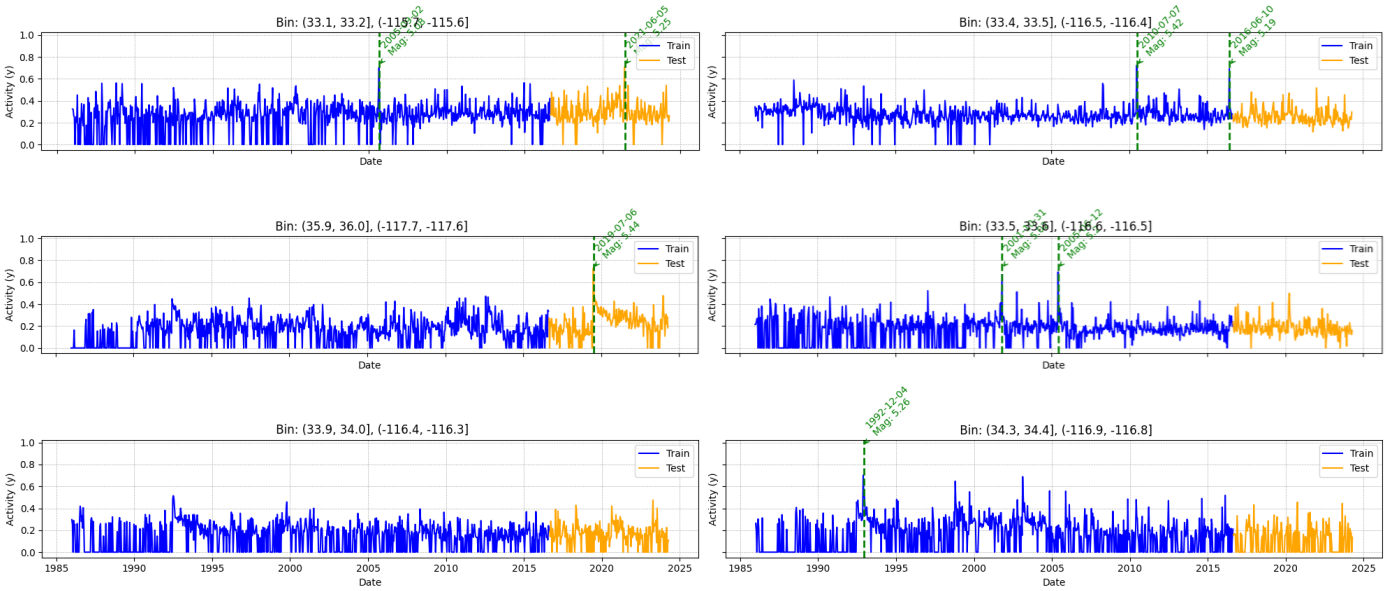

Figure 5. Six time series from randomly selected spatial bins, highlighting earthquakes of magnitude greater than 5.

3.1. Graph Structure for GNNCoder

In this subsection, we introduce our method for creating the graph structure needed for GNN models. This graph structure enables GNNCoder to effectively utilize and learn from the spatial relationships and interactions between different data points. An earthquake graph structure can be constructed based on various relationships, such as the locations of fault lines and their interactions, or the positions of seismic sensors [18]. However, to objectively evaluate the effectiveness of different deep learning architectures on the multifaceted earthquake problem, we create a geographical graph based solely on the proximity of spatial bins. This approach ensures a fair comparison by excluding additional sources of information and focusing exclusively on the intrinsic spatial relationships within the data.

To construct this graph, we treat each spatial bin as a node and employ the epsilon nearest neighbor graph (epsilon-NNG) algorithm to define the edges [49]. The epsilon-NNG algorithm operates by establishing connections between bins based on their proximity to one another. Specifically, spatial bins that fall within a predefined epsilon distance threshold are linked, forming edges in the resulting graph. This method allows for the identification of spatial relationships and dependencies among neighboring bins, providing valuable insights into the spatial clustering and patterns of seismic activity.

In determining the epsilon value, we considered the geographical distances between bin centers to ensure effective connectivity. Using an epsilon of 0.05 would result in no connections, as the distance between the centers of two adjacent bins is 0.10 geographical degrees. At epsilon = 0.10, bins would connect only to their horizontal and vertical neighbors, forming a simpler structure that lacks diagonal connections. With epsilon = 0.15, however, each bin connects to its horizontal, vertical, and diagonal neighbors, creating a more interconnected and robust graph to capture spatial interactions between areas and their fault lines. In our experiments, we use an epsilon of 0.15 degrees. This setup enhances the GNNCoder's ability to capture the spatial relationships, where we use the attention mechanism to decide which areas are related together.

Focusing on the 500 most active bins, the epsilon-NNG algorithm initially generates a multi-component graph. To achieve full connectivity, we link each component to the nearest node in an adjacent component, thereby creating a single, cohesive graph. This step is essential, as GNN layers aggregate information from specific depths within the graph, and isolated nodes can introduce noise into the embedding process. By utilizing

this unweighted graph, we allow the attention mechanism within the GNN layers to determine the significance of each edge, enhancing the model's ability to capture critical spatial relationships. Figure 6 represents our final graph structure.

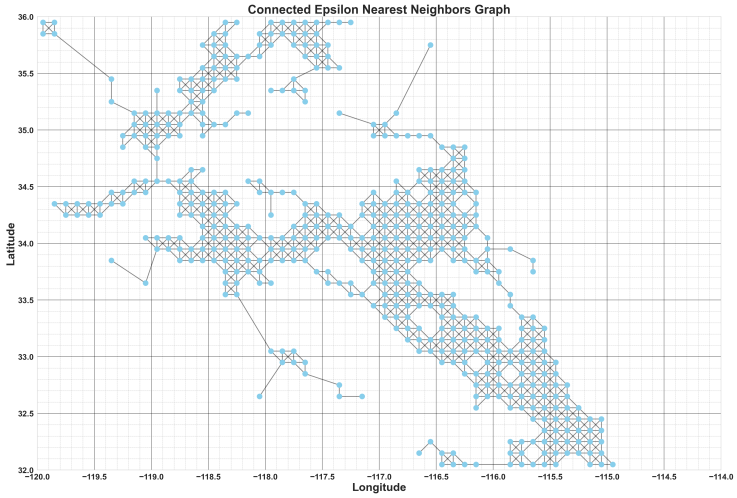

Figure 6. The final graph structure representing the 500 most active bins, created using an epsilon of 0.15 degrees. Initially forming a multi-component graph, components are linked to ensure full connectivity.

3.2. Pre-Training Datasets for Transformer Models

In this subsection, we describe the pre-training datasets for our selected foundation models. Starting with TimeGPT, it is trained on an extensive dataset comprising time series from various domains, such as finance, economics, demographics, healthcare, weather, IoT sensor data, energy, web traffic, sales, transport, and banking. This dataset includes more than 100 billion data points [28].

Chronos is trained using a vast collection of publicly available time series datasets, further enhanced by a synthetic dataset generated via Gaussian processes to improve generalization. The inclusion of synthetic data ensures that Chronos can generalize well, providing robust zero-shot performance on unseen forecasting tasks [30].

Time-LLM, which reprograms GPT-2 for time series forecasting, leverages GPT-2's pre-trained language model that was trained on diverse natural language corpora, such as the WebText dataset. This dataset includes over 8 million web pages and is designed to capture a wide range of linguistic contexts and patterns [50].

Additionally, we pre-train iTransformer, PatchTST, and TSMixer on three diverse datasets: the Weather dataset [32], the TrafficL dataset [51], and the M4 dataset [52]. Both the Weather and TrafficL datasets share important temporal and spatial characteristics with seismic data, making them particularly relevant for earthquake nowcasting. The Weather dataset captures seasonal and temporal variations along with spatial dependencies across regions, closely mirroring the extended temporal and spatial correlations seen in seismic activity. Similarly, the TrafficL dataset incorporates spatial patterns and periodic trends reflective of recurrent flows in specific regions, which can help the model detect cyclic spatial-temporal behaviors that often precede seismic events. These spatial and temporal similarities provide valuable foundations for pre-training the models on data that resemble seismic structures. The M4 dataset, in contrast, is a widely recognized multi-purpose dataset for time series, offering extensive temporal variety across numerous domains, which bolsters the models' generalization across different time scales. This approach allows us to leverage well-established temporal and spatial patterns in these datasets [32,53].

3.2.1. TrafficL Dataset

The TrafficL dataset, as presented in [51], collected by the California Department of Transportation, includes hourly occupancy rates from 862 road sensors, spanning from January 2015 to December 2016. This dataset could be particularly beneficial for earthquake forecasting. Traffic data capture both long-term trends and short-term fluctuations, providing a comprehensive view of temporal dynamics that might be similar to those found in seismic data.

3.2.2. Weather Dataset

The Weather dataset, as presented in [32], comprises 21 meteorological measurements recorded every 10 min throughout the year 2020 at the Weather Station of the Max Planck Biogeochemistry Institute in Jena, Germany. This dataset might be beneficial in earthquake forecasting as it provides extensive time series data that help the models learn complex temporal patterns and variability in environmental conditions, which can be similar to the temporal dynamics of seismic activities.

3.2.3. M4 Dataset

The M4 dataset, as presented in [52], is a comprehensive and large-scale collection of time series data that serves as a standard benchmark for time series forecasting [54]. It includes various types of time series data, offering a robust baseline for training models to handle different temporal patterns and anomalies. With thousands of time series across multiple domains, the dataset exposes models to a wide range of temporal behaviors, enhancing their adaptability to the unique characteristics of seismic data. As a widely recognized benchmark, the M4 dataset ensures that pre-trained models are calibrated against a high standard, improving their reliability.

We use the monthly dataset from M4, which includes 48,000 time series. The length of these time series varies from 60 to 2812 data points. To ensure sufficient sequence length for creating samples, we sort all the time series in this dataset and retain the 32,000 longest time series in our experiments.

4. Models Description

This section provides an in-depth description of our comprehensive methodology to develop an earthquake nowcasting system using time series foundation models and advanced deep learning architectures. Our proposed method systematically compares several key architectures to effectively capture and analyze seismic data, highlighting their respective strengths and limitations.

In this work, we introduce two powerful models: Multi Foundation Quake and GNNCoder. Multi Foundation Quake employs a straightforward yet unique approach to aggregate multiple foundation models, significantly enhancing the performance of each individual model.

On the other hand, GNNCoder utilizes a graph attention network, moving beyond traditional methods that predominantly focus on temporal features. This method emphasizes geospatial relationships and clusters, leveraging GNNs to analyze seismic data, innovatively.

In addition, we modify some powerful foundation models to forecast earthquakes, aiming to analyze their capacities to uncover temporal features as well as indirect spatial features inherent in seismic data. We also incorporate memory-based models that have demonstrated satisfactory results in earthquake forecasting [2,18]. We discuss each model's structure and specific mechanisms for capturing dependencies in seismic data.

4.1. Pre-Trained Transformer Models

This section provides a detailed description of the transformer models evaluated in our study for earthquake nowcasting, emphasizing their architectural innovations and unique capabilities. The pre-trained transformer models evaluated in our study—iTransformer, PatchTST, TimeGPT, Time-LLM and Chronos—bring unique innovations to the task of

earthquake nowcasting. By leveraging self-attention mechanisms, these models offer advanced capabilities for capturing complex temporal patterns and enhancing nowcasting accuracy [55].

iTransformer, introduced by Liu et al. (2024) [26], presents a novel adaptation of the transformer architecture specifically designed for time series forecasting. This innovative model addresses key challenges faced by traditional forecasting methods, such as capturing long-range dependencies and handling multivariate data effectively. Instead of employing a conventional encoder-decoder structure, the iTransformer utilizes an inverted dimension approach. In this configuration, the time points of individual series (each spatial bin) are embedded into variate tokens. Then, the tokens from different regions are processed through multi-head self-attention mechanisms, creating temporal-spatial tokens. This enables the model to capture seismic spatial relationships.

A critical component of the iTransformer is the attention mechanism, which is pivotal for capturing dependencies in the time series data. The model projects the input data into three distinct vectors: queries (Q), keys (K), and values (V). The attention scores, calculated using the dot product of the query and key vectors, determine the relevance of each element in the sequence relative to others. These scores are scaled and passed through a softmax function to obtain attention weights, which highlight the importance of various elements. The weighted sum of the value vectors, guided by these attention weights, allows the model to aggregate information from the entire sequence from all variates effectively. The model employs multiple attention heads, each independently attending to different parts of the input sequence.

Then, each variate token undergoes further processing through a position-wise fully connected feed-forward network. This network facilitates the learning of nonlinear representations for each variate token, enhancing the model's ability to represent complex temporal patterns. According to the authors, by leveraging pre-training on the TrafficL and Weather dataset, the model has to be able to gain a robust understanding of temporal patterns, which is crucial for accurate earthquake nowcasting.

PatchTST, introduced by Nie et al. (2023) [27], is a novel model designed to enhance the efficiency and accuracy of multivariate time series forecasting using transformer-based architectures. This model incorporates two key innovations: the segmentation of time series data into subseries-level patches and the concept of channel-independence. By segmenting the time series into patches, PatchTST preserves local semantic information within each patch, which serves as input tokens for the transformer. The input data include all earthquake time series from each bin (small spatial square); so, this means input tokens for the transformer are representations from a long period of series. This segmentation not only retains important local temporal patterns but also significantly reduces the computational and memory overhead associated with generating attention maps, as the patches are smaller than the full time series. Consequently, PatchTST can process longer historical data more efficiently, enabling it to capture long-term dependencies that are crucial for accurate forecasting.

The second critical component of PatchTST is its channel-independent design. In traditional multivariate time series models, channels are often treated together, which can lead to complex interactions and increased computational demands. In contrast, PatchTST treats each channel as an independent univariate time series, sharing the same embedding and transformer weights across all channels. The channel-independent design also facilitates the transferability of the model across different datasets. By pre-training on one dataset and fine-tuning on another, PatchTST achieves state-of-the-art forecasting accuracy, showcasing its ability to generalize across different domains [27].

TimeGPT, introduced by Garza et al. (2023) [28], represents a groundbreaking advancement in time series forecasting by developing the first foundation model tailored for this domain. The architecture of TimeGPT leverages insights from transformer-based models. It processes a window of historical values to produce forecasts, incorporating local positional encoding to enrich the input. The architecture follows an encoder-decoder

structure with multiple layers, each featuring residual connections and layer normalization. The decoder's output is mapped to the forecasting window dimension through a linear layer, allowing the model to capture the diversity of past events and accurately nowcast potential future distributions.

Time-LLM, as proposed by Jin et al. (2024) [29], is a cutting-edge model for time series forecasting that utilizes the pre-trained GPT-2 model from OpenAI [50]. Time-LLM transforms time series data into a form that GPT-2 can understand by tokenizing the input data into patches and embedding these patches to align with GPT-2's word embeddings. These patch embeddings, enhanced with additional statistical features, are passed into GPT-2 to generate output representations, which are then projected into the final forecast through a linear layer. This process allows Time-LLM to harness the capabilities of large language models like GPT-2 for time series tasks, efficiently reprogramming them to handle temporal data.

One of the key innovations of Time-LLM lies in its Prompt-as-Prefix technique, where task-specific prompts are prepended to time series data, enabling the LLM to interpret it in the context of its pre-training. This approach allows Time-LLM to excel in both few-shot and zero-shot learning scenarios, improving its performance on time series forecasting tasks without the need for extensive fine-tuning.

Chronos, presented by Ansari et al. (2024) [30], is a sophisticated text-based framework designed for pre-trained probabilistic time series models, leveraging the principles of natural language processing to enhance time series forecasting. The core innovation in Chronos lies in its tokenization process, where time series values are scaled and quantized into a fixed vocabulary. This approach allows the framework to convert continuous time series data into a discrete sequence of text-like tokens, making it compatible with transformer-based language models. The framework utilizes models based on the T5 family [24], with parameter sizes ranging from 20 million to 710 million, to capture varying degrees of complexity in time series patterns.

The training process for Chronos is both comprehensive and robust. The training corpus is extensive, comprising a large collection of publicly available datasets from diverse domains. To further reinforce the generalization capabilities of the models, the authors generated a synthetic dataset using Gaussian processes. This combination of real-world and synthetic data ensures that Chronos models can learn intricate temporal dependencies and variances, preparing them for a wide array of forecasting scenarios.

4.2. Graph Neural Networks Models

We propose an innovative approach, called GNNCoder, that enhances the understanding of spatial relationships among geological regions using graph neural networks and encoder-decoder components. This methodology creates a holistic framework for earthquake forecasting, addressing the limitations of existing models that often overlook complex interactions between geological features and the temporal evolution of seismic activities.

Upon our introduced earthquake-correlated graph structure, GNNCoder incorporates the roles of faults and other geographical entities. GNNs are particularly adept at modeling and analyzing complex dependencies in graph-structured data, making them ideal for capturing the intricate geographical interactions and dependencies inherent in earthquake data.

Our model architecture includes an MLP-based encoder-decoder component, complemented by multiple graph attention network (GAT) layers. The encoder-decoder, constructed with dense layers, is essential for capturing and transforming input data into a meaningful representation. The GAT layers enhance the model's capacity to handle graph-structured data by dynamically adjusting the importance of neighboring nodes through an attention mechanism [56]. This capability is crucial for earthquake forecasting, as it enables the model to focus on the most relevant connections and interactions within the seismic data. Ultimately, the model architecture concludes with an additional dense layer designed to forecast seismic energy values.

The attention mechanism allows GATs to adaptively focus on specific parts of the graph relevant to the task. A key feature of GATs is their utilization of self-attention, or intra-graph attention mechanisms, to compute the hidden representations of each node in the graph. The attention coefficients, which are central to this process, are computed as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathrm{a}^T[\mathrm{W}\mathrm{h}_i \| \mathrm{W}\mathrm{h}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\mathrm{a}^T[\mathrm{W}\mathrm{h}_i \| \mathrm{W}\mathrm{h}_k]))} \quad (2)$$

where h_i is the feature vector of node i, W is a weight matrix, a is a weight vector in the attention mechanism, $\|$ denotes concatenation, and α_{ij} is the attention coefficient between nodes i and j. Unlike other graph neural network models that depend on the entire neighborhood's aggregate information, GATs allow for the weighting of nodes' features based on their relevance [56,57].

In our experiments, we utilize three different GNNCoders, varying from 1-layer to 3-layer GAT architectures. The 1-layer GAT model consists of a single graph attention layer that aggregates information from the immediate neighbors of each node using attention scores to weigh the importance of these neighbors. Each node's features are combined with their neighbors' features, weighted by the attention scores, followed by a non-linear activation function. The 2-layer and 3-layer GAT models extend this architecture by adding more graph attention layers. These additional layers allow the model to capture dependencies up to two and three hops away, respectively, thereby enhancing its ability to learn broader spatial interactions. However, increasing the depth also increases computational complexity and the risk of overfitting, as highlighted by [56].

4.3. Memory-Based Models

Earthquake nowcasting relies heavily on the analysis of temporal patterns in seismic data. Aftershocks and foreshocks exhibit patterns that are critical for understanding and forecasting seismic activity [6]. Seismic data are inherently complex, characterized by irregular intervals and varying magnitudes of seismic waves. Memory-based models are adept at identifying and learning from these patterns due to their ability to maintain long-term dependencies. We use memory-based models such as DilatedRNN, TFT, and LSTM to forecast earthquake time series, and we provide a detailed description of them below.

DilatedRNN, as described by Chang et al. (2017) [33], is a novel architecture designed to address the challenges of learning long-term dependencies in sequential data. Traditional RNNs, such as LSTMs and GRUs, often struggle with these dependencies due to issues like vanishing and exploding gradients. DilatedRNN introduces a dilation mechanism inspired by dilated convolutions used in CNNs. This mechanism allows the model to skip certain time steps and capture longer temporal dependencies more efficiently. By incorporating dilations into the recurrent architecture, DilatedRNN can effectively balance between capturing short-term and long-term dependencies without a significant increase in computational complexity.

The architecture of DilatedRNN is characterized by its unique dilation patterns, which specify the intervals at which the model accesses past time steps. These patterns are carefully designed to ensure that the model captures a wide range of temporal dependencies. For instance, a dilation pattern might access every other time step in the first layer, every fourth time step in the second layer, and so on. This hierarchical approach allows the network to maintain a larger receptive field and efficiently integrate information from various time scales. The use of dilations mitigates the problem of long-term dependency learning by reducing the effective path length through the network, which, in turn, enhances gradient flow and model performance.

Temporal Fusion Transformer (TFT), detailed by Lim et al. (2021) [34], is a sophisticated attention-based architecture designed for interpretable multi-horizon time series forecasting. TFT integrates the capabilities of deep learning with the necessity for interpretability, leveraging both recurrent layers for local sequence processing and self-attention

mechanisms to capture long-term dependencies. This dual approach enables TFT to learn temporal relationships at multiple scales effectively.

A key feature of TFT is its specialized components that ensure the model's high performance and interpretability. These include variable selection networks that judiciously choose relevant features from a potentially large set of inputs, thereby enhancing the model's ability to focus on the most informative variables. Additionally, TFT employs gating layers that dynamically suppress irrelevant or redundant components within the model, which not only streamlines the computational process but also mitigates overfitting. By combining these elements, TFT achieves a robust balance between complexity and interpretability, offering insights into how different variables and temporal patterns influence the forecasting outcomes.

4.4. Convolutional and MLP-Based Models

We also employ several powerful and renowned convolutional models due to their ability to automatically and adaptively learn spatial hierarchies from seismic patterns. Furthermore, we use MLP-based models that excel at integrating diverse data sources, such as historical earthquake records.

TSMixer, as described by Chen et al. (2023) [31], represents a novel approach to time series forecasting that leverages the simplicity and effectiveness of multi-layer perceptrons. Unlike transformers and memory-based models, which often rely on recurrent or attention-based mechanisms to capture temporal dependencies, TSMixer utilizes mixing operations along both the time and feature dimensions. This design choice allows the model to efficiently extract information and capture complex dynamics inherent in multivariate time series data. By focusing on linear models, TSMixer demonstrates that high-capacity architectures are not always necessary for achieving state-of-the-art performance, challenging the prevailing notion that more complex models are inherently superior.

TimesNet, described by Wu et al. (2023) [36], is an innovative framework designed to address the inherent challenges in time series analysis by transforming 1D temporal data into 2D representations. The core idea behind TimesNet is the decomposition of complex temporal variations into intraperiod and interperiod variations, which are then mapped into 2D tensors. This transformation is pivotal as it allows the model to utilize 2D convolutional kernels to effectively capture and process temporal patterns that are otherwise difficult to discern in a 1D format. By embedding intraperiod variations into the columns and interperiod variations into the rows of the 2D tensors, TimesNet leverages the power of 2D convolutional networks to achieve superior representation and modeling of time series data.

The primary component of TimesNet is the TimesBlock, a task-general backbone specifically designed for time series analysis. TimesBlock is equipped with a parameter-efficient inception block that adapts to the multi-periodicity inherent in time series data. This block can dynamically discover and extract intricate temporal variations from the transformed 2D tensors. The inception mechanism within TimesBlock enables the model to handle multiple scales of temporal patterns, thereby enhancing its ability to forecast, classify, impute, and detect anomalies across various time series datasets. The adaptability and efficiency of TimesBlock make it a versatile tool for a wide range of time series analysis tasks.

Temporal Convolutional Network (TCN), introduced by Bai et al. (2018) [35], is an innovative architecture that leverages convolutional layers for sequence modeling tasks, challenging the traditional dominance of recurrent networks like LSTMs. TCNs are designed to handle sequence data by applying convolutional layers across the temporal dimension, thus capturing temporal dependencies through a hierarchy of filters. This approach enables TCNs to model long-range dependencies more effectively than recurrent networks, as they avoid the vanishing gradient problem typically associated with deep recurrent architectures. A key characteristic of the TCN is its use of dilated convolutions. Dilated convolutions allow the network to have a larger receptive field without significantly increasing the number of parameters or the computational cost. This dilation means that

the convolutional layers can skip certain inputs, allowing the TCN to aggregate information from a wider range of previous time steps, making it particularly well-suited for tasks requiring long memory.

Time-series Dense Encoder (TiDE), described by Daset al. (2023) [37], is an innovative approach designed to address the challenges of long-term time-series forecasting. Unlike traditional models that either rely on the simplicity of linear methods or the complexity of transformer-based architectures, TiDE leverages the strengths of an MLP-based encoder-decoder framework. This model offers a balanced combination of simplicity, speed, and the capability to handle both covariates and non-linear dependencies in the data. At its core, TiDE uses a dense encoding mechanism that efficiently captures the temporal dependencies inherent in time-series data, ensuring accurate and robust forecasts over extended horizons. The architecture of TiDE is built around an encoder-decoder structure, where the encoder processes input time-series data to generate a dense representation, and the decoder utilizes this representation to produce the forecast. This structure allows TiDE to effectively learn from historical data while adapting to the presence of external variables (covariates), which can influence the future trajectory of the time series.

4.5. Multi Foundation Quake

Finally, we introduce Multi Foundation Quake, an innovative approach for earthquake forecasting that leverages the power of multiple foundation models to enhance the accuracy and robustness of nowcasting. The architecture of Multi Foundation Quake consists of two main components: foundation models and a pattern model. The foundation models, such as iTransformer, TFT, and PatchTST, each bring unique strengths to the table. For example, the iTransformer utilizes a specific transformer architecture for capturing long-range dependencies, whereas other models might excel in different areas. The outputs from these foundation models serve as inputs to a pattern network. A pattern network is a non-foundation model trained directly on the target task, focusing solely on learning patterns relevant to that specific domain. This concept can be extended to develop a MultiFoundationPattern model for other domains.

In Multi Foundation Quake 1, we utilize six models, including DilatedRNN, iTransformer-TrafficL, TFT, TCN, and TSMixer-TrafficL, in the first component, and adopt an LSTM model for the pattern model in the second component. The LSTM is adept at learning sequential dependencies and temporal dynamics, making it well-suited for processing the enriched feature representations provided by the foundation models.

In Multi Foundation Quake 2, we follow a similar architecture but utilize a graph attention network (GAT) for the pattern model in the second component. The choice of GAT is driven by its ability to capture temporal dependencies across all locations, while also considering temporal patterns between neighboring areas. This approach is crucial for effectively modeling the interactions between different seismic regions, enhancing the overall accuracy of the earthquake nowcasting process.

The training process of Multi Foundation Quake involves several key steps. Initially, each foundation model is individually pre-trained on a pre-training dataset based on its designed purpose. In addition to the foundation models, our approach can incorporate other large models as well. Each foundation model is subsequently fine-tuned on seismic data, encapsulating various temporal patterns and dependencies relevant to earthquake nowcasting. These features are concatenated and fed into the pattern model, which is trained to learn the sequential dependencies and improve the accuracy of earthquake nowcasting.

Multi Foundation Quake offers several advantages over traditional earthquake nowcasting models. By combining the strengths of multiple foundation models, it captures a wider range of temporal patterns and dependencies. The integration of diverse feature representations ensures that the model is robust to various seismic data characteristics, leading to more accurate and reliable earthquake nowcasting. Additionally, the modular design of Multi Foundation Quake allows for the incorporation of additional founda-

tion models in the future, enhancing its scalability and adaptability to evolving seismic nowcasting techniques.

4.6. Model Training and Implementation

Prior to training, the raw earthquake data are preprocessed to generate time series inputs for each spatial bin. The dataset is divided into 14-day intervals to create biweekly samples of seismic activity. The energy released by earthquakes within each bin and time period is calculated using the logarithm of the summed seismic energy. This preprocessing step ensures that the input data are normalized, facilitating the training process.

In our experiments, we utilize transfer learning by performing both pre-training and fine-tuning through supervised learning. The input sequences (X) consist of either 52 or 130 values representing past seismic activities, with the subsequent value serving as the target (Y). This means that 2 or 5 years of information is used to forecast the subsequent value, enabling the models to learn temporal dependencies effectively.

For pattern models like GNNCoder and LSTM, we leverage supervised learning, where each input sample x_i has a corresponding label y_i, representing the next value in the sequence. This consistent labeling enables the models to learn temporal patterns effectively. In contrast, for foundation models, we adopt a transfer learning approach that involves both pre-training and fine-tuning in a supervised framework. Here, the objective is to optimize the model parameters to minimize the prediction error, measured by the Mean Squared Error (MSE) loss function. MSE is particularly effective for emphasizing larger errors, which is crucial for capturing significant seismic events.

Training is conducted over multiple epochs, with each model undergoing separate hyperparameter optimization, including the number of epochs. The learning rate for all models is set at 1×10^{-4} with a batch size of 32. For transformer-based models, the number of layers and attention heads varies depending on the specific architecture, allowing us to tailor each model's capacity to the task requirements. We applied the Adam optimizer to train the GNNCoder, and to avoid over-fitting, we applied L2 regularization and dropout techniques. The weights of the GNN layers were initialized using the Glorot uniform initializer [58].

For the following models (iTransformer, PatchTST, TSMixer), transfer learning is applied by pre-training on large-scale datasets, such as Weather, TrafficL, and M4. This step enables the models to learn general temporal patterns and improve their performance on the earthquake nowcasting task. The pre-trained models are then fine-tuned on the earthquake dataset, allowing them to adapt to the specific characteristics of seismic data. The Chronos and TimeGPT models are already trained on large-scale datasets and, according to the authors, do not require fine-tuning. The Time-LLM is already trained on the WebText dataset and we fine-tune it on seismic data.

For the transformer models, we utilize the Python package [59], which is based on PyTorch. For the GNN models, we employ the Python package [60], which is built on TensorFlow.

This study is based on a variant of the Earthquake code in the MLCommons [61] Science benchmarks [62–65]. We will submit our measurements there as our answer to their challenge to improve scientific discovery in this area.

5. Models Evaluation and Comparison

In this section, we detail our experimental setup and evaluation process and present the results and discussion. We compare the performance of advanced deep learning architectures and foundation models on earthquake datasets. Furthermore, we conduct deep investigations of earthquake time series, spatial dependencies, and feature analysis to provide comprehensive insights into model performance. We begin by explaining the evaluation metrics, followed by a comprehensive presentation of the results for both the proposed models and the baselines.

5.1. Evaluation Metrics

The scientific objective of the present work is to enhance the quality of earthquake nowcasting using deep learning in a region of Southern California. Similar to that used in previous work [3,8], we use the Normalized Nash-Sutcliffe Efficiency (NNSE) to evaluate the models [66]. NNSE is a normalized statistic that determines the relative magnitude of the residual variance compared to the measured data variance. It is used to assess the predictive power of earthquake nowcasting models. The formula for NSE is:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2} \tag{3}$$

$$\text{NNSE} = 1/(2 - \text{NSE}) \tag{4}$$

where, O_i is the observed value at time i, P_i is the nowcasted value at time i, $\bar{O}$ is the mean of the observed values, n is the number of observations.

This metric ranges from 0 to 1, where 1 signifies a perfect match between the model predictions and the observations, and 0.5 indicates that the model's predictions are as accurate as the mean of the observed data. This metric is particularly useful in the context of earthquake nowcasting as it provides a clear measure of how well the model's predictions match the observed data, accounting for the variability inherent in earthquake occurrences [2].

To comprehensively evaluate the performance of our models, we also employ two other metrics: Mean Squared Error (MSE), and Mean Absolute Error (MAE). We use MSE as our loss function, making it a natural and consistent choice for performance evaluation. These metrics offer a thorough understanding of various facets of the models' performance. These metrics provide a detailed understanding of various aspects of model performance. However, NNSE holds a significant advantage over MSE and MAE, as it is not dataset-dependent. This allows for a more consistent comparison of model results, thereby enhancing the reliability and accuracy of seismic nowcasting in future applications.

5.2. Results and Discussion

The performance evaluation of various deep learning models for earthquake nowcasting in Southern California reveals significant insights into the strengths and weaknesses of each approach. Table 1 defines the model characteristics and provides a detailed comparison of the models based on key metrics such as NNSE, MSE, and MAE. The third column of the table indicates a model is a Foundation model (F) or Pattern model (P). F refers to models that are pre-trained on large, diverse datasets to capture general temporal patterns, which can then be fine-tuned for specific tasks like earthquake nowcasting. On the other hand, P models are trained directly on the target task, focusing solely on learning the patterns relevant to that specific domain. The table is sorted by decreasing MSE. The results highlight the importance of model architecture and pre-training datasets in enhancing nowcasting accuracy for seismic activities.

Our introduced models, Multi Foundation Quake 1, Multi Foundation Quake 2, and GNNCoder, demonstrated superior performance across multiple metrics. Multi Foundation Quake 2, as the best model, achieved an MSE of 0.00625, an MAE of 0.0514, and an NNSE of 0.6175. This model leverages a hybrid architecture that combines several foundation models with a GNN as the pattern model. The GNN's ability to capture both spatial dependencies and temporal patterns across different seismic regions resulted in improved performance.

Multi Foundation Quake 1 follows a similar foundation model structure but replaces the GNN with an LSTM for the pattern model. Multi Foundation Quake 1 achieved an MSE of 0.00626, an MAE of 0.0516, and an NNSE of 0.6174. This demonstrates the effectiveness of LSTM in capturing sequential dependencies and temporal dynamics, although Multi Foundation Quake 2's GNN slightly outperformed it by better leveraging spatial relationships.

Table 1. Comparison of the performance of deep learning models for earthquake nowcasting in Southern California, ranked by MSE in descending order. The table compares various models used in this work, detailing their architectures, types (F for Foundation Model, P for Pattern Model), and datasets for pre-training and fine-tuning. The "Training Strategy" column lists the datasets used in the format: pre-training dataset/fine-tuning dataset. For Pattern models, there is no pre-training, and fine-tuning is performed on earthquake data.

Model	Architecture	Type	Training Strategy	MSE	MAE	NNSE
TimeGPT	Transformer	F	A broad dataset1/None	0.01042	0.0593	0.5484
iTransformer-M4	Transformer	F	M4/Earthquake	0.00702	0.0537	0.5902
Time-LLM	Transformer	F	WebText/Earthquake	0.00652	0.0522	0.6077
TSMixer-M4	MLP	F	M4/Earthquake	0.00651	0.0535	0.6081
Chronos	Transformer	F	A broad dataset2/None	0.00650	0.0519	0.6087
PatchTST-TrafficL	Transformer	F	TrafficL/Earthquake	0.00644	0.0501	0.6107
TiDE	MLP	P	None/Earthquake	0.00643	0.0519	0.6110
TSMixer-TrafficL	MLP	F	TrafficL/Earthquake	0.00643	0.0505	0.6111
TimesNet	CNN	P	None/Earthquake	0.00643	0.0560	0.6112
PatchTST-M4	Transformer	F	M4/Earthquake	0.00641	0.0504	0.6117
PatchTST-Weather	Transformer	F	Weather/Earthquake	0.00641	0.0502	0.6119
iTransformer-TrafficL	Transformer	F	TrafficL/Earthquake	0.00639	0.0513	0.6125
TCN	CNN	P	None/Earthquake	0.00637	0.0535	0.6132
VanillaTransformer	Transformer	P	None/Earthquake	0.00635	0.0498	0.6141
TFT	Transformer+RNN	P	None/Earthquake	0.00635	0.0555	0.6142
LSTM	RNN	P	None/Earthquake	0.00631	0.0514	0.6156
DilatedRNN	RNN	P	None/Earthquake	0.00630	0.0510	0.6159
GNNCoder	GNN	P	None/Earthquake	0.00628	0.0522	0.6166
Multi Foundation Quake 1	Hybrid+LSTM	F	Multi-domain/Earthquake	0.00626	0.0516	0.6174
Multi Foundation Quake 2	Hybrid+GNN	F	Multi-domain/Earthquake	0.00625	0.0514	0.6175

The GNNCoder models also showed strong performance. The GNNCoder (1-layer GAT) achieved an MSE of 0.00628, an MAE of 0.0522, and an NNSE of 0.6166. These results suggest that the one-layer GNNCoder effectively captures the spatial relationships inherent in seismic data, leveraging the proximity of spatial bins to nowcast earthquake activities accurately. We discuss the importance of graph structure and the number of GAT layers in Section 5.2.3, indicating how the increase in the depth of the network can impact performance.

The DilatedRNN model also performed well, with an NNSE of 0.6159 and an MSE of 0.00630, indicating its capability to model temporal dependencies effectively. The LSTM model, known for its effectiveness in time series forecasting, showed comparable performance to the DilatedRNN, with an MSE of 0.00631, an MAE of 0.0514, and an NNSE of 0.6156. However, their performances were slightly less favorable compared to GNNCoder. This suggests that while the DilatedRNN and LSTM capture sequential patterns efficiently, they may not fully exploit the spatial relationships between seismic events as effectively as the GNNCoder.

Pre-trained foundation models, including iTransformer, TimeGPT, PatchTST, and TSMixer, demonstrated varying degrees of success, heavily influenced by their pre-training datasets. TimeGPT, as the first foundation model in this domain, performed poorly with an MSE of 0.01042 and the lowest NNSE of 0.5484. TimeGPT's undesirable performance could be attributed to its inherent overgeneralization as a foundation model, the absence of fine-tuning on the specific seismic dataset, and the complex temporal resolution required to capture long-term earthquake patterns. Additionally, the iTransformer-M4 model, with an MSE of 0.00702 and an NNSE of 0.5902, emphasizes that pre-training on an irrelevant dataset can considerably decrease the accuracy of earthquake nowcasting. Conversely, the iTransformer-TrafficL model achieved an NNSE of 0.6125 and an MSE of 0.00639, suggesting that pre-training on the TrafficL dataset, which captures temporal dynamics from a road network, provided beneficial insights for earthquake nowcasting.

The PatchTST-Weather model, with an MSE of 0.00641 and an MAE of 0.0502, showed that weather data can offer valuable pre-training information, enhancing the model's ability to capture complex temporal patterns in seismic data. Similarly, the PatchTST-M4 and PatchTST-TrafficL models showed moderate performance, with MSEs of 0.00641 and 0.00644, respectively. These results underscore the critical role of selecting appropriate pre-training datasets to improve transformer model performance in specific domains. However, the VanillaTransformer outperformed all pre-trained models, demonstrating that focusing solely on learning patterns relevant to specific domain data (in our case, earthquake data) is more important for achieving high predictive accuracy in nowcasting.

Both Chronos and Time-LLM, as textual models, demonstrated moderate performance by applying natural language processing techniques to time series data, converting numerical values into text-like tokens for analysis. Time-LLM, which utilizes OpenAI's GPT-2 pre-trained on the WebText dataset, delivered underwhelming results with an MSE of 0.00652 and an NNSE of 0.6077. Despite the powerful capabilities of GPT-2, Time-LLM still struggles to properly capture the complex temporal and spatial dependencies required for high-accuracy earthquake nowcasting. This demonstrates that even state-of-the-art language models may not directly translate to success in the earthquake domain without additional modifications that account for both temporal and spatial intricacies.

The TimesNet model, designed to capture local temporal features, and the TiDE model, which focuses on non-linear relationships, both showed moderate performance. TimesNet achieved an MSE of 0.00643 and an NNSE of 0.6112, while TiDE had an MSE of 0.00643 and an NNSE of 0.6110. These models, while effective, did not match the performance of the best GNN and RNN models, indicating that capturing spatial and temporal dependencies is crucial for accurate earthquake nowcasting.

From a broader perspective, pre-trained FMs exhibited weaker performance compared to pattern models for two key reasons. First, the higher MSE and lower NNSE scores among FMs highlight the challenges these models face in transferring learned knowledge from broad or irrelevant pre-training datasets to the specific task of earthquake nowcasting. In contrast, pattern models, which focus on direct learning from earthquake data, consistently achieved lower MSEs and higher NNSEs, demonstrating their effectiveness in this domain.

Second, GNN models excel in understanding spatial relationships by performing specialized graph-based analyses that aggregate data from neighboring regions. In contrast, FMs like iTransformer attempt to derive spatial information from the entire dataset, including regions that may be irrelevant to a target area. This broad approach can introduce noise and hinder the model's ability to distinguish meaningful patterns and dependencies.

5.2.1. Earthquake Time Series Analysis

Earthquake time series analysis involves the investigation of temporal patterns within seismic activity data to forecast significant peaks, indicative of substantial energy release during large seismic events. This approach is vital for understanding and nowcasting the occurrence of earthquakes over time, enabling better preparedness and response strategies.

Figure 7 shows the released energy time series plots for six randomly selected spatial bins, providing additional insights into model performance. The plots compare the models' predictions, specifically GNNCoder 1-layer, DilatedRNN, TiDE, and iTransformer-M4, against the actual observed seismic activities over time. Notably, the GNNCoder and DilatedRNN models excel in capturing noticeable spikes in observed activity, illustrating their effectiveness in short-term earthquake nowcasting. This ability to anticipate imminent seismic events is critical for timely disaster preparedness and response, highlighting the practical applicability of these models in real-world scenarios.

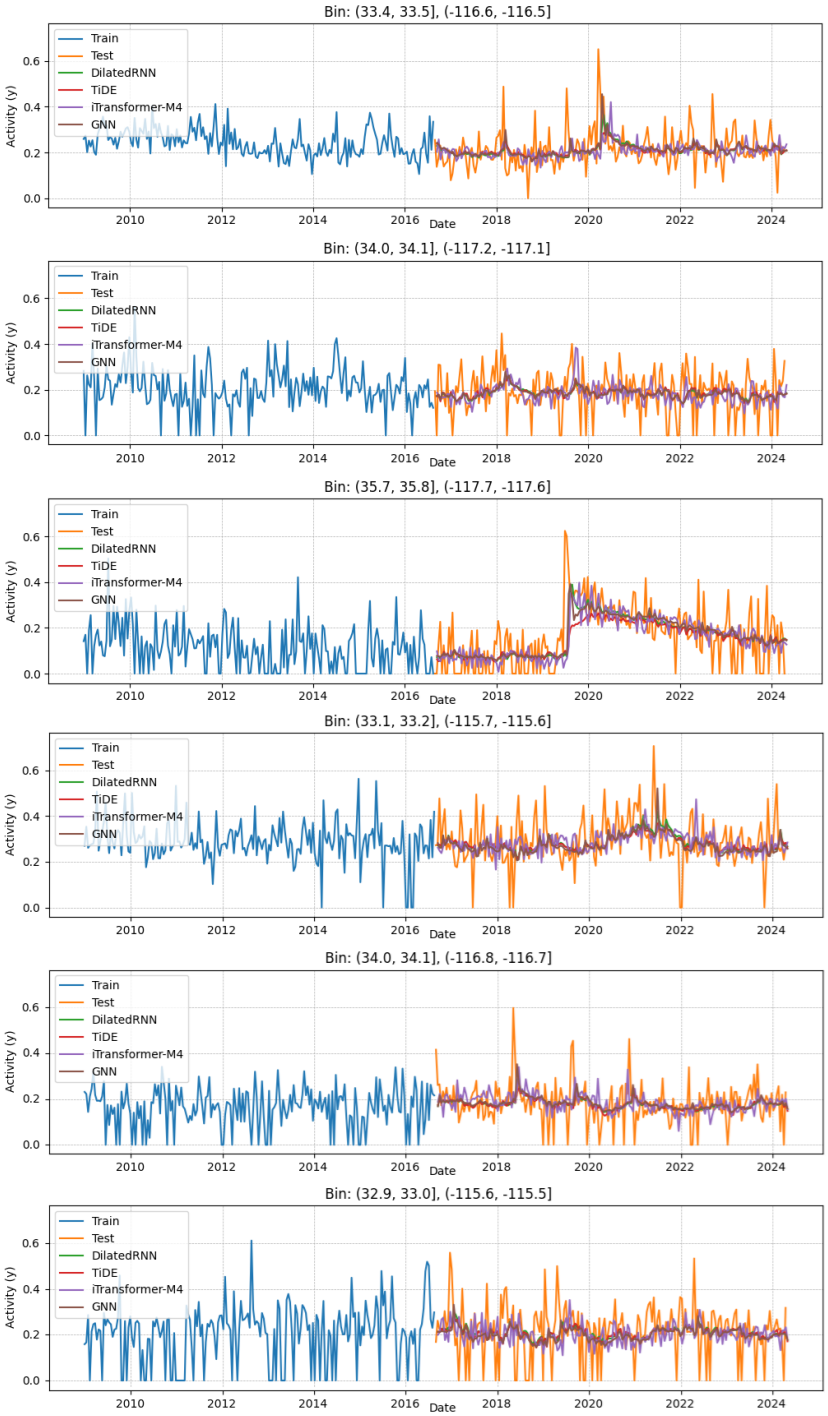

Figure 7. Released energy time series plots for six randomly selected spatial bins, comparing model predictions (GNNCoder one-layer, DilatedRNN, TiDE, iTransformer-M4) against actual observed seismic activities. The brown line represents our GNN approach, which shows a closer match with the actual time series, capturing crucial upward slopes that may signal an impending earthquake. The green and red lines occasionally miss these trends, making more errors where even slight changes in seismic activity are critical. The purple line from the iTransformer-M4 model fails to accurately capture the time series values and exhibits excessive fluctuations.

For instance, in the spatial bin (33.4, 33.5), (−116.6, −116.5), the GNN model's predictions closely follow the observed activity trends, while other models, like TiDE and iTransformer-M4, show more significant deviations. This pattern is observed across multiple bins, indicating the robustness of the GNNCoder in different spatial contexts.

5.2.2. Multi Foundation Quake Analysis

The performance of Multi Foundation Quake was further assessed by experimenting with different combinations of input models. Table 2 details the results of these experiments, illustrating the impact of integrating selected foundation models on the model's performance across key evaluation metrics. The models incorporated in the analysis include iTransformer-M4, PatchTST-TrafficL, iTransformer-TrafficL, TFT, LSTM, and DilatedRNN.

Multi Foundation Quake 1 and Multi Foundation Quake 2 utilize different approaches as the pattern component. While Multi Foundation Quake 2 employs a graph neural network to capture spatial information through a graph structure, Multi Foundation Quake 1 uses an LSTM to model temporal dependencies. This experiment focuses on Multi Foundation Quake 1, where the LSTM component processes inputs from the selected models to assess how well the model integrates information from these pre-trained models for the task of earthquake nowcasting.

As presented in Table 2, Multi Foundation Quake 1, incorporating outputs from all the selected models, achieved an MSE of 0.00627, an MAE of 0.0518, and an NNSE of 0.6171. Notably, Multi Foundation Quake 1 outperformed all individual input models, benefiting from the combined representation of temporal and spatial patterns, which significantly enhanced its accuracy in earthquake nowcasting.

Table 2. Performance evaluation of Multi Foundation Quake 1 and its variations with different input models. The table lists the performance metrics MSE, MAE, and NNSE for each model configuration. Earthquake time series features are always included as inputs to the model, while asterisks (*) indicate the specific input models used in each configuration.

Section	Model	MSE	MAE	NNSE	iTrans-M4	Patch-Traf	iTrans-Traf	TFT	LSTM	DilatedRNN
Section A: Individual Results of Input Models										
A	iTransformer-M4	0.00702	0.0537	0.5902	*					
	PatchTST-TrafficL	0.00644	0.0501	0.6107		*				
	iTransformer-TrafficL	0.00639	0.0513	0.6125			*			
	TFT	0.00635	0.0555	0.6142				*		
	LSTM	0.00631	0.0514	0.6156					*	
	DilatedRNN	0.00630	0.0510	0.6159						*
Section B: Systematic Evaluation of the Effect of Removing Lower-Performing Models										
B	Multi Foundation Quake 1	0.00627	0.0518	0.6171	*	*	*	*	*	*
	Multi Foundation Quake 1	0.00626	0.0516	0.6174		*	*	*	*	*
	Multi Foundation Quake 1	0.00627	0.0517	0.6172			*	*	*	*
	Multi Foundation Quake 1	0.00627	0.0515	0.6169				*	*	*
	Multi Foundation Quake 1	0.00627	0.0514	0.6171					*	*
	Multi Foundation Quake 1	0.00627	0.0515	0.6170						*
Section C: Systematic Evaluation of Each Input Model's Impact										
C	Multi Foundation Quake 1	0.00627	0.0518	0.6170	*	*	*	*	*	
	Multi Foundation Quake 1	0.00626	0.0515	0.6172	*	*	*	*		*
	Multi Foundation Quake 1	0.00627	0.0516	0.6171	*	*	*		*	*
	Multi Foundation Quake 1	0.00627	0.0518	0.6171	*	*		*	*	*
	Multi Foundation Quake 1	0.00627	0.0517	0.6170	*		*	*	*	*
	Multi Foundation Quake 1	0.00626	0.0516	0.6174		*	*	*	*	*

Further analysis revealed that excluding the least effective input model, iTransformer-M4, led to an improvement, resulting in an MSE of 0.00626, an MAE of 0.0516, and an NNSE of 0.6174. This suggests that the higher MSE of iTransformer-M4 (0.00702) may introduce noise or less relevant information, thereby marginally detracting from the overall predictive performance.

In contrast, removing the best-performing model, DilatedRNN, resulted in an MSE of 0.00627, an MAE of 0.0518, and an NNSE of 0.6170. This indicates the critical importance of

effective model contributions, as the DilatedRNN outputs were closely aligned with the target time series.

5.2.3. Spatial Analysis

To assess the impact of increasing the depth of the GAT layers on performance, we evaluated GNNCoder models with different numbers of layers. Table 3 demonstrates the importance of considering spatial relationships, where the GNNCoder 1-layer model outperforms the GNNCoder 3-layer model across multiple metrics. This outcome may seem counterintuitive at first, as one might expect a deeper model to capture more intricate patterns and dependencies within the data. However, as shown in Figure 8, our approach necessitated the creation of a graph based on spatial bins. These spatial bins serve as nodes in constructing the graph, which unfortunately cannot encompass all parts of fault lines. Consequently, there are bins containing crucial fault information that our graph failed to consider.

Table 3. Comparison of the performance of GNNCoder using different numbers of GAT layers, ranked by MSE in descending order.

Model	Layer	MSE	MAE	NNSE
GNNCoder	2-layer	0.00632	0.0520	0.6153
GNNCoder	3-layer	0.00629	0.0524	0.6162
GNNCoder	1-layer	0.00628	0.0522	0.6166

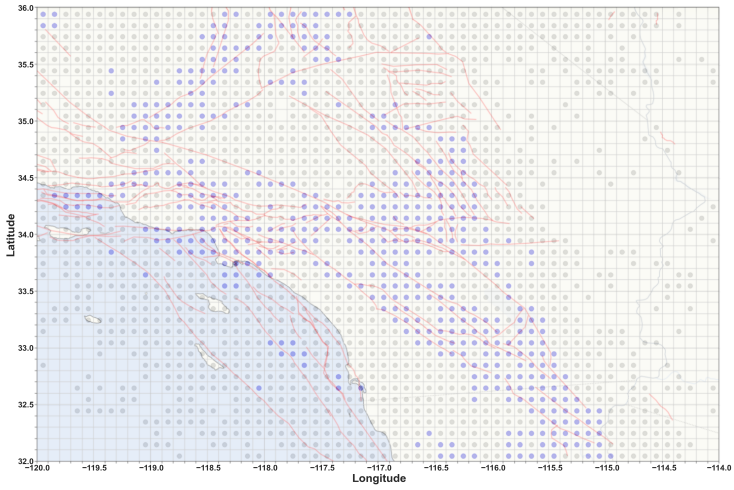

Figure 8. This plot illustrates the spatial bins overlaid on the fault lines to assess the extent to which the fault lines are captured by the bins (graph nodes). It highlights the limitations of the current graph, where some critical fault lines fall outside the spatial bins, impacting the performance of deeper GNN models like the GNNCoder 3-layer model.

The limitation of the graph construction method is particularly detrimental to deeper GNN models, such as the GNNCoder 3-layer model. Deeper models typically rely on the aggregation of information across multiple layers, which can amplify the impact of missing or incomplete data within the graph structure. In this case, the spatial bins that were not included in the graph represent significant gaps in the fault information, hindering the GNNCoder 3-layer model's ability to fully leverage its depth. As a result, the GNNCoder 3-layer model may struggle to effectively capture the underlying patterns of the data, leading to slightly poorer performance compared to the GNNCoder 1-layer model.

In contrast, the GNNCoder 1-layer model, being shallower, is less affected by the incomplete graph representation. Its simpler structure allows it to focus on more immediate, localized relationships within the spatial bins that are included in the graph. This enables the GNNCoder 1-layer model to perform better despite the limitations of the graph construction. Therefore, the results highlight the importance of considering the quality and

completeness of the graph structure when designing GNN models for this type of data. A more comprehensive and accurate graph that covers all relevant fault lines might enable deeper GNNCoder models to outperform their shallower counterparts. For future work, constructing a graph structure based on alternative relationships, such as connections along fault lines, holds promise. We are actively exploring this approach and plan to discuss the results in upcoming research.

On the other hand, large multivariate transformer-based models, such as PatchTST and iTransformer, utilize channel-independent methods to aggregate input data. In our experiments, the input data for all models consisted of time series from different regions. We expected that these multivariate models would effectively capture the spatial relationships between these time series from our extensive dataset. However, the results reveal that multivariate transformer-based models struggle to accurately capture the latent spatial relationships essential for precise earthquake nowcasting. In contrast, GNNs are particularly adept at understanding spatial dependencies through graph-based analyses that effectively aggregate data from neighboring regions. This localized approach allows GNNs to model the spatial intricacies crucial for accurate earthquake nowcasting.

5.2.4. Feature Analysis

Input selection and feature engineering are fundamental components in the development of effective deep-learning models. The identification and utilization of relevant features are pivotal in any field to the ability to learn and make accurate model predictions. This is particularly true in the field of earthquake nowcasting, where the incorporation of geographical interactions can significantly enhance model performance.

A notable limitation of pre-trained models is the requirement to maintain a consistent input structure during both the pre-training and fine-tuning phases. This restriction hinders the integration of sophisticated domain-specific features that could potentially improve nowcasting performance on earthquake data.

Table 4 explores the effects of different input configurations, using Multiplicity and the Exponential Moving Average (EMA) as inputs, to compare the performance of the top models, GNNCoder, DilatedRNN, and LSTM.

As explained in our papers [2,43], Multiplicity is a critical factor in enhancing the nowcasting accuracy of earthquake nowcasting. Multiplicity refers to the count of earthquake events within a defined spatial-temporal bin that exceeds a specific magnitude threshold, capturing the frequency and intensity of seismic activity and providing a straightforward measure of earthquake occurrence rates. In this study, the magnitude threshold is set at 3.29, and five intervals, ranging from 2 weeks to 260 weeks, are used to calculate Multiplicity. In addition, we employ EMA, which averages the past 5 to 150 samples, further refining the input data.

Table 4. Performance comparison of GNN and DilatedRNN models using various input configurations. The table highlights the impact of incorporating Multiplicity and EMA features on the models' nowcasting accuracy.

Model	Input	MSE	MAE	NNSE
LSTM	Single feature	0.00631	0.0514	0.6156
LSTM	+ Multiplicity	0.00630	0.0506	0.6158
DilatedRNN	Single feature	0.00630	0.0510	0.6159
LSTM	+ Multiplicity + EMA	0.00629	0.0527	0.6162
LSTM	+ EMA	0.00628	0.0517	0.6164
GNNCoder 1-layer	+ Multiplicity	0.00628	0.0520	0.6165
GNNCoder 1-layer	Single feature	0.00628	0.0522	0.6166
GNNCoder 1-layer	+ Multiplicity + EMA	0.00627	0.0517	0.6169
DilatedRNN	+ Multiplicity	0.00627	0.0517	0.6169
GNNCoder 1-layer	+ EMA	0.00627	0.0525	0.6172
DilatedRNN	+ EMA	0.00627	0.0519	0.6174
DilatedRNN	+ Multiplicity + EMA	0.00626	0.0517	0.6174

As shown in Table 4, incorporating relevant features enhanced the performance of all three models. The DilatedRNN model with multiplicity and EMA inputs demonstrated the best performance, achieving an MSE of 0.00626, an MAE of 0.0517, and an NNSE of 0.6174. This suggests that the addition of multiplicity and EMA features significantly enhances the DilatedRNN's ability to accurately model sequential patterns in the data.

The GNNCoder with multiplicity and EMA inputs also performed well, with an MSE of 0.00627, an MAE of 0.0517, and an NNSE of 0.6169. Despite its strong performance, the DilatedRNN outperformed the GNNCoder, indicating that the integration of temporal features is particularly beneficial for the memory-based DilatedRNN, enhancing its capacity to capture the temporal complexities of seismic data more effectively.

Additionally, Table 4 highlights a significant limitation of foundation models, which are constrained to managing only the target time series and cannot accept relevant features.

6. Conclusions

This study presents a comprehensive evaluation of foundation models and advanced deep learning architectures for earthquake nowcasting, focusing on Southern California's seismically active region. Our study demonstrates that the selection of appropriate model architectures and pre-training datasets plays a critical role in enhancing nowcasting accuracy for seismic activities.

Our analysis shows that the introduced Multi Foundation Quake model outperforms other models by leveraging the strengths of various foundation models, effectively capturing both temporal and spatial dependencies in seismic data. This model showcases the potential of combining diverse pre-trained models to improve earthquake nowcasting, emphasizing the importance of multi-model integration.

Additionally, the GNNCoder model outperforms other models by effectively capturing spatial relationships and leveraging geographical interactions, resulting in more accurate earthquake nowcasting. Memory-based models like DilatedRNN and LSTM also show strong performance in handling sequential dependencies, though their accuracy could be further improved by incorporating spatial information. The integration of spatial and temporal features is crucial for enhancing nowcasting accuracy.

A notable issue identified in this study is that large multivariate transformer-based models were not successful in capturing the latent spatial relationships between neighboring areas. Future research should focus on enhancing channel-independent methods in multivariate models to better capture subtle spatial dependencies.

In addition, the performance of pre-trained foundation models varied significantly depending on the pre-training datasets. While models pre-trained on datasets with some temporal and spatial similarities to seismic data, such as iTransformer-TrafficL and PatchTST-Weather, performed relatively well, the unique and complex patterns inherent to earthquake data presented challenges. Although the Weather and TrafficL datasets share certain characteristics with seismic patterns, earthquake occurrences exhibit distinctive temporal and spatial behaviors that may not be captured fully by these general datasets. Consequently, pre-training can sometimes introduce distractions that hinder the model's ability to focus on earthquake-specific features. This finding highlights the importance of carefully selecting or designing pre-training datasets that closely align with seismic data to ensure effective knowledge transfer. In cases where pre-training datasets diverge from the domain's unique patterns, alternative strategies, such as custom-designed loss functions that mitigate irrelevant knowledge transfer, may be more beneficial. Such approaches could allow pre-training and fine-tuning datasets to be incorporated synergistically, enhancing the models' ability to learn earthquake-specific patterns more effectively.

Given the inherent unpredictability and complexity of seismic activity, even minor improvements in model performance are significant. Although the differences in evaluation metrics may appear small due to the normalization of the dataset and the rarity of large seismic events, these incremental gains can meaningfully enhance early warning systems. By consistently observing performance trends across MSE, MAE, and NNSE metrics and

various model configurations, we have ensured the reliability and generalizability of our findings. These improvements, however incremental, represent important steps toward better forecasting high-risk events, potentially saving lives and reducing economic losses. In future work, we plan to further study the significance of these improvements.

Future research directions include the development of hybrid models that integrate the strengths of GNNs and RNNs, leveraging both spatial and temporal information to enhance nowcasting capabilities. Improving graph construction methods will also be crucial to better capture the complexities of seismic data. Our study underscores the importance of feature engineering and input selection; incorporating scientific features like Multiplicity and EMA significantly improved model performance. Additionally, integrating more diverse data sources, such as physics equations, could provide a more comprehensive understanding of seismic patterns and further enhance nowcasting accuracy.

Several important geoscience issues should be explored together with the AI topics listed above. These include the spatial extent of the earthquake, which is particularly important as AI models, including spatial links, performed best in this initial study. We will also look at nowcasting over different time periods as in our earlier paper [2], which looked 4 years into the future from a 2-week time series. We will also try to quantify the origin of the nowcasting accuracy by applying these ideas to simulated ERAS [44] and ETAS earthquakes [39–41]. We will also explore other geographical regions and different time periods.

This research advances the state of the art in earthquake nowcasting by demonstrating the efficacy of GNNs and pre-trained transformer models. Our improved accuracy and reliability have the potential to enhance disaster response efforts, minimize economic losses, and save lives by providing timely and precise nowcasting of seismic events. This research represents a significant step towards bridging the gap between advanced deep learning methodologies and practical applications in understanding earthquake occurrence and mitigation.

Author Contributions: Conceptualization, A.J., G.F. and J.B.R.; methodology, all; software, A.J., G.F. and J.B.R.; validation, all; data curation, A.J., G.F. and J.B.R.; writing—original draft preparation, A.J. and G.F.; writing— review and editing, all; visualization, all. All authors have read and agreed to the published version of the manuscript.

Funding: The University of Virginia authors gratefully acknowledge the partial support of DE-SC0023452: FAIR Surrogate Benchmarks Supporting AI and Simulation Research and the Biocomplexity Institute at the University of Virginia. Research by JBR was supported in part under DoE grant DE-SC0017324 to the University of California, Davis. Portions of this work were carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract with NASA. Research by LGL was supported in part by the University of California Irvine Academic Senate award.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code and dataset are available at: https://doi.org/10.5281/zenodo.13358007, (accessed on 22 August 2024). The earthquake data used were sourced from the US Geological Survey (USGS) online catalogs [45].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jordan, T.H.; Chen, Y.T.; Gasparini, P.; Madariaga, R.; Main, I.; Marzocchi, W.; Papadopoulos, G.; Sobolev, G.; Yamaoka, K.; Zschau, J. Operational earthquake forecasting: State of knowledge and guidelines for utilization. *Ann. Geophys.* **2011**, *54*, 315–391.
2. Fox, G.C.; Rundle, J.B.; Donnellan, A.; Feng, B. Earthquake Nowcasting with Deep Learning. *GeoHazards* **2022**, *3*, 199–226. [CrossRef]
3. Rundle, J.B.; Donnellan, A.; Fox, G.; Crutchfield, J.P.; Granat, R. Nowcasting Earthquakes: Imaging the Earthquake Cycle in California With Machine Learning. *Earth Space Sci.* **2021**, *8*, e2021EA001757. [CrossRef]
4. de Arcangelis, L.; Godano, C.; Grasso, J.R.; Lippiello, E. Statistical physics approach to earthquake occurrence and forecasting. *Phys. Rep.* **2016**, *628*, 1–91. [CrossRef]

5. Chuang, R.Y.; Wu, B.S.; Liu, H.C.; Huang, H.H.; Lu, C.H. Development of a statistics-based nowcasting model for earthquake-triggered landslides in Taiwan. *Eng. Geol.* **2021**, *289*, 106177. [CrossRef]
6. Rundle, J.B.; Donnellan, A. Nowcasting Earthquakes in Southern California With Machine Learning: Bursts, Swarms, and Aftershocks May Be Related to Levels of Regional Tectonic Stress. *Earth Space Sci.* **2020**, *7*, e2020EA001097. [CrossRef]
7. Mousavi, S.M.; Ellsworth, W.L.; Zhu, W.; Chuang, L.Y.; Beroza, G.C. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.* **2020**, *11*, 3952. [CrossRef]
8. Rundle, J.B.; Donnellan, A.; Fox, G.; Crutchfield, J.P. Nowcasting earthquakes by visualizing the earthquake cycle with machine learning: A comparison of two methods. *Surv. Geophys.* **2022**, *43*, 483–501. [CrossRef]
9. Perol, T.; Gharbi, M.; Denolle, M. Convolutional neural network for earthquake detection and location. *Sci. Adv.* **2018**, *4*, e1700578. [CrossRef]
10. Harirchian, E.; Lahmer, T.; Rasulzade, S. Earthquake hazard safety assessment of existing buildings using optimized multi-layer perceptron neural network. *Energies* **2020**, *13*, 2060. [CrossRef]
11. Jafari, A.; Haratizadeh, S. GCNET: Graph-based prediction of stock price movement using graph convolutional network. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105452. [CrossRef]
12. Jafari, A.; Haratizadeh, S. NETpred: Network-based modeling and prediction of multiple connected market indices. *arXiv* **2022**, arXiv:2212.05916.
13. Shariatmadari, A.H.; Guo, S.; Srinivasan, S.; Zhang, A. Harnessing the Power of Knowledge Graphs to Enhance LLM Explainability in the BioMedical Domain. Proceedings of the LLMs4Bio Workshop at AAAI 2024, 2024; pp. 1–8. Available online: https://llms4science-community.github.io/aaai2024/papers/LLMs4Bio24_paper_10.pdf (accessed on 10 May 2024).
14. Zhang, X.; Reichard-Flynn, W.; Zhang, M.; Hirn, M.; Lin, Y. Spatiotemporal Graph Convolutional Networks for Earthquake Source Characterization. *J. Geophys. Res. Solid Earth* **2022**, *127*, e2022JB024401. [CrossRef] [PubMed]
15. Bilal, M.A.; Ji, Y.; Wang, Y.; Akhter, M.P.; Yaqub, M. An Early Warning System for Earthquake Prediction from Seismic Data Using Batch Normalized Graph Convolutional Neural Network with Attention Mechanism (BNGCNNATT). *Sensors* **2022**, *22*, 6482. [CrossRef] [PubMed]
16. McBrearty, I.W.; Beroza, G.C. Earthquake location and magnitude estimation with graph neural networks. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 3858–3862.
17. McBrearty, I.W.; Beroza, G.C. Earthquake phase association with graph neural networks. *Bull. Seismol. Soc. Am.* **2023**, *113*, 524–547. [CrossRef]
18. van den Ende, M.P.A.; Ampuero, J.P. Automated Seismic Source Characterization Using Deep Graph Neural Networks. *Geophys. Res. Lett.* **2020**, *47*, e2020GL088690. [CrossRef]
19. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
20. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. arXiv **2017**, arXiv.:1706.03762 [CrossRef]
22. Sadhukhan, B.; Chakraborty, S.; Mukherjee, S. Predicting the magnitude of an impending earthquake using deep learning techniques. *Earth Sci. Inform.* **2023**, *16*, 803–823. [CrossRef]
23. Saad, O.M.; Chen, Y.; Savvaidis, A.; Fomel, S.; Chen, Y. Real-time earthquake detection and magnitude estimation using vision transformer. *J. Geophys. Res. Solid Earth* **2022**, *127*, e2021JB023657. [CrossRef]
24. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
25. ScienceFMHub Portal for Science Foundation Model Community. 2023. Available online: http://sciencefmhub.org (accessed on 3 November 2023).
26. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv* **2023**, arXiv:2310.06625.
27. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv* **2023**, arXiv:2211.14730. http://arxiv.org/abs/2211.14730.
28. Garza, A.; Mergenthaler-Canseco, M. TimeGPT-1. *arXiv* **2023**, arXiv:2310.03589.
29. Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J.Y.; Shi, X.; Chen, P.Y.; Liang, Y.; Li, Y.F.; Pan, S.; et al. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. *arXiv* **2023**, arXiv:2310.01728. http://arxiv.org/abs/2310.01728.
30. Ansari, A.F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S.S.; Arango, S.P.; Kapoor, S.; et al. Chronos: Learning the language of time series. *arXiv* **2024**, arXiv:2403.07815.
31. Chen, S.A.; Li, C.L.; Yoder, N.; Arik, S.O.; Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv* **2023**, arXiv:2303.06053.
32. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
33. Chang, S.; Zhang, Y.; Han, W.; Yu, M.; Guo, X.; Tan, W.; Cui, X.; Witbrock, M.; Hasegawa-Johnson, M.A.; Huang, T.S. Dilated recurrent neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. https://arxiv.org/abs/1710.02224.

34. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
35. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
36. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. *arXiv* **2023**, arXiv:2210.02186. http://arxiv.org/abs/2210.02186.
37. Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv* **2023**, arXiv:2304.08424.
38. Holschneider, M.; Zöller, G.; Clements, R.; Schorlemmer, D. Can we test for the maximum possible earthquake magnitude? *J. Geophys. Res. Solid Earth* **2014**, *119*, 2019–2028. [CrossRef]
39. Zhuang, J. Long-term earthquake forecasts based on the epidemic-type aftershock sequence (ETAS) model for short-term clustering. *Res. Geophys.* **2012**, *2*, e8. [CrossRef]
40. Field, E.H.; Milner, K.R.; Hardebeck, J.L.; Page, M.T.; van der Elst, N.; Jordan, T.H.; Michael, A.J.; Shaw, B.E.; Werner, M.J. A Spatiotemporal Clustering Model for the Third Uniform California Earthquake Rupture Forecast (UCERF3-ETAS): Toward an Operational Earthquake Forecast. *Bull. Seismol. Soc. Am.* **2017**, *107*, 1049–1081. [CrossRef]
41. Rundle, J.B.; Baughman, I.; Zhang, T. Nowcasting ETAS Earthquakes: Information Entropy of Earthquake Catalogs. *arXiv* **2023**, arXiv:2310.14083. [CrossRef]
42. Rundle, J.B.; Fox, G.; Donnellan, A.; Ludwig, L.G. Nowcasting earthquakes with QuakeGPT: Methods and first results. *arXiv* **2024**, arXiv:2406.09471.
43. Rundle, J.B.; Yazbeck, J.; Donnellan, A.; Fox, G.; Ludwig, L.G.; Heflin, M.; Crutchfield, J. Optimizing Earthquake Nowcasting With Machine Learning: The Role of Strain Hardening in the Earthquake Cycle. *Earth Space Sci.* **2022**, *9*, e2022EA002343. [CrossRef]
44. Rundle, J.B.; Baughman, I.; Zhang, T. Nowcasting earthquakes with stochastic simulations: Information entropy of earthquake catalogs. *Earth Space Sci.* **2024**, *11*, e2023EA003367. [CrossRef]
45. of United States Geological Survey, E.H.P. USGS Search Earthquake Catalog Home Page. Available online: https://earthquake.usgs.gov/earthquakes/search/ (accessed on 1 May 2024).
46. Field, E.H. Overview of the Working Group for the Development of Regional Earthquake Likelihood Models (RELM). *Seismol. Res. Lett.* **2007**, *78*, 7–16. [CrossRef]
47. Scholz, C.H. *The Mechanics of Earthquakes and Faulting*; Cambridge University Press: Cambridge, UK, 2019.
48. Hanks, T.C.; Kanamori, H. A moment magnitude scale. *J. Geophys. Res. Solid Earth* **1979**, *84*, 2348–2350. [CrossRef]
49. Eppstein, D.; Paterson, M.S.; Yao, F.F. On nearest-neighbor graphs. *Discret. Comput. Geom.* **1997**, *17*, 263–282. [CrossRef]
50. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
51. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA. 8–12 July 2018; pp. 95–104.
52. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *Int. J. Forecast.* **2020**, *36*, 54–74. [CrossRef]
53. Haugsdal, E.; Aune, E.; Ruocco, M. Persistence initialization: A novel adaptation of the transformer architecture for time series forecasting. *Appl. Intell.* **2023**, *53*, 26781–26796. [CrossRef]
54. Oreshkin, B.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2019**, arXiv:1905.10437.
55. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; NIPS'17, pp. 5998–6008.
56. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
57. Wang, X.; Ji, H.; Shi, C.; Wang, B.; Wang, P.; Cui, P.; Yu, P.S. Heterogeneous graph attention network. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2022–2032.
58. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; JMLR Workshop and Conference Proceedings; pp. 249–256.
59. Olivares, K.G.; Challú, C.; Garza, F.; Canseco, M.M.; Dubrawski, A. *NeuralForecast: User Friendly State-of-the-Art Neural Forecasting Models*; PyCon: Salt Lake City, UT, USA, 2022.
60. Grattarola, D.; Alippi, C. Graph neural networks in TensorFlow and keras with spektral [application notes]. *IEEE Comput. Intell. Mag.* **2021**, *16*, 99–106. [CrossRef]
61. MLCommons. MLCommons Homepage: Machine Learning Innovation to Benefit Everyone. Available online: https://mlcommons.org/en/ (accessed on 7 December 2021).

62. von Laszewski, G.; Fleischer, J.P.; Knuuti, R.; Fox, G.C.; Kolessar, J.; Butler, T.S.; Fox, J. Opportunities for enhancing MLCommons efforts while leveraging insights from educational MLCommons earthquake benchmarks efforts. *Front. High Perform. Comput.* **2023**, *1*. Available online: https://par.nsf.gov/biblio/10473591 (accessed on 23 October 2023). [CrossRef]
63. Thiyagalingam, J.; von Laszewski, G.; Yin, J.; Emani, M.; Papay, J.; Barrett, G.; Luszczek, P.; Tsaris, A.; Kirkpatrick, C.; Wang, F.; et al. AI Benchmarking for Science: Efforts from the MLCommons Science Working Group. In Proceedings of the HPC on Heterogeneous Hardware (H3) Workshop at ISC Conference, Hamburg, Germany, 25 May 2023.
64. Group, M.S.W. MLCommons Science Working Group Invites Researchers to Run New Benchmarks. Available online: https://www.hpcwire.com/off-the-wire/mlcommons-science-working-group-invites-researchers-to-run-new-benchmarks/ (accessed on 4 September 2023).
65. MLCommons Science Working Group. MLCommons Science Working Group GitHub for Benchmarks. 2022. Available online: https://github.com/mlcommons/science (accessed on 27 December 2012).
66. Nossent, J.; Bauwens, W. Application of a normalized Nash-Sutcliffe efficiency to improve the accuracy of the Sobol'sensitivity analysis of a hydrological model. In Proceedings of the EGU General Assembly Conference Abstracts, Vienna, Austria, 22–27 April 2012; p. 237.