SeisLM: a Foundation Model for Seismic Waveforms

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce the Seismic Language Model (SeisLM), a foundational model designed to analyze seismic waveforms—signals generated by Earth's vibrations such as the ones originating from earthquakes. SeisLM is pretrained on a large collection of open-source seismic datasets using a self-supervised contrastive loss, akin to BERT in language modeling. This approach allows the model to learn general seismic waveform patterns from unlabeled data without being tied to specific downstream tasks. When fine-tuned, SeisLM excels in seismological tasks like event detection, phase-picking, onset time regression, and foreshock–aftershock classification.

1 Introduction

Seismology is a data-centric field that often sees significant progress through improvements in data quality and quantity (Havskov & Ottemoller, 2010; Zhou, 2014). Today, the field benefits from an extensive collection of seismic recordings gathered over years by networks of thousands of stations worldwide (Hafner & Clayton, 2001; Mousavi et al., 2019a; Quinteros et al., 2021; Michelini et al., 2021; Cole et al., 2023; Niksejel & Zhang, 2024; Chen et al., 2024; Zhong & Tan, 2024). Over the last decades, millions of these recordings have been manually inspected and labeled by domain experts. This wealth of data and labels has fueled the rise of machine-learning models, which automate the analysis of these expanding seismic records. A growing body of models, including convolutional networks (Ross et al., 2018; Zhu & Beroza, 2018; Woollam et al., 2019; Mousavi et al., 2019c), recurrent networks (Soto & Schurr, 2021; Yoma et al., 2022), and transformers (Mousavi et al., 2020; Li et al., 2024; Münchmeyer et al., 2021) have been applied to seismic data analysis, particularly in tasks like earthquake detection and characterization.

Despite these advances, most current machine-learning models in seismology still depend on *labeled, task-specific datasets*, not making use of more than a petabyte of openly available unlabeled waveforms. This mirrors the early stages of machine learning in fields like computer vision and natural language processing, where models were initially trained on similarly specialized datasets such as MNIST (Lecun et al., 1998), CIFAR (Krizhevsky & Hinton, 2009), Sentiment140 (Go et al., 2009), and IMDB dataset (Maas et al., 2011). Yet, these task-specific models eventually gave way to general-purpose foundation models, trained on a wealth of unlabled data, which are capable of handling a broader range of tasks with minimal fine-tuning. Exemplars of open-weight foundation models include BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and Llama (Touvron et al., 2023a,b; Dubey et al., 2024) for text processing, Wav2Vec2 (Baevski et al., 2020) and Hubert (Hsu et al., 2021) for speech understanding, and CLIP (Radford et al., 2021) and MAE (He et al., 2022) for vision modeling. These foundation models rely on *self-supervised learning* from unlabeled data, allowing them to scale up training samples and learn features without being tied to specific tasks.

In this work, we introduce the Seismic Language Model (SeisLM), a self-supervised model for analyzing single-station seismic waveforms. SeisLM uses a standard encoder-only transformer

Submitted to Foundation Models for Science Workshop, 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

architecture, similar to Wav2Vec2 and BERT. Our results demonstrate that this model, when pretrained on worldwide earthquake activity records, extracts generalizable features that effectively address various downstream tasks, nearly always surpassing models tailored for specific tasks. The main contributions of the paper are summarized below:

- We introduce a self-supervised foundation model for seismic waveforms. To our knowledge, it represents the first application of self-supervised learning on unlabeled seismic waveforms.
- We demonstrate that the model's self-supervised features, although not trained on any labeled samples, display clear and interpretable characteristics. Specifically, the model groups waveform features into noise and earthquake clusters.
- We show that the self-supervised model generalizes well to a wide array of downstream tasks. When compared with supervised baselines, the advantage of pretraining–finetuning is particularly noticeable when the downstream tasks have limited labeled data.

2 Background and related work

Supervised-learning models for seismic tasks. The efforts of using supervised machine learning to automate seismic waveform analysis stretch back several decades. We briefly review a non-exhaustive selection of neural network approaches. Early methods used shallow multilayer perceptrons (MLPs) to classify seismic waveforms (Enescu et al., 1996; Baevski et al., 2020; Dai & MacBeth, 1997; Zhao & Takano, 1999; Gentili & Michelini, 2006). Starting from 2010s, 1D convolutional neural networks (ConvNets) have been prevalent in seismic applications due to their efficiency and flexibility in handling variable-length input. For instance, the Generalized Phase Detection model (Ross et al., 2018) uses a 1D convolutional network for phase classification tasks. Inspired by the U-Net (Ronneberger et al., 2015), a convolutional network originally designed for 2D image segmentation, Zhu & Beroza (2018); Woollam et al. (2019) used similar architectures in 1D for onset and phase picking tasks. Mousavi et al. (2019c) proposed a residual convolutional network for earthquake detection, drawing on ideas from residual networks used in image classification (He et al., 2016). In addition to ConvNets, recurrent networks (RNNs) have also been applied to seismic tasks. These networks include DeepPhasePick (Soto & Schurr, 2021), which handles event detection and phase picking. Finally, the recent success of transformers and their self-attention mechanisms (Vaswani et al., 2017) has inspired their use in seismic analysis. The Earthquake Transformer (Mousavi et al., 2020) combines recurrent networks and self-attention mechanisms for joint event detection, phase detection, and onset picking. While Earthquake Transformer is a Transformer–CNN–RNN hybrid approach, Seismogram transformer (Li et al., 2024) shows that a plain transformer can be used to solve different earthquake-monitoring tasks when coupled with different head modules.

Unsupervised learning models for seismic tasks. Unsupervised machine learning has been used to uncover patterns in unlabeled seismic data, primarily through clustering and visualization. Esposito et al. (2008) cluster volcanic event waveforms to explore the link between active volcanic vents and their explosive waveforms. Yoon et al. (2015) group waveforms with similar features in a database, then use a search method to identify those resembling earthquake signals. Mousavi et al. (2019b) use convolutional autoencoders to cluster and differentiate hypocentral distances and first-motion polarities. Seydoux et al. (2020) combine scattering networks with a Gaussian mixture model to cluster seismic signal segments, demonstrating applications in blind detection and recovery of repeating precursory seismicity.

Foundation models for seismic tasks and their relationships to our work. There exist a few foundation models for seismic applications, although they differ from our approach in several aspects. Sheng et al. (2023) proposed a foundation model for *seismic imagery data*, which are visual representations of the Earth's subsurface structures. These images are generated by seismic waves reflecting off rock boundaries, capturing the differences in physical properties between various geological layers. In contrast, our work focuses on seismic waveforms, which are time-series data. In this regard, the closest related models are Si et al. (2024) and Li et al. (2024), which also handle seismic waveforms. Both, however, rely on labeled datasets for pretraining. Specifically, Si et al. (2024) uses event annotations, such as phase and source information, for a contrastive approach. Li et al. (2024) uses a *supervised pretraining* method, training a single model for various classification and regression tasks, including earthquake detection and phase picking, using labeled data. Our approach is distinct in that we use only *unlabeled waveforms* for pretraining. This is motivated by the consideration that unlabeled waveforms are much more accessible and abundant than labeled ones. To our knowledge, SeisLM is the first foundation model self-supervisedly trained on unlabeled seismic waveforms.

3 Seismic Language Model

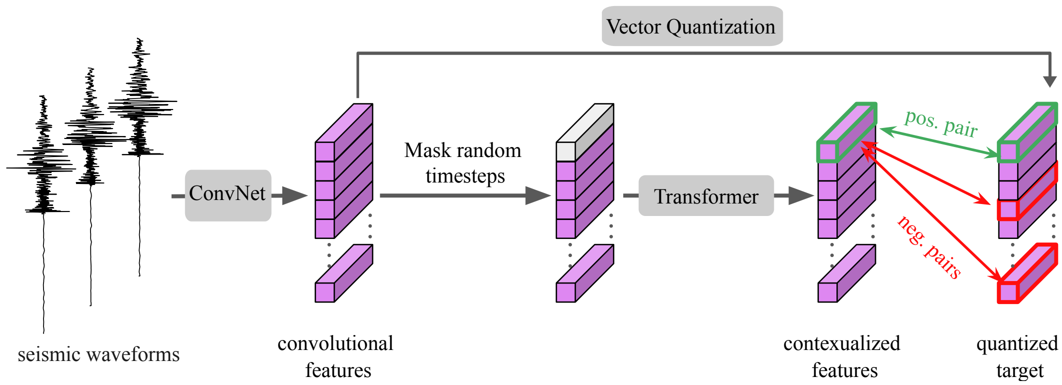

Figure 1: **Illustration of the self-supervised learning of Seismic Language Model (SeisLM).** A ConvNet encodes raw 3-channel seismic waveforms from a single station into a feature sequence. The model then follows two paths. In the lower path, we apply random masking to these waveform features before passing them to a transformer. The transformer aims to reconstruct aspects of the masked convolutional features. In the upper path, we prepare the reconstruction targets: continuous-valued convolutional features are discretized into a sequence of vectors with a finite vocabulary size through vector quantization (VQ; Van Den Oord et al., 2017; Razavi et al., 2019). This overall model closely resembles Wav2vec2 (Baevski et al., 2020) for audio self-supervised learning.

Our language model is an encoder-only transformer that focuses on the task of predicting features of masked timesteps. This model architecture is standard, closely following Wav2Vec2 (Baevski et al., 2020) for speech signal modeling and BERT (Devlin et al., 2019) for text modeling. In Fig. 1, we show a general overview of the model, which consists of a ConvNet, a quantizer, and a transformer. We now explain the role of each module and defer their detailed hyperparameters to Section 5.

3.1 SeisLM architecture

Model input. The input to the model are raw seismic waveforms, which are a sequence of vectors $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$; each sample $\boldsymbol{x}_t \in \mathbb{R}^3$ has three channels that correspond to ground motion recorded by a single seismometer for three orthogonal axes: East–West, North–South, and Up–Down; this format is standard in seismic data. Most seismic datasets use a sampling rate of 100 Hz or of the same order of magnitude (see Table 1); we thus use waveforms at 100 Hz for consistency and resample the waveform to 100Hz in case the original sampling rate differs. We standardize each channel of a waveform by subtracting the channel mean and dividing by the channel standard deviation.

ConvNet encoder. The raw waveforms $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ first undergo a 1D ConvNet, yielding convolutional features $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L)$ with $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$. The purpose of the 1D ConvNet is twofold: (i) filter the raw waveform and lift the 3-dimensional waveform signals to a higher dimension ($d_v > 3$), and (ii) downsample the sequence of the raw waveform in length ($L < T$), so that self-attention layers can be applied to this shorter sequence with lower computational complexity.

Transformer encoder. The convolutional features are then fed into a sequence of transformer blocks (Vaswani et al., 2017) after masking and position embedding. The masking part replaces convolutional features at random timesteps by a fixed embedding vector (details in Section 4.1). For position embedding, as in Baevski et al. (2020), we apply a 1D group convolutional layer (Krizhevsky et al., 2012) with a large kernel to obtain relative positional embedding, and then sum the output with the masked features. The position-embedded masked features are then fed to the transformer. The transformer is the heart of the model, as its self-attention mechanism (Vaswani et al., 2017) captures contextual information. We write the transformer output as $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$ with $\boldsymbol{a}_t \in \mathbb{R}^{d_q}$.

Quantization. During pretraining, the transformer encoder aims to reconstruct the unmasked convolutional seismic features from their masked corruptions. We use *quantized* convolutional features as the reconstruction targets: Given an input $\boldsymbol{v}_t \in \mathbb{R}^{d_v}$ of the raw waveform, the quantization module (Jegou et al., 2010) intuitively retrieves the nearest neighbor of $\boldsymbol{v}_t$ over a finite codebook $\mathcal{Q} := \{\boldsymbol{q}_{(1)}, \ldots, \boldsymbol{q}_{(n_q)}\} \subset \mathbb{R}^{d_q}$ and use the resulting vector as the target; the parenthesized indices here refer to the enumeration of the code vectors, which differs from the unparenthesized ones used to denote timesteps. Using quantized waveforms as the target proved more effective than non-quantized waveforms in previous speech self-supervised learning research (Baevski et al., 2020, 2019). Baevski et al. (2020) suggested that quantization reduces specific artifacts, such as speaker and background noise, which simplifies the reconstruction task and prevents the model from fitting on irrelevant details. To obtain the quantized vectors, a quantization module $Q : \mathbb{R}^{d_v} \to \mathcal{Q}$ is applied to the feature vector at each timestep independently with $\boldsymbol{q}_t := Q(\boldsymbol{v}_t)$. To parameterize the quantization function Q, we follow Jegou et al. (2010) and use learnable matrices $\boldsymbol{W} \in \mathbb{R}^{n_q \times d_v}$ to compute

$$[\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L] = \text{LayerNorm}\Big([\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]\Big) \tag{1}$$

$$i_t := \arg\max\big(\boldsymbol{W}\boldsymbol{z}_t\big) \in \{1, \ldots, n_q\}, \quad \text{for all } \; t \in [L] \tag{2}$$

$$\boldsymbol{q}_t := \boldsymbol{q}_{(i_t)} \in \mathcal{Q} \subset \mathbb{R}^{d_q}. \tag{3}$$

Here, $\arg\max\big(\boldsymbol{W}\boldsymbol{z}_t\big)$ indicates the entry to the largest value of the vector $\boldsymbol{W}\boldsymbol{z}_t$. Since $\arg\max$ is not differentiable, in practice, we use the Gumbel-Softmax trick (Jang et al., 2017) as a differentiable relaxation of the argmax in the forward pass of the model. Furthermore, following Baevski et al. (2020), we introduce multiple codebooks, identify one codeword from each of the codebook, and then concatenate them. This concatenation approach increases the number of possible quantization vectors at the expense of more parameters; for example, if we use two codebooks, each with n_q codewords, then the total possible quantization vectors is n_q^2.

4 Training

To pretrain the SeisLM, we use a masked reconstruction objective similar to masked language modeling in BERT (Devlin et al., 2019) and masked audio modeling in Wav2vec2 (Baevski et al., 2020). For each masked time step, the pretraining goal is to identify the correct quantized latent representation from a candidate set. After the pretraining, the model is finetuned on labeled samples.

4.1 Pretraining setup

Masking. A portion of the convolutional features $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L)$ is randomly replaced by a shared trainable feature vector during each forward pass of pertaining. To select the masking indices, similar to Baevski et al. (2020), we uniformly sample 6.5% of all time-steps to be starting indices and mask the subsequent 10 time-steps.

Contrastive loss. We pretrain SeisLM with a standard contrastive objective: At each timestep t, we encourage the transformer output $\boldsymbol{a}_t$ to positively correlate with the quantized feature vector $\boldsymbol{q}_t$ of the same timesteps, and negatively correlate with K quantized feature vectors sampled from other timesteps of the same input sequence. Denoting these K negative examples at each timestep t by $\mathcal{N}_t := \{\boldsymbol{n}_t^1, \ldots, \boldsymbol{n}_t^K\} \subset \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_L\}$, we let the contrastive loss of each time t be

$$L(\boldsymbol{a}_t, \boldsymbol{q}_t, \mathcal{N}_t) := -\log \frac{\exp\Big[\text{sim}(\boldsymbol{a}_t, \boldsymbol{q}_t)/\kappa\Big]}{\exp\Big[\text{sim}(\boldsymbol{a}_t, \boldsymbol{q}_t)/\kappa\Big] + \sum_{\boldsymbol{n} \in \mathcal{N}_t} \exp\Big[\text{sim}(\boldsymbol{a}_t, \boldsymbol{n})/\kappa\Big]}. \tag{4}$$

where $\kappa > 0$ is a fixed temperature.

While $L(\boldsymbol{q}_t, \mathcal{Q}_t)$ in (4) is the main loss used for masked pretraining, we add auxiliary losses to encourage the codebook vectors in $\mathcal{Q}$ to be less redundant; this is achieved with an entropy regularization as in Baevski et al. (2020).

Codevector diversity loss. Optimizing the quantization module faces the common issue of underutilized codebooks (Dieleman et al., 2018; Łańcucki et al., 2020; Dhariwal et al., 2020; Mentzer et al., 2024): codewords may remain unused. To address this, following prior works (Baevski et al., 2020; Dieleman et al., 2018), we use a diversity loss to encourage the uniform use of codebook vector. Concretely, let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{BL}\}$ be a batch of B covolutional waveform sequences, each with length L; we let $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{BL}\}$ be the softmax probabilities of codevector assignment: $\boldsymbol{p}_j := \mathrm{softmax}(\boldsymbol{W}\boldsymbol{z}_j) \in \mathbb{R}^{n_q}$, which is a differentiable relaxation of the hard assignment in (2). The average of these codevector assignment probabilities, $\overline{\boldsymbol{p}} := \frac{1}{BL}\sum_{j=1}^{BL} \boldsymbol{p}_j \in \mathbb{R}^{n_q}$ is another probability vector that describes the average usage of all codevector. The diversity loss is defined as $\frac{1}{n_q}\langle \overline{\boldsymbol{p}}, \log \overline{\boldsymbol{p}} \rangle$. However, this diversity loss can itself lead to numerical instability if its strength is not carefully tuned. Our experience shows that this instability is in part due to the *highly unbalanced codebook usage at initialization*. This imbalance triggers a large diversity loss at the outset, leading to substantial initial optimization updates as the model tries to correct it. In Appendix A, we propose a simple way to initialize the model such that the diversity loss remains small. During pretraining, we combine the diversity loss with the contrastive loss. The balance between them is controlled by a hyperparameter.

4.2 Finetuning setup

To finetune pretrained models to a downstream, labeled dataset task, we add a randomly initialized shallow network to process the output of SeisLM. Since SeisLM down-samples waveforms through its convolutional layers, the transformer output has a shorter length than the raw input. Thus, for sequence-labeling tasks that predict each timestep at the original frequency, we use linear interpolation followed by convolutional layers to upsample the latent representation; more details are in Appendix B. During the finetuning, we simply train the parameters of both the SeisLM and the task head. We are aware of prior work that freezes some parts of the pretrained model or uses a scheduler to gradually unfreeze the pretrained model (Baevski et al., 2020) during finetuning; however, these more involved approaches did not bring consistent improvement in our finetuning experiments.

5 Experiment

5.1 Pretraining experiments

	Traces	Region	Tr. length	Sampling rate [Hz]	Type
ETHZ	36,743	Switzerland	variable	100 - 500	Regional
INSTANCE	1,291,537	Italy	120 s	100	Regional
Iquique	13,400	Northern Chile	variable	100	Regional
STEAD	1,265,657	global	60 s	100	Regional
GEOFON	275,274	global	variable	20 - 200	Teleseismic
MLAAPDE	1,905,887	global	120	40	Teleseismic
PNW	183,909	Pacific Northwest	150 s	100	Regional
OBST2024	60,394	global	60 s	100	Regional, submarine

Table 1: Overview of the pretraining datasets from SeisBench (Woollam et al., 2022). While waveforms from these datasets come with various labels such as phase labels (e.g., P-phase vs S-phase), we only use the raw, unannotated data in the training fold for pretraining.

Pretraining data. For the pretraining dataset, we combine waveforms from eight seismic datasets, accessed through the SeisBench (Woollam et al., 2022) framework, into a unifying dataset. The eight datasets are ETHZ (Swiss Seismological Service at ETH Zurich, 1983, 2005, 2008; AlpArray Seismic Network, 2014; European Organization for Nuclear Research (CERN), 2016), INSTANCE (Michelini et al., 2021), Iquique (Woollam et al., 2019), STEAD (Mousavi et al., 2019a), GEOFON (Quinteros et al., 2021), MLAAPDE (Cole et al., 2023), PNW (Ni et al., 2023), and OBST2024 (Niksejel & Zhang, 2024). These datasets consist of preselected waveform snippets, encompassing examples of earthquakes, noise, and exotic signals such as explosions and landslides. Due to this preselection, the prevalence of earthquake signals in this data is substantially higher than on a randomly recorded seismic trace. An overview of these datasets is provided in Table 1. These datasets cover examples

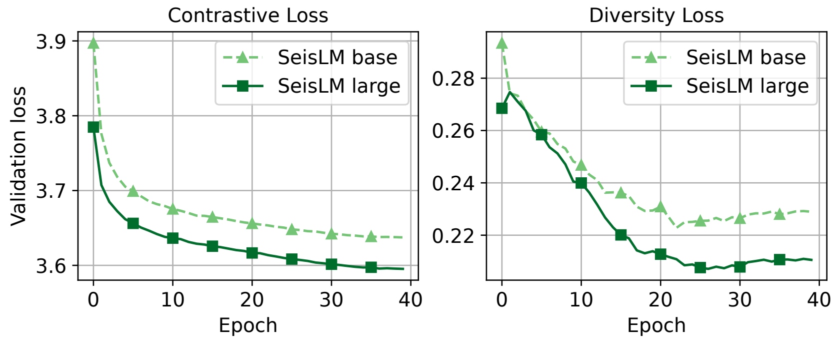

Figure 2: **Pretraining loss of SeisLM.**

from different world regions, different event-to-station distances, and a wide magnitude range. We randomly sample 30s windows from the traces.

Model and training hyperparameters. We briefly outline the hyperparameters used in pretraining and provide full details in Appendix B. We pretrained two variants of models: *SeisLM-base* and *SeisLM-large*. They share the same ConvNet and quantization configurations but *SeisLM-large* uses a larger transformer module than *SeisLM-base*: SeisLM-base includes 6 transformer blocks, while SeisLM-large has 12. The SeisLM-base contains 11.4 million parameters, while SeisLM-large contains 90.7 million parameters. We trained our model with the Adam optimizer (Kingma & Ba, 2015) for 40 epochs. We trained SeisLM-base on four A100-40G GPUs and trained SeisLM-large on four A100-80G GPUs. the pretraining of SeisLM-base and SeisLM-large takes approximately 5 and 8 days, respectively. Figure 2 plots the validation losses of two SeisLM models during pretraining.

Visualizing learned features through dimensionality reduction. Does the reduction of pretraining loss, shown in Figure 2, mean that the model learns useful features from the data? As a sanity check, we run a simple dimensionality reduction experiment. This experiment visualizes whether the pretrained SeisLM, without fitting on any labeled data, could reasonably separate noise and earthquake traces. We collect 1000 noise traces and 1000 earthquake traces from the INSTANCE dataset and input them into SeisLM. For each trace, we average the features from the last layer of SeisLM along the time axis, producing one embedding vector per trace. This process is akin to the bag-of-words model in natural language processing. We apply t-SNE (van der Maaten & Hinton, 2008) to non-linearly reduce the dimensionality of the trace embeddings to 2, to facilitate visualization (Figure 3). The results indicate that, with randomly initialized weights, the SeisLM embeddings of noise (●) and earthquake (▲) traces heavily overlap (left panel of Figure 3); however, after self-supervised pretraining, the separation between the embeddings of noise and earthquake traces gets greatly improved (right panel of Figure 3). We emphasize again the embeddings are learned without using any label; they are colored using labels in Figure 3 for probing purposes.

5.2 Finetuning on phase-picking tasks

We now test whether self-supervised SeisLMs transfer effectively to downstream seismic tasks. Among the many potential downstream tasks, detecting and determining seismic phase types and their onset time are arguably the most fundamental ones; these tasks are typically jointly referred to as *phase-picking* tasks. More specifically, seismic phase onset time is the moment of seismic waves emitted by a source, such as an earthquake, reach a seismic instrument; we usually observe two main phase types of seismic waves, the faster longitudinal P waves and the slower S waves. The results of seismic phase picking form the basis of many subsequent seismological workflows, in particular, earthquake detection through phase association (Zhu et al., 2022; Münchmeyer, 2024), source characterization (Bormann, 2012) or seismic travel-time tomography (Nolet, 1987). All of these steps are integral for accurate and precise seismic hazard assessment.

For a quantitative analysis, we consider the three evaluation tasks defined in the large-scale benchmark by Münchmeyer et al. (2022):

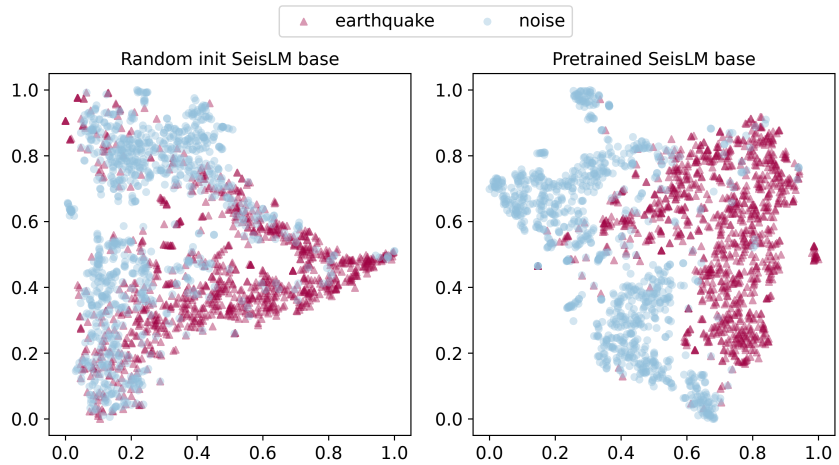

Figure 3: **t-SNE embeddings of SeisLM features.** Compared to a randomly initialized SeisLM-base (left panel), a self-supervised SeisLM-base (right panel) separate the embeddings of earthquake and noise waveforms more effectively.

1. **Event detection**: Given a window of a seismic waveform, determine if it contains an event.
2. **Phase identification**: Given a window containing exactly one phase arrival, determine if it is a P or an S phase.
3. **Onset regression**: Given a window containing exactly one phase arrival of the known type (P or S), determine the onset time.

We show event detection and phase identification results in the main text, and place the onset regression result in Appendix B.

Setup of the baseline models and SeisLMs. In the benchmark study of Münchmeyer et al. (2022), PhaseNet (Zhu & Beroza, 2018) achieves the best overall performance for the three phasepicking tasks described above. We, therefore, use PhaseNet as a baseline, with the same PhaseNet hyperparameters as in Münchmeyer et al. (2022). Note that PhaseNet solves the three-way phase-picking task: for each sample, PhaseNet outputs a 3-dimensional probability vector corresponding to the noise probability, P-phase probability, and the S-phase probability (Zhu & Beroza, 2018; Münchmeyer et al., 2021). For a head-to-head comparison, we follow this joint-training approach to finetune SeisLM. We add two convolutional layers on top of the pre-trained SeisLM with a Softmax activation function in the end, so that it outputs a 3-dimensional probability vector at each timestep, just like the PhaseNet. More details of the finetuning hyperparameters are in Appendix B. For both models, we use 1 minus the noise probability for the event detection. We use the ratio of the peak of the P and S as predictions for the phase identification task. We use the peak position of the relevant phase prediction for the onset regression task.

Finetuning dataset. We use three labeled phase-picking datasets from Seisbench for finetuning (Woollam et al., 2022; Münchmeyer et al., 2022): ETHZ, GEOFON, and STEAD. These datasets reflect different data availability scenarios: ETHZ contains 22k training traces (low data), GEOFON provides 161k traces (medium data), and STEAD offers more than 1 million traces (abundant data). To evaluate model performance across various sample sizes, we divide each dataset into fractions, ranging from 5% to 100%. This allows us to test the models with varying amounts of labeled data. We hypothesize that pretrained models to perform much better than randomly initialized networks in low-data scenarios. In abundant-data scenarios, we anticipate that randomly initialized networks will also perform well, but pretraining should not hinder performance; therefore, we include the large STEAD dataset to stress test the pretrained model.

Event detection. Figure 4 illustrates the event identification results across three datasets. When comparing event detection accuracy at various fractions of the training dataset, pretrained SeisLM models (▲, ■) consistently outperformed PhaseNet (●). The advantage of SeisLM is especially

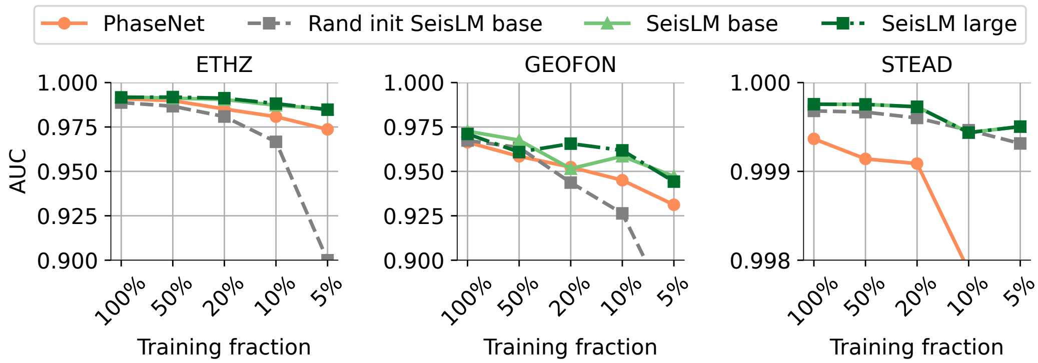

Figure 4: **Performance of models on the event detection task.** Each panel indicates a finetuning dataset. The x axis indicate the fraction of training dataset; the y axis shows the AUC metric: it represents the area under the curve that plots the true positive rate against the false positive rate at various threshold levels for a binary classification task.

pronounced with a limited number of labeled samples, such as when using just 5% of the training data. However, the difference in performance between SeisLM-base (▲) and SeisLM-large (■) is minimal, presumably because this event detection is relatively simple task. Additionally, we compared a SeisLM model fine-tuned from pretrained weights (▲, ■) with a SeisLM-base model initialized with random weights (■). The results show that pretraining benefits performance, particularly when labels are scarce. When there is sufficient labeled data, such as the case of STEAD dataset, then a randomly initialized SeisLM can perform reasonably well.

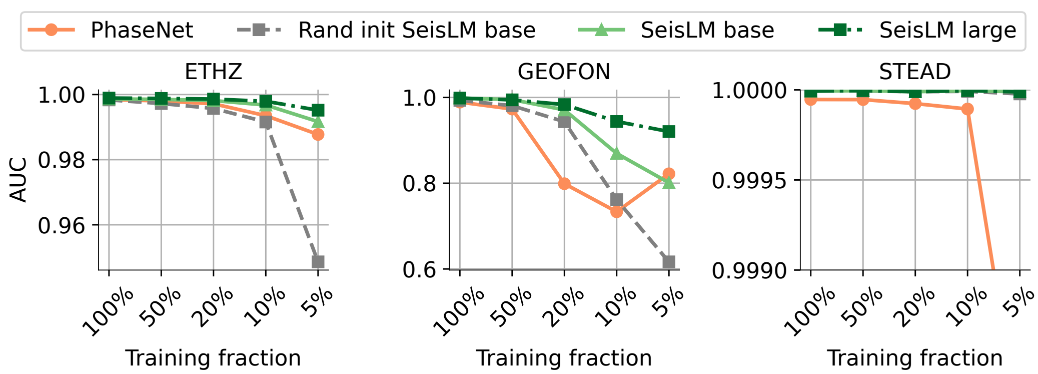

Figure 5: **Performance of models on the phase identification task.**

Phase identification. Figure 5 displays phase identification results across the same three datasets. As with event detection, pretrained SeisLM models (▲, ■) generally deliver higher accuracy than models trained from scratch (●, ■), with the gap widening in low-data scenarios. Additionally, SeisLM-large (■) surpasses SeisLM-base (▲) in this task. When using a substantial amount of data from the largest STEAD dataset, all SeisLM models—whether randomly initialized or pretrained—perform the task near perfect.

5.3 Finetuning on foreshock–aftershock classification tasks

A major challenge in seismology is detecting subtle changes in seismic recordings before and after earthquakes. Gaining insights to these subtle changes can offer early warnings of impending hazards. Previous research has impressively shown that machine learning models can be trained to identify foreshock and aftershock seismic waves (Laurenti et al., 2024). Specifically, Laurenti et al. (2024) classified waveform signals into different categories based on the time relative to the 2016 M6.5 Norcia mainshock in Italy. We apply SeisLM to tackle the same task.

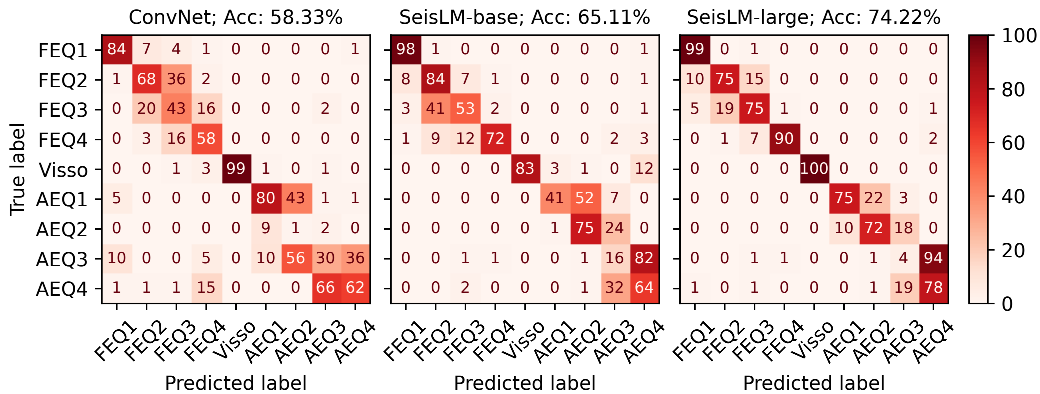

Figure 6: **Confusion matrices of models evaluated on the test fold of the foreshock-–aftershock classification dataset.** The x-axis represents the predicted labels, and the y-axis represents the true labels. The values in the matrices indicate the percentage of predicted samples. The event classes are ordered by time.

Data and model. Following the exact dataset setting of Laurenti et al. (2024, Section 3.1.1), we focused on the waveform recordings from the NRCA station. The foreshock, mainshock, and aftershock events are categorized into nine classes, ranging from FEQ1 (earliest foreshocks), to Visso (the main shock), and finally to AEQ4 (latest aftershocks). These classes are displayed as the labels of the x and y of Figure 6. We use the 7-layer ConvNet from Laurenti et al. (2024, Section 8.2.1) as our baseline model. To fine-tune SeisLM, we add convolutional layers on top of its transformer block; these convolutional layers are followed by global average pooling and a linear head. See Appendix B for more details.

Results. Figure 6 displays the confusion matrices on the test-fold of the foreshock–aftershock dataset. SeisLM's fine-tuning (middle and right panels) improves accuracy over the ConvNet baseline (left panel). Furthermore, reassuringly, the confusion matrices show that SeisLM's errors often occur in temporal proximity—e.g., misclassifying FEQ2 traces as FEQ3 traces and vice versa. Overall, our results provide further support to the hypothesis in Laurenti et al. (2024): fault or source properties before and after a major earthquake show detectable changes that can be identified in seismic recordings.

6 Discussion

Foundation models for seismic waveforms are in their early stages, and important insights are still missing. Take model scaling, for example. In text modeling, researchers have investigated the optimal model size and token count for training transformers within a fixed compute budget, most notably through the Chinchilla scaling law (Hoffmann et al., 2022). We currently lack comparable insights for seismic tasks. Despite this, SeisLM shows the promise of self-supervised learning on unlabeled seismic waveforms—the same strategy behind many seminal foundation models in vision and language modelling. This self-supervised approach enables the pre-trained model to excel in downstream tasks, often surpassing task-specific baselines. It becomes especially helpful when labeled data for downstream tasks is scarce.

The early stage of seismic foundation model research is in contrast with their potential for immense impact. Indeed, earthquakes rank among the most dangerous natural hazards, and even small advances in early warning and hazard assessment could substantially improve safety and reduce economic damage. Leveraging the petabytes of existing seismic data—and likely exponentially more from emerging technologies (Shearer et al., 2023; Zhan, 2020)—self-supervised learning methods applied to vast amounts of unlabeled seismic data may significantly improve seismic data analysis. With the introduction of SeisLM, we have taken a step in this direction.

References

AlpArray Seismic Network. Eastern alpine seismic investigation (EASI)—alparray complimentary experiment, 2014. URL http://networks.seismo.ethz.ch/networks/xt/.

Baevski, A. and Auli, M. Adaptive input representations for neural language modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Baevski, A., Schneider, S., and Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*, 2019.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Proceedings of the Advances in neural information processing systems*, 33:12449–12460, 2020.

Bormann, P. *New Manual of Seismological Observatory Practice (NMSOP-2)*. Deutsches Geo-Forschungszentrum GFZ ; IASPEI, Potsdam, 2012. doi: 10.2312/GFZ.NMSOP-2.

Chen, Y., Savvaidis, A., Saad, O. M., Dino Huang, G.-C., Siervo, D., O'Sullivan, V., McCabe, C., Uku, B., Fleck, P., Burke, G., et al. TXED: The Texas earthquake dataset for AI. *Seismological Research Letters*, 95(3):2013–2022, 2024.

Cole, H. M., Yeck, W. L., and Benz, H. M. MLAAPDE: A machine learning dataset for determining global earthquake source parameters. *Seismological Research Letters*, 94(5):2489–2499, 2023.

Dai, H. and MacBeth, C. The application of back-propagation neural network to automatic picking seismic arrivals from single-component recordings. *Journal of Geophysical Research: Solid Earth*, 102(B7):15105–15113, 1997.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NNACL)*, 2019.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Dieleman, S., Van Den Oord, A., and Simonyan, K. The challenge of realistic music generation: modeling raw audio at scale. *Advances in neural information processing systems*, 31, 2018.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Enescu, N. et al. Seismic data processing using nonlinear prediction and neural networks. In *IEEE NORSIG Symposium, Espoo, Finland*, 1996.

Esposito, A. M., Giudicepietro, F., D'Auria, L., Scarpetta, S., Martini, M. G., Coltelli, M., and Marinaro, M. Unsupervised neural analysis of very-long-period events at stromboli volcano using the self-organizing maps. *Bulletin of the Seismological Society of America*, 98(5):2449–2459, 2008.

European Organization for Nuclear Research (CERN). CERN seismic network, 2016. URL http://networks.seismo.ethz.ch/networks/c4/.

Gentili, S. and Michelini, A. Automatic picking of P and S phases using a neural tree. *Journal of Seismology*, 10(1):39–63, Jan 2006.

Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Hafner, K. and Clayton, R. W. The southern california earthquake data center (SCEDC). *Seismological Research Letters*, 72(6):705–711, 2001.

Havskov, J. and Ottemoller, L. *Routine data processing in earthquake seismology: with sample data, exercises and software*. Springer Science & Business Media, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with Gumbel-Softmax. In *Proceedings of the International Conference on Learning Representations*, 2017.

Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 25, 2012.

Łańcucki, A., Chorowski, J., Sanchez, G., Marxer, R., Chen, N., Dolfing, H. J., Khurana, S., Alumäe, T., and Laurent, A. Robust training of vector quantized bottleneck models. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

Laurenti, L., Paoletti, G., Tinti, E., Galasso, F., Collettini, C., and Marone, C. Probing the evolution of fault properties during the seismic cycle with deep learning, 2024.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, S., Yang, X., Cao, A., Wang, C., Liu, Y., Liu, Y., and Niu, Q. SeisT: A foundational deep-learning model for earthquake monitoring tasks. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL)*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Mentzer, F., Minnen, D., Agustsson, E., and Tschannen, M. Finite scalar quantization: VQ-VAE made simple. In *Proceedings of the International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=8ishA3LxN8`.

Michelini, A., Cianetti, S., Gaviano, S., Giunchi, C., Jozinović, D., and Lauciani, V. INSTANCE—the Italian seismic dataset for machine learning. *Earth System Science Data*, 13(12):5509–5544, 2021. doi: 10.5194/essd-13-5509-2021. URL `https://essd.copernicus.org/articles/13/5509/2021/`.

Mousavi, S. M., Sheng, Y., Zhu, W., and Beroza, G. C. STanford EArthquake Dataset (STEAD): A global data set of seismic signals for ai. *IEEE Access*, 7:179464–179476, 2019a.

Mousavi, S. M., Zhu, W., Ellsworth, W., and Beroza, G. Unsupervised clustering of seismic signals using deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, 16(11): 1693–1697, 2019b.

Mousavi, S. M., Zhu, W., Sheng, Y., and Beroza, G. C. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific Reports*, 9(1):10267, 2019c.

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., and Beroza, G. C. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(1):3952, 2020.

Münchmeyer, J. Pyocto: A high-throughput seismic phase associator. *Seismica*, 3(1), 2024.

Münchmeyer, J., Bindi, D., Leser, U., and Tilmann, F. The transformer earthquake alerting model: A new versatile approach to earthquake early warning. *Geophysical Journal International*, 225(1): 646–656, 2021.

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

Nguyen, T. Q. and Salazar, J. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, 2019.

Ni, Y., Hutko, A., Skene, F., Denolle, M., Malone, S., Bodin, P., Hartog, R., and Wright, A. Curated pacific northwest ai-ready seismic dataset. *Seismica*, 2(1), May 2023.

Niksejel, A. and Zhang, M. OBSTransformer: a deep-learning seismic phase picker for OBS data using automated labelling and transfer learning. *Geophysical Journal International*, 237(1): 485–505, 2024.

Nolet, G. Seismic wave propagation and seismic tomography. In *Seismic tomography: With applications in global seismology and exploration geophysics*, pp. 1–23. Springer, 1987.

Quinteros, J., Strollo, A., Evans, P. L., Hanka, W., Heinloo, A., Hemmleb, S., Hillmann, L., Jaeckel, K.-H., Kind, R., Saul, J., et al. The GEOFON program in 2020. *Seismological Research Letters*, 92(3):1610–1622, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL `https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf`.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Ronneberger, O., Fischer, P., and Brox, T. U-net: convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical image Computing and Computer-assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015.

Ross, Z. E., Meier, M.-A., Hauksson, E., and Heaton, T. H. Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, 108(5A):2894–2901, 2018.

Seydoux, L., Balestriero, R., Poli, P., Hoop, M. d., Campillo, M., and Baraniuk, R. Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning. *Nature Communications*, 11(1):3972, 2020.

Shearer, P. M., Meng, H., and Fan, W. Earthquake detection using a nodal array on the san jacinto fault in california: Evidence for high foreshock rates preceding many events. *Journal of Geophysical Research: Solid Earth*, 128(3):e2022JB025279, 2023.

Sheng, H., Wu, X., Si, X., Li, J., Zhang, S., and Duan, X. Seismic Foundation Model (SFM): A new generation deep learning model in geophysics. *arXiv preprint arXiv:2309.02791*, 2023.

Si, X., Wu, X., Sheng, H., Zhu, J., and Li, Z. SeisCLIP: A seismology foundation model pre-trained by multi-modal data for multi-purpose seismic feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Soto, H. and Schurr, B. DeepPhasePick: A method for detecting and picking seismic phases from local earthquakes based on highly optimized convolutional and recurrent deep neural networks. *Geophysical Journal International*, 227(2):1268–1294, 2021.

Swiss Seismological Service at ETH Zurich. National seismic networks of switzerland, 1983. URL http://networks.seismo.ethz.ch/networks/ch/.

Swiss Seismological Service at ETH Zurich. Temporary deployments in switzerland associated with aftershocks and other seismic sequences, 2005. URL http://networks.seismo.ethz.ch/networks/8d/.

Swiss Seismological Service at ETH Zurich. Seismology at school program, ETH Zurich, 2008. URL http://networks.seismo.ethz.ch/networks/s/.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Woollam, J., Rietbrock, A., Bueno, A., and De Angelis, S. Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network. *Seismological Research Letters*, 90(2A):491–502, 2019.

Woollam, J., Münchmeyer, J., Tilmann, F., Rietbrock, A., Lange, D., Bornstein, T., Diehl, T., Giunchi, C., Haslinger, F., Jozinović, D., Michelini, A., Saul, J., and Soto, H. SeisBench—A Toolbox for Machine Learning in Seismology. *Seismological Research Letters*, 93(3):1695–1709, 03 2022.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10524–10533. PMLR, 2020.

Yoma, N. B., Wuth, J., Pinto, A., de Celis, N., Celis, J., Huenupan, F., and Fustos-Toribio, I. J. End-to-end LSTM based estimation of volcano event epicenter localization. *Journal of Volcanology and Geothermal Research*, 429:107615, 2022.

Yoon, C. E., O'Reilly, O., Bergen, K. J., and Beroza, G. C. Earthquake detection through computationally efficient similarity search. *Science Advances*, 1(11):e1501057, 2015.

Zhan, Z. Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas. *Seismological Research Letters*, 91(1):1–15, 2020.

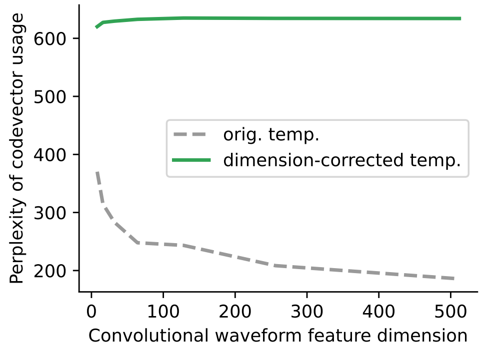

Figure 7: The influence of the standard temperature τ and dimension-corrected temperature $\tau/\sqrt{n_q}$ in a randomly initialized Gumbel quantizer. When the convolutional feature dimension d_v (x-axis) increases, the perplexity of codevector (y-axis) increases in the case of standard temperature (grey curve); this indicates uneven usage of codebook vectors. With the dimensionality correction, the perplexity stays roughly constant (green curve).

Zhao, Y. and Takano, K. An artificial neural network approach for broadband seismic phase picking. *Bulletin of the Seismological Society of America*, 89(3):670–680, 1999.

Zhong, Y. and Tan, Y. J. Deep-learning-based phase picking for volcano-tectonic and long-period earthquakes. *Geophysical Research Letters*, 51(12):e2024GL108438, 2024.

Zhou, H.-W. *Practical Seismic Data Analysis*. Cambridge University Press, 2014.

Zhu, W. and Beroza, G. C. PhaseNet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 10 2018.

Zhu, W., McBrearty, I. W., Mousavi, S. M., Ellsworth, W. L., and Beroza, G. C. Earthquake phase association using a bayesian gaussian mixture model. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023249, 2022.

A Details of the model

Quantization. We show that this phenomenon of uneven usage of randomly initialized codebooks can be easily understood. During training, the forward pass of the Gumbel-quantizer computes

$$\boldsymbol{p}_t \coloneqq \operatorname{softmax}\big[(\boldsymbol{W}\boldsymbol{z}_t + \boldsymbol{n})/\tau\big] \in \mathbb{R}^{n_q}, \quad \text{with } \boldsymbol{n}_j \overset{\text{iid}}{\sim} \operatorname{Gumbel}(0,1) \text{ for all } j \in [n_q], \tag{5}$$

$$i_t \sim \operatorname{Categorical}(\boldsymbol{p}_t), \quad i_t \in \{1, \ldots, n_q\} \tag{6}$$

where τ is a temperature. At initialization, the entries of the weight projection matrix $\boldsymbol{W}$ are typically drawn from a Normal distribution[1]. Assume that the convolutional feature $\boldsymbol{z} \in \mathbb{R}^{d_v}$ follows a normal distribution. In this case, the entries of $(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{n})/\tau$ follow a zero-mean Gaussian distribution with variance proportional to d_v, the dimension of convolutional features. Given that d_v is typically in the order of hundreds, the variance is in the same order, leading to nearly one-hot vectors after the softmax. This makes the categorical sampling nearly deterministic and less exploratory for codevectors. Additionally, since larger models often use greater codevector dimensions d_v, larger models more prone to this problem. We illustrate this in Figure 7. As a simple fix, we re-parametrize the temperature τ as $\tau \coloneqq \tau' \sqrt{n_q}$. This re-parametrization breaks the link between the convolutional feature dimension d_v and its impact on uneven codevector usage at initialization.

[1]As in the implementation of Gumbel quantizer of Fairseq and Hugging Face transformer

B Experimental details

B.1 Pretraining experiments

Model hyperparameters We pretrained two variants of models: *SeisLM-base* and *SeisLM-large*. They share the same ConvNet and quantization configurations but *SeisLM-large* uses a larger transformer module than *SeisLM-base*. For the ConvNet module, each model uses two convolutional layers with 256 channels, a kernel size of 3, and a stride of 2. In the vector quantization module, each model uses two groups of code vectors, each containing 320 vectors. Furthermore, each model's position embedding component (placed at the start of the transformer module) uses a grouped convolutional layer (Krizhevsky et al., 2012) with a kernel size of 128 and 16 groups. In the rest of the transformer module, SeisLM-base includes 6 pre-norm transformer blocks, while SeisLM-large has 12. Unlike the standard transformer block, the pre-norm version applies layer normalization before the self-attention and feedforward layers. This modification often leads to more stable training (Baevski & Auli, 2019; Nguyen & Salazar, 2019; Xiong et al., 2020). Each transformer block employs a 12-headed self-attention layer and a residual 2-layer MLP with 3072 hidden units. The number of output units of the MLP is 240 for SeisLM-base and 768 for SeisLM-large. With these settings, SeisLM-base contains 11.4 million parameters, while SeisLM-large contains 90.7 million parameters.

Training hyperparameters For the contrastive loss, we randomly sample $K = 100$ quantization vectors from the convolutional feature sequences as negative examples, with a temperature $\kappa = 0.1$ in (4). We trained our model with the Adam optimizer (Kingma & Ba, 2015) for 40 epochs. We traind SeisLM-base with a global batch size of 112 on four A100-40G GPUs, and trained SeisLM-large with a global batch size of 192 on four A100-80G GPUs. The learning rate scheduler uses cosine annealing with a linear warmup. The maximum learning rate is 5e-4 for SeisLM-base and 1e-3 for SeisLM-large, with the same warmup fraction of 20%. During training, we decreased the Gumbel temperature from 2.0 to 0.5. We did not apply dropout, drop layers, or weight decay during pretraining. We trained SeisLM-base with 16-bit precision and SeisLM-large with 32-bit precision. With these settings, the pretraining of SeisLM-base and SeisLM-large takes approximately 5 and 8 days, respectively. Figure 2 plots the validation losses of two SeisLM models during pretraining.

B.2 Phase-picking experiments

Hyperparameters of the finetuned SeisLM. Since Pretrained SeisLM down-samples waveforms through its convolutional layers, the transformer's output is shorter than the raw input. For phase-picking tasks, we upsample the latent representation to match the input length using linear interpolation. We then concatenate this upsampled representation with the raw waveforms and apply two convolutional layers to fit the labels. Specifically, we use two convolutional layers with a kernel width of 3, stride of 1, and GELU activations. These layers maintain the number of channels in the transformer features. For fine-tuning SeisLM-base, we use 240 + 3 convolutional filters, and for SeisLM-large, we use 768 + 3 filters. We also apply dropout with a rate of 0.2 after each convolutional layer.

Onset regression. Figure 8 displays the onset regression result, which recapitulated our findings on the two phasepicking tasks above. Pretrained SeisLM (▲, ■) generally achieves lower onset regression than train-from-scratch baselines (●, ■).

B.3 Foreshock–aftershock experiments

Hyperparameters of the finetuned SeisLM. For foreshock–aftershock tasks, we add a 4-layer convolutional network on top of pretrained SeisLM. These convolutional layers have a kernel width of 3, stride of 2, and GELU activations, and they maintain the number of channels in the transformer features. A global average pooling layer and a linear head follow the convolutional layers, turning the features into a vector of 9 classes.

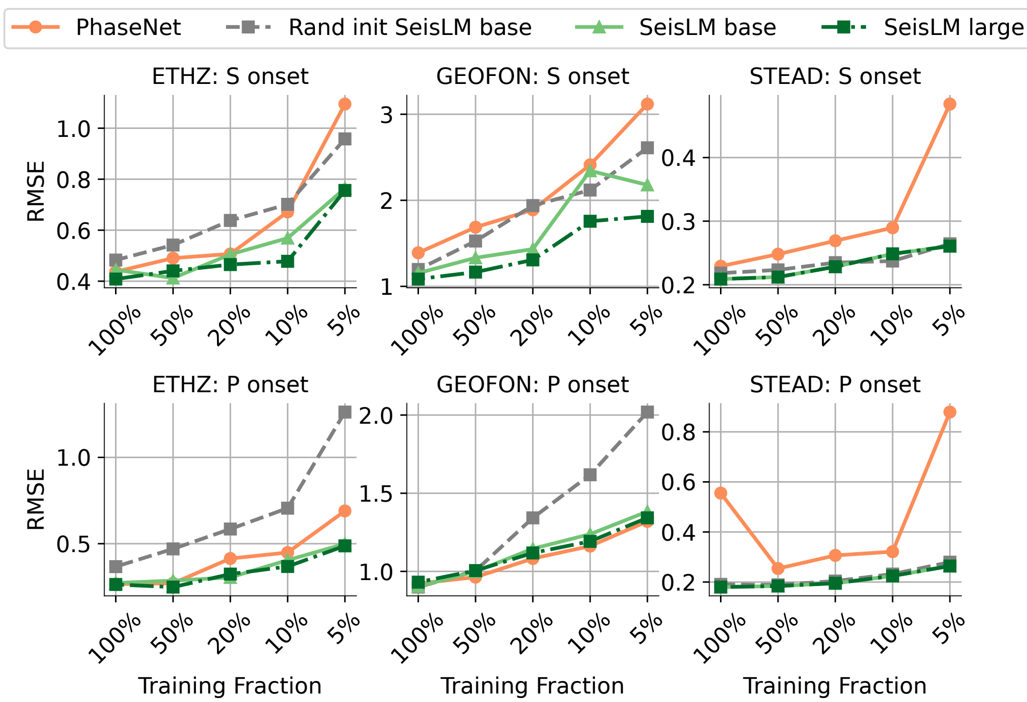

Figure 8: **Onset regression.**