Leveraging Physics-Informed Neural Networks in Geotechnical Earthquake Engineering: An Assessment on Seismic Site Response Analyses

Chenying Liu * Jorge Macedo † Alexander Rodriguez ‡

Abstract

The primary objective of this study is to assess the potential of physics-informed neural networks (PINNs) for seismic site response analyses (SRA). PINNs constitute a novel computational paradigm that combines physical principles with data-driven methods to solve differential equations. Despite the growing exploration of machine and deep learning in geotechnical earthquake engineering, the integration of PINNs remains limited. The study first addresses key challenges in applying PINNs to SRAs. In particular, the broad range of frequencies in ground motion recordings, which complicates the training process, and neural network architectural issues are discussed. Fourier feature embedding, a relatively new technique in image processing, learning rate adjustment, a tailored training strategy, and a PINN architecture are proposed to address the identified challenges. The proposed framework is evaluated by comparing SRA results from the implemented PINN and traditional numerical techniques, considering different ground motions and soil systems. The results of the proposed PINN and numerical techniques are identical, highlighting the robustness of the proposed framework. The encouraging results suggest there is significant potential for PINNs in general geotechnical earthquake engineering applications, which is also discussed.

1 Introduction

The primary objective of this study is to evaluate the potential of physics-informed neural networks (PINNs) for seismic site response analyses, a critical issue in geotechnical earthquake engineering

*Graduate Student Researcher, Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA
†Associate Professor, Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA
‡Assistant Professor, University of Michigan, Ann Arbor, MI 48109, USA

(GEE). Towards this end, a PINN-based framework for conducting site response analyses is proposed. The insights gained from this study are anticipated to contribute to the broader application of PINNs in various GEE challenges in future research endeavors. Recent advancements in earthquake engineering have increasingly integrated machine learning (ML) and deep learning (DL) techniques across seismology, structural engineering, and geotechnical engineering (Xie et al. (2020); Maurer and Sanger (2023); Mousavi et al. (2020)). ML and DL offer substantial benefits, including enhanced analysis of complex datasets (Alimoradi and Beck (2015)), development of performance-based earthquake (PBE) procedures (Kazemi et al. (2023); Liu and Macedo (2022, 2024); Macedo et al. (2021)), and improved computational efficiency (Tomar and Burton (2021)). Despite these advancements, the application of PINNs in earthquake engineering remains limited. PINNs represent a novel computational paradigm integrating physical principles with data-driven approaches to solve differential equations and model physical phenomena. Unlike traditional neural networks that rely solely on data, PINNs incorporate physics through specially formulated loss functions, ensuring that model predictions align with established physical principles. For a comprehensive review of PINNs, the reader is referred to Raissi et al. (2019).

Several studies discuss recent advancements in machine learning and deep learning applications in GEE. For example, Dupuis et al. (2023) developed a deep neural network to evaluate the minimum usable frequency in New Zealand earthquake ground motion records for ground motion model (GMM) development. Liu and Macedo (2022) employed a residual neural network to estimate slope displacements induced by subduction-type earthquakes. Wu et al. (2023) utilized a gated recurrent unit neural network to assess the seismic response of underground structures in layered soils. In a more recent study, Maurer and Sanger (2023) reviewed 75 publications on ML/DL applications in liquefaction problems, identifying challenges and underscoring the potential of ML/DL to advance earthquake engineering practices. All these studies highlight the advantages of ML/DL, in particular, regarding data assessment and PBE evaluations over traditional statistical approaches. In the domain of seismic site response analyses, recent ML/DL efforts have focused on site characterization and data-driven model development. For example, Ji et al. (2023) developed a deep convolutional neural network for site classification using image-based topographic information and microtremor horizontal-to-vertical spectral ratio (HVSR) data. Ilhan et al. (2019) proposed deep learning-based site amplification models using large-scale simulations from Central and Eastern North America, demonstrating improved performance over traditional methods. Lee et al. (2023) developed site amplification prediction models with random forest and deep neural networks. Van Nguyen et al. (2024) employed bidirectional long short-term memory networks for predicting seismic site response,

showing strong performance in numerical predictions and centrifuge tests. Zhu et al. (2023) used a site amplification database to validate the performance of machine learning-based models in hold-out tests.

Efforts to apply PINNs in earthquake engineering remain limited compared to ML/DL applications. Indeed, the authors are unaware of previous efforts focusing on seismic site response analyses. Recent PINN applications have largely focused on geophysical, hydraulic, and physics-related problems. For example, Fukushima et al. (2023) used PINNs to estimate frictional parameters and predict fault slips in fault-slip evolution. Rasht-Behesht et al. (2022) applied PINNs to acoustic wave propagation, with potential extensions to teleseismic plane waves and seismic point sources. Brandolin et al. (2024) utilized PINNs for interpolating seismic data to derive velocity models and seismic images for geophysical applications. Guo and Fang (2023) incorporated physical principles into PINNs for structural parameter identification, demonstrating its application to estimating stiffness in frame structures. Gao et al. (2023) used PINN combined with data measurements for predicting the phreatic surface in seepage applications. PINNs have also been applied to solve the Navier-Stokes equations, reducing computational costs by leveraging sparse data compared to traditional numerical methods (Cai et al. (2021)). Laubscher (2021) highlighted the potential of PINNs in multi-physics problems by integrating multiple governing equations into a neural network, facilitating the simultaneous learning of different aspects of multi-physics systems.

2 Deep Learning and PINNs

Deep learning focuses on developing and applying artificial neural networks (ANNs) that emulate the intricate workings of the human brain. Deep learning architectures are typically composed of multiple layers of neurons organized into input, hidden, and output layers. The input layer receives raw data, which is then transformed into more abstract and sophisticated representations through successive hidden layers. Each neuron in a layer is connected to neurons in the subsequent layer via synaptic weights, which are adjusted during training. Figure 1 shows a simple architecture of a fully connected neural network (multi-layer perceptron). The input layer receives two features, x_1 and x_2, which are subsequently processed by the hidden layers. The output layer produces the label y. The transformation performed by hidden layers for this architecture can be mathematically represented as:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \tag{1}$$

where $\mathbf{h}(l)$ denotes the activations of the l-th layer, $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector for the l-th layer, and σ is a non-linear activation function, examples of which are the ReLU (rectified linear unit, Goodfellow et al. (2016)), Sigmoid (Goodfellow et al. (2016)), or Tanh (Goodfellow et al. (2016)) functions. The training process, aimed at optimizing these parameters (e.g., $\mathbf{W}$ and $\mathbf{b}$), typically employs stochastic gradient descent (SGD) and its variants, leveraging the back-propagation algorithm (Rumelhart et al., 1986) to minimize a loss function $L(\theta)$ that quantifies the discrepancy between predicted (y) and actual outputs (y_{true}):

$$\theta^* = \arg\min_{\theta} L(\theta, y, y_{true}) \tag{2}$$

where θ encompasses all the learnable parameters (weights and biases) in the network. The remarkable capability of deep learning models to learn from vast amounts of high-dimensional data directly results from their depth and the non-linear transformations they apply. This ability is further augmented by advancements in hardware that enable parallelization of matrix/tensor operations, notably GPUs and TPUs, which facilitate the efficient training of deep networks (Wang et al., 2019).

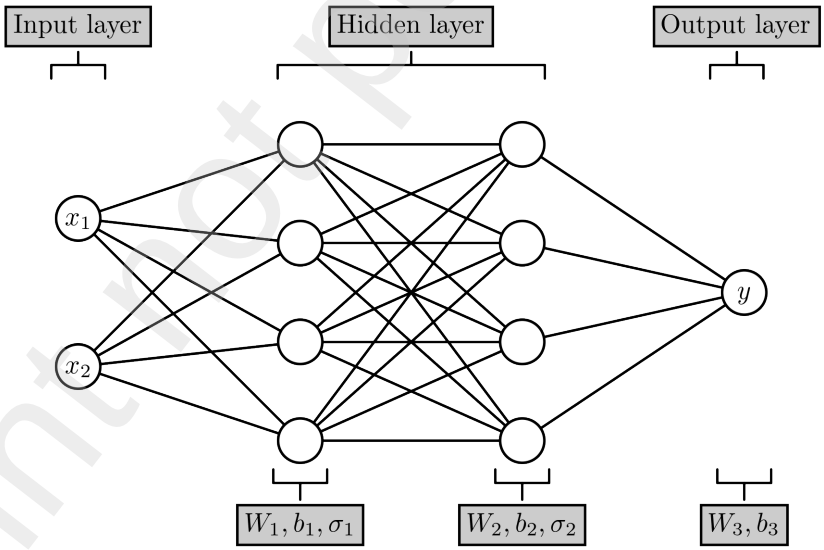

Figure 1: A schematic illustration of a fully connected neural network (multi-layer perceptron). The input consists of two features, x_1 and x_2, which are processed by the network to generate the output prediction, y.

As previously discussed, PINNs combine the strengths of deep learning with the rigor of domain-specific physical laws and constraints. Unlike traditional neural networks that rely exclusively on

data, PINNs integrate governing physical equations, such as differential equations, into the learning framework through the formulation of the loss functions. Without loss of generality, consider a partial differential equation (PDE) of this form:

$$\frac{\partial \mathbf{f}(t,\mathbf{x})}{\partial t} + \mathcal{N}[\mathbf{f}](t,\mathbf{x}) = 0,\ t \in [0,T],\ \mathbf{x} \in \Omega \tag{3}$$

subject to the initial and boundary conditions:

$$\mathbf{f}(t=0,\mathbf{x}) = \mathbf{g}(\mathbf{x}), \mathbf{x} \in \Omega \tag{4}$$

$$\mathcal{B}[\mathbf{f}](t,\mathbf{x}) = 0,\ t \in [0,T], \mathbf{x} \in \partial\Omega \tag{5}$$

where $\mathcal{N}[\cdot]$ is a linear or nonlinear differential operator, and $\mathcal{B}[\cdot]$ is a boundary operator corresponding to the boundary conditions, Ω is the domain of the independent variables $\mathbf{x}$, and T is the upper limit of time t. In addition, $\mathbf{f}(t,\mathbf{x})$ describes the unknown latent solution that is governed by the PDE system. As neural networks are well-known as universal function approximators, the unknown solution $\mathbf{f}(t,\mathbf{x})$ can be represented by a deep neural network $\mathbf{f}_\theta(t,\mathbf{x})$, where θ denotes all tunable parameters of the neural network. The fundamental principle underlying PINNs is the incorporation of the governing equations directly into the neural network's loss function. This is achieved by embedding the residuals of the governing equations into the loss, which can be expressed as:

$$L(\theta) = \lambda_{ic} L_{ic}(\theta) + \lambda_{bc} L_{bc}(\theta) + L_r(\theta) \tag{6}$$

$$L_{ic}(\theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} |\mathbf{f}_\theta(0, \mathbf{x}_{ic}^i) - \mathbf{g}(\mathbf{x}_{ic}^i)|^2 \tag{7}$$

$$L_{bc}(\theta) = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} |\mathcal{B}[\mathbf{f}_\theta](t_{bc}^i, \mathbf{x}_{bc}^i)|^2 \tag{8}$$

$$L_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} |\frac{\partial \mathbf{f}_\theta}{\partial t}(t_r, \mathbf{x}_r) + \mathcal{N}[\mathbf{f}_\theta](t_r, \mathbf{x}_r)|^2 \tag{9}$$

where $L_{ic}(\theta), L_{bc}(\theta)$, and $L_r(\theta)$ represent the loss functions associated with the initial conditions, boundary conditions, and PDE residuals, respectively. N_{ic}, N_{bc}, and N_r represent the number of initial conditions, boundary conditions, and PDE residuals. λ_{ic} and λ_{bc} represent the weight factors of the corresponding loss functions. t_r, t_{bc}, $\mathbf{x}_r$, $\mathbf{x}_{bc}$, and $\mathbf{x}_{ic}$ presents either the vertices of a fixed mesh (e.g., a finite element mesh) or randomly sampled points in the domain. By leveraging automatic differentiation, the Jacobian matrices, Hessian matrices, and higher-order

gradients needed to learn the parameters of the neural network can be computed using established procedures (e.g., Goodfellow et al. (2016)).

The primary distinction between training a PINN and a traditional supervised learning model lies in their behavior regarding overfitting. For instance, in a noise-free regression task illustrated in Figure 2(a), when a regression model is overly complex relative to the ground truth function, it tends to overfit the data, resulting in nearly zero loss. However, the predicted values at points of interest (represented by the vertical lines in Figure 2(a)) closely approximate the true values. This situation is acceptable if the model's intended use is solely as a lookup dictionary and not for extrapolation or interpolation beyond its training domain. Conversely, overfitting a PINN (as illustrated in Figure 2(b) and (c)) does not guarantee that the solution function $\mathbf{f}_\theta(t, \mathbf{x})$ predicted by the PINN closely approximates the real solution $\mathbf{f}(t, \mathbf{x})$ even at training point locations. The assurance provided by a PINN is solely that $\mathbf{f}_\theta(t, \mathbf{x})$ and its partial derivatives satisfy the differential equations (assuming zero loss) and the prescribed initial and boundary conditions. In essence, the uniqueness of a solution to a well-posed differential equation is ensured only when these constraints are satisfied at every point within the domain. However, practical training of a PINN typically achieves only a soft satisfaction of these conditions across a finite set of training samples. Consequently, without careful tuning and caution, multiple solutions to any well-posed differential equation may emerge. This observation highlights the importance of careful handling of the training process, which will be detailed in subsequent sections. Leveraging domain knowledge also plays a key role in training PINNs, as it can guide the selection of hyperparameters, model architecture, and validation techniques (Raissi et al., 2019; Cai et al., 2021).

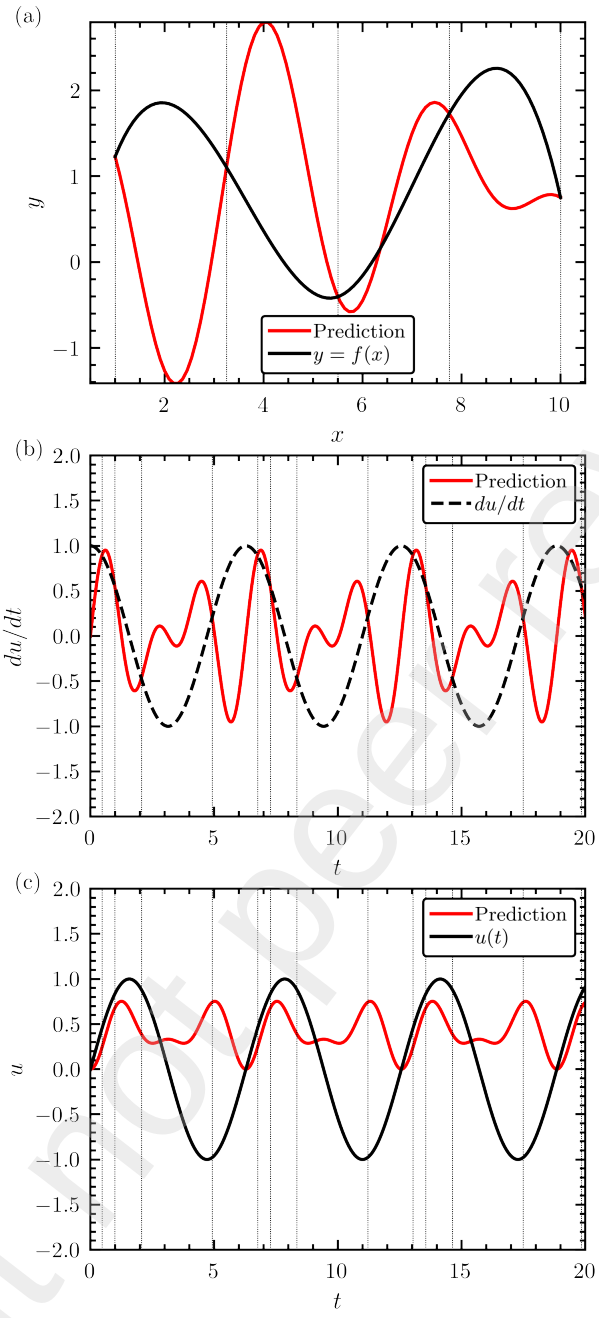

Figure 2: Overfitting of a regression neural network and a PINN. (a) A regression model overfits the training data (denoted by vertical dotted lines), matching the observations at training points x. (b) and (c) A PINN for solving the problem: $du/dt = \cos t, u(0) = 0, u = \sin t$, which also overfits the training data (denoted by vertical dotted lines); but in this case, the predictions at the training points t do not align with the observations $u(t)$.

3 Seismic Site Response

Seismic site response analyses in GEE aim to propagate ground motion to the surface from a reference level, typically associated with bedrock conditions. This process helps to understand how soil affects the filtering or amplification of ground motion frequencies, providing critical inputs for seismic design. In practice, one-dimensional (1D) site response analyses are the most common approach for site response analyses (Rodríguez-Marek et al., 2001; Kramer, 1996); hence, this study will focus on this approach. In 1D analyses, ground motion is approximated by the vertical propagation of horizontal shear waves, with soil behavior modeled as a Kelvin-Voigt solid characterized by linear elastic shear modulus and viscous damping (Idriss and Seed, 1968). Non-linearity is often addressed using the equivalent-linear method (Idriss and Sun, 1992; Markham et al., 2016). The wave propagation equation can be solved in either the frequency or time domain. Frequency domain analysis involves decomposing seismic ground motion into its frequency components via Fourier transforms, while time domain analysis directly models seismic ground motion over time. This study evaluates the potential of PINNs to address the 1D site response problem using time-domain analysis.

The soil layers in 1D time-domain site response analysis can be idealized as a discrete lumped mass system (Figure 3). Hence, the governing equations are a set of ordinary differential equations (ODEs) derived from the motion equation:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}(t)\dot{\mathbf{u}}(t) + \mathbf{K}(t)\mathbf{u}(t) = -\mathbf{M}I\ddot{u}_g(t) \tag{10}$$

subject to the following initial conditions:

$$\mathbf{u}(0) = \mathbf{0}, \dot{\mathbf{u}}(0) = \mathbf{0} \tag{11}$$

where $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$, and $\mathbf{u}$ are the vectors of nodal relative acceleration, velocity, and displacement, respectively. Each has a length equal to the number of degrees of freedom. $\ddot{u}_g$ is the acceleration at the base of the soil column (i.e., the earthquake excitation). $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the mass, viscous damping, and stiffness matrices, respectively. For linear elastic analysis, the matrices $\mathbf{C}$ and $\mathbf{K}$ are time-independent and derived from the initial soil properties. In equivalent linear analysis, although $\mathbf{C}$ and $\mathbf{K}$ remain constant over time, they are compatible with the effective shear strain, which is a scalar-valued function of $\mathbf{u}(t)$ over all t and also depends of shear modulus and damping ratio reduction relationships.

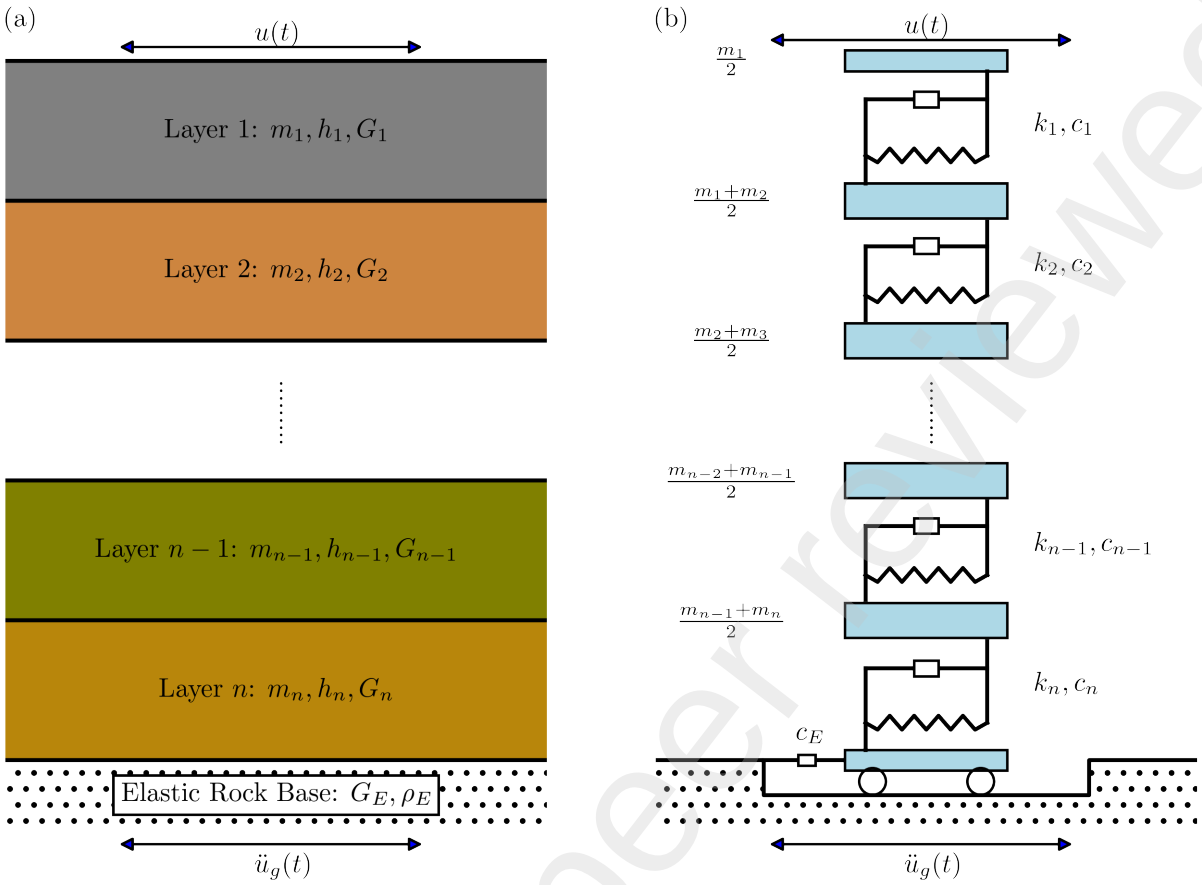

Figure 3: Illustration of the lumped mass formulation for seismic site response considering n layers: (a) Idealized soil stratigraphy and (b) multi-degree-of-freedom lumped masses.

The physics (i.e., equations of motion) of the site response problem can be incorporated into the loss function of o a PINN, by considering $N(t;\Theta) = \mathbf{u}(t)$ and minimizing:

$$L(\Theta) = \lambda L_{ic}(\Theta) + L_r(\Theta) \tag{12}$$

$$L_{ic}(\Theta) = |N(0;\Theta)|^2 + |\dot{N}(0;\Theta)|^2 \tag{13}$$

$$L_r(\Theta) = \frac{1}{N_t}\sum_{i=1}^{N_t} ||\mathbf{M}\ddot{N}(t_i;\Theta) + \mathbf{C}(t_i)\dot{N}(t_i;\Theta) + \mathbf{K}(t_i)N(t_i;\Theta) + \mathbf{M}I\ddot{u}_g(t_i)||_2^2 \tag{14}$$

where $N(t;\Theta)$ is a neural network with parameters Θ, that takes as input a scalar-valued time t. L_{ic} and L_r are the loss function corresponding to the initial conditions and ODE residuals. λ is a weighting factor for adjusting the relative weight between L_{ic} and L_r, therefore, their contribution to the gradients of Θ. Note that the Euclidean norm is used in L_r to account for the multiple degrees of freedom. Figure 4 shows a schematic illustration of a PINN for solving the site response problem. The detailed implementation and associated discussions are described in subsequent sections.

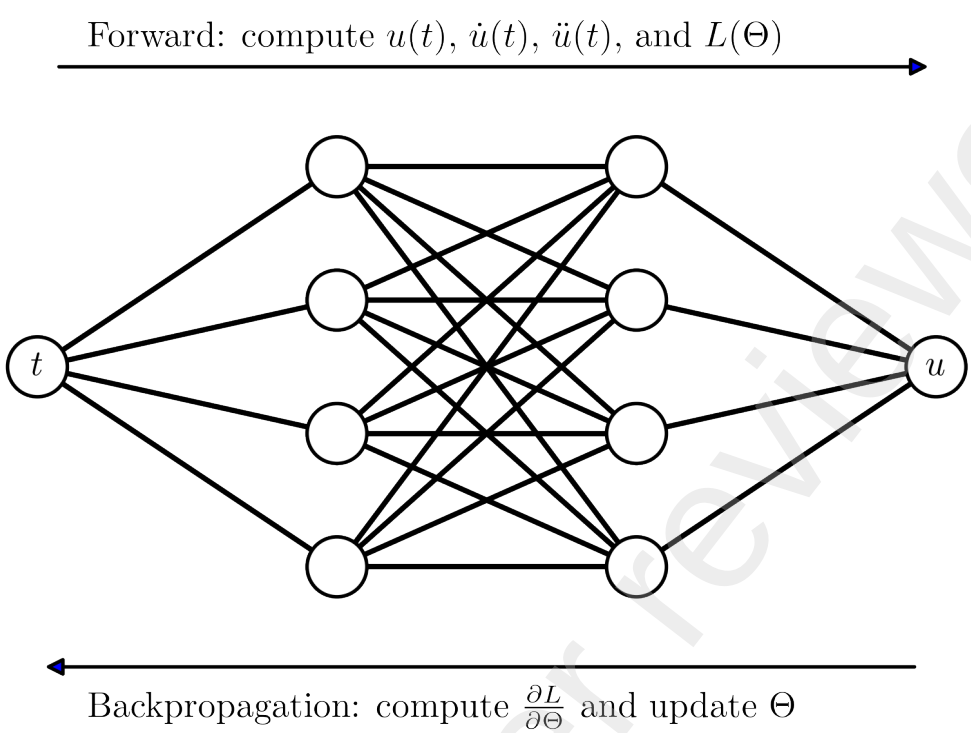

Figure 4: Illustration of a PINN for solving seismic site response problems. In each training iteration, a forward pass is executed to compute the variables $u(t)$, $\dot{u}(t)$, and $\ddot{u}(t)$, and to estimate the loss. This is followed by backpropagation, which calculates the gradient of the loss with respect to the PINN parameters (Θ) and updates the parameters accordingly.

4 PINN Implementation Challenges and Proposed Solutions

Training PINNs is more complex than training traditional neural networks due to the intricacy of the loss functions. These loss functions often incorporate residuals of differential equations, boundary conditions, and initial conditions (e.g., Equation 6), which can impede convergence. Additionally, the loss components may operate on different scales, necessitating careful weighting to ensure that the network appropriately captures the underlying physics. In the context of seismic site response, the input consists of transient seismic excitations across a range of frequencies. Thus, PINNs must account for the transient nature of the responses and the frequency spectrum, presenting further challenges as discussed subsequently.

Without loss of generality, we consider the system in Figure 5(a) to discuss the formulation of PINNs for conducting site response assessments. In doing so, we also highlight potential challenges and proposed solutions. The system is a homogeneous soil laid on top of a rigid bedrock, and it is excited by the ground motion acceleration shown in Figure 5(b), which is applied at the top of

the rigid bedrock. The objective is to predict the acceleration, velocity, and displacement at the surface of the soil layer. The assessments included various combinations of model architectures, optimization algorithms, and feature engineering techniques, with the findings summarized below.

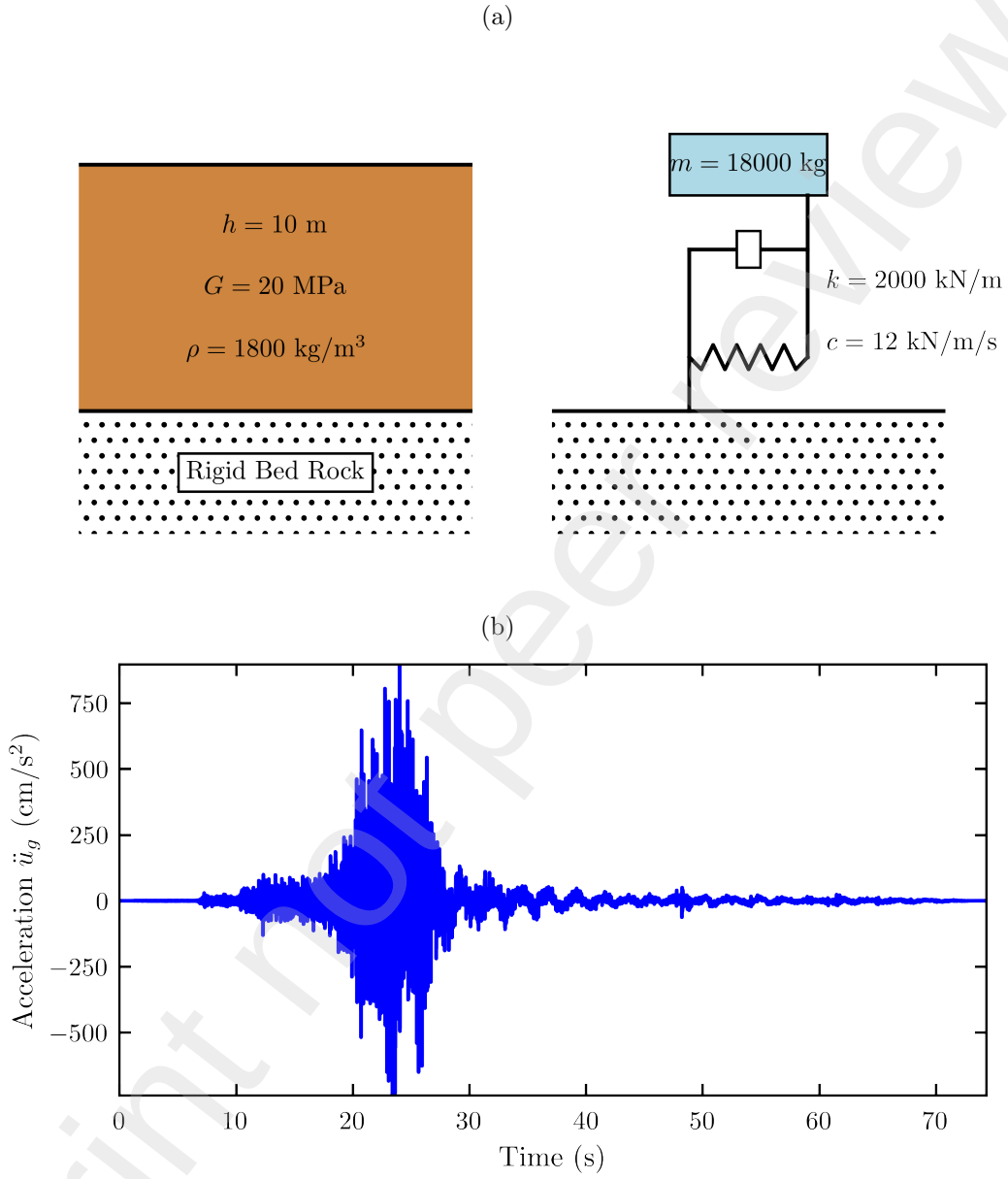

Figure 5: 1D single-layer seismic site response problem considered to illustrate the challenges in PINNs. (a) Soil configuration and lumped mass formulation (b) Acceleration time series of input ground motion.

4.1 Spectral Bias and Fourier Feature Embedding

As previously discussed, the transient nature and wide frequency range of ground motion excitation impose challenges for training PINNs. We illustrate this with a PINN based on a 4-layer perceptron applied to the system shown in Figure 5(a) and the excitation depicted in Figure 5(b). The choice of a 4-layer perceptron is motivated by its prevalent use in PINN applications ((Cai et al., 2021; Wang et al., 2019, 2022)). Figure 6 compares the PINN predictions with a numerical solution obtained using the Newmark-beta (NB) method (Chopra, 2007). The results reveal that the perceptron exhibits a slow decrease in Root Mean Squared Error (RMSE) and requires extensive iterations to capture the desired frequency content. For example, after 5000 iterations, the frequency content of the numerical solution remains inadequately recovered, and the RMSE remains high (RMSE = 0.94). This phenomenon aligns with the concept of spectral bias (Rahaman et al. (2019)) in neural tangent kernel theory, where neural networks tend to more readily capture low-frequency components of a target function compared to high-frequency components (Jacot et al., 2018). Throughout the training, neural networks initially grasp the overall coarse-grained data structure. As training progresses, they begin to fit finer, more intricate data details, often corresponding to high-frequency components. This progressive learning process can impact both the convergence rate and the final performance of the model.

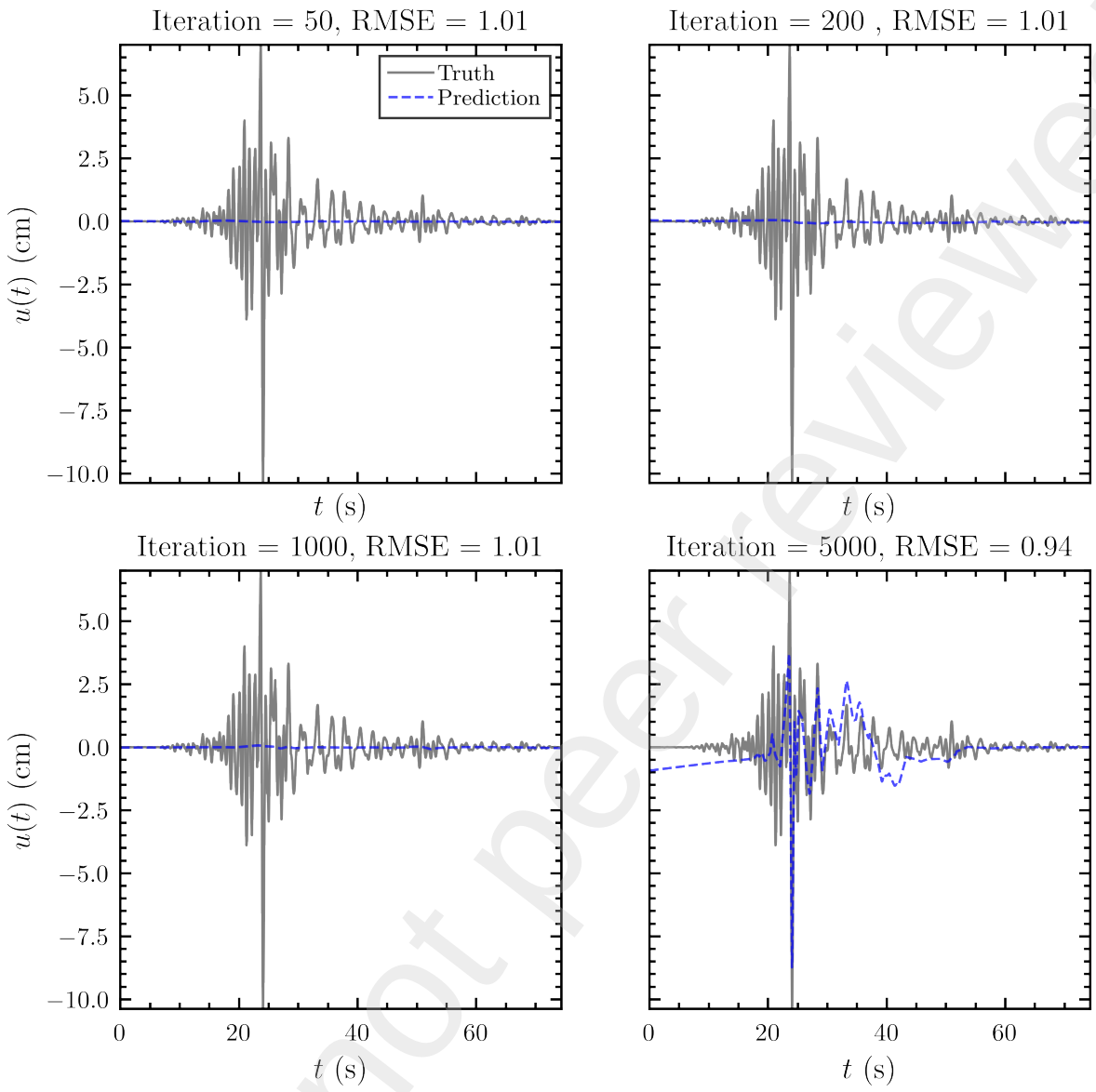

Figure 6: Training dynamics of a vanilla multi-layer perceptron in estimating the solution to the 1D site response problem shown in Figure 5. The 'truth' refers to the solution estimated using the NB method. The perceptron cannot capture the high-frequency components of the solution even after a large number of iterations.

We propose employing Fourier feature embedding recently developed by Tancik et al. (2020) for image reconstruction to mitigate spectral bias and enhance convergence speed. Fourier feature embedding is a method designed to augment the capability of neural networks to capture high-frequency components of the target function. This is achieved by transforming input features into a higher-dimensional space using sinusoidal functions, enabling the network to learn complex

patterns more effectively. Mathematically, Fourier feature embedding can be expressed as:

$$\gamma(\mathbf{v}) = \begin{bmatrix} \sin(2\pi \mathbf{B}\mathbf{v}) \\ \cos(2\pi \mathbf{B}\mathbf{v}) \end{bmatrix}^T \tag{15}$$

where $\mathbf{v}$ is the input feature vector (or the scalar t in the site response problem) of length k and $\gamma(\mathbf{v})$ is the embedded Fourier feature vector with a length of m. $\mathbf{B}$ is a m by k matrix of frequencies, sampled from $\mathcal{N}(\mathbf{0}, \sigma^2 I)$. The hyperparameters m and σ are pre-specified before the neural network training. Figure 7 shows the same perceptron as in Figure 6, but augmented with $m = 100$ Fourier features and $\sigma = 0.6$ for the training process. As Figure 7 demonstrates, incorporating Fourier features can significantly expedite PINN training. For example, after 50 iterations, the RMSE becomes negligible, and the predictions align well with the numerical solution.

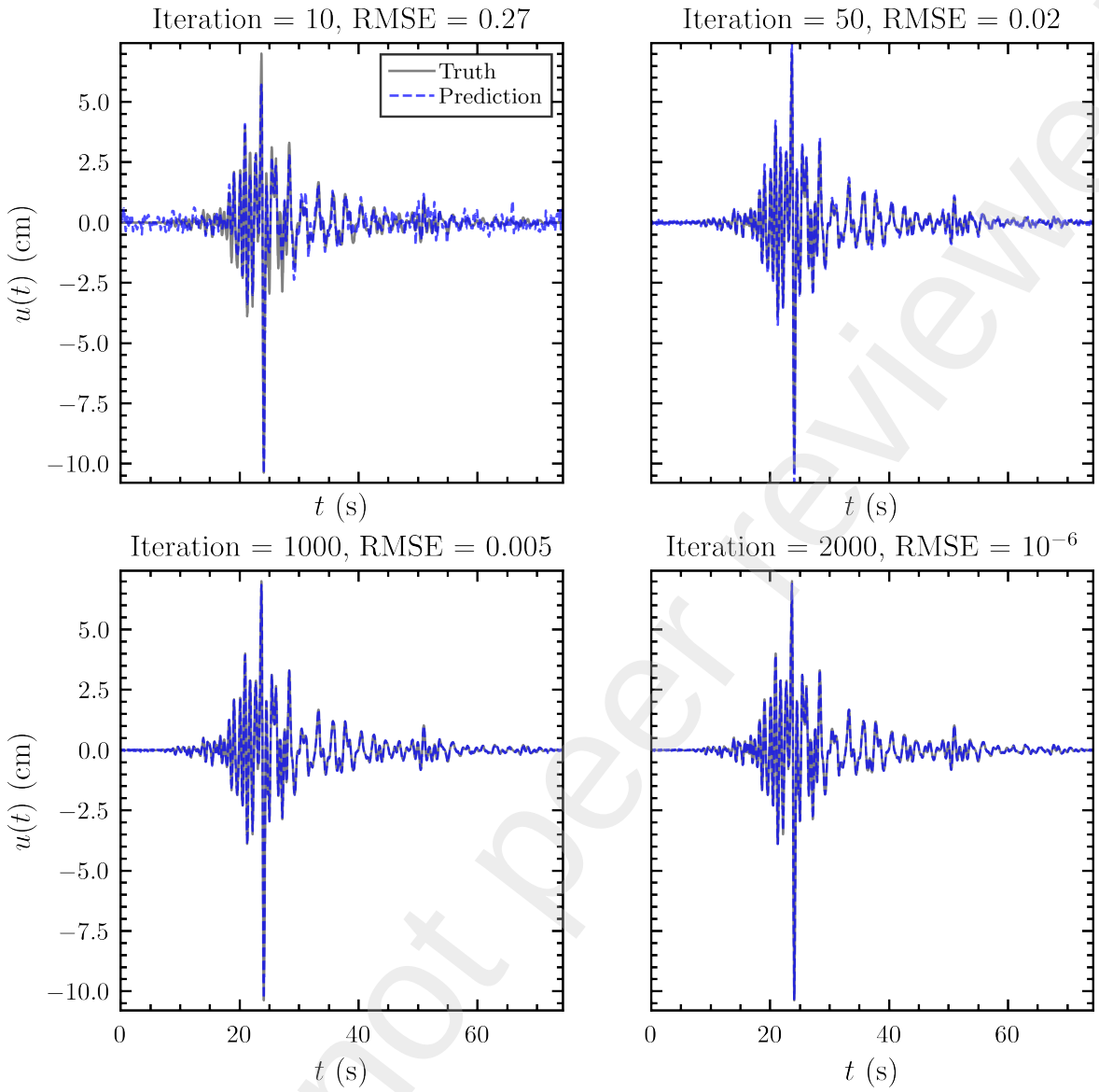

Figure 7: Training dynamics of a multi-layer perceptron with Fourier feature embedding in estimating the solution to the 1D site response problem shown in Figure 5. The 'truth' refers to the solution estimated using the NB method. The perceptron with Fourier feature embedding can approximate the truth in a few iterations.

According to Tancik et al. (2020), increasing the value of m allows neural networks to capture a broader range of frequency components, though this comes at the cost of more computational time. Figure 8 (left column) illustrates the estimation of $u(t)$ for the site response problem shown in Figure 5. Using the same model architecture as in Figure 7 but with varying m and σ values and training the model for 1000 iterations, it is evident that increasing m enhances the performance of the PINN. Specifically, the RMSE for $m = 200$ is slightly better than that for $m = 100$, although the training time for $m = 200$ is longer because PINN training involves computing high-order

gradients. The parameter σ controls the frequency range covered by the Fourier features in the neural tangent kernel. A larger σ allows for the inclusion of higher frequency components, which helps the neural network learn finer details. Conversely, a smaller σ restricts the frequency range, resulting in overly smooth representations that may miss important details. Figure 8 (right column) demonstrates the impact of different σ values on PINN performance. It is observed that large and small σ values can degrade the performance of the PINN. For example, a σ of 1.5 can lead to highly oscillatory learned representations.

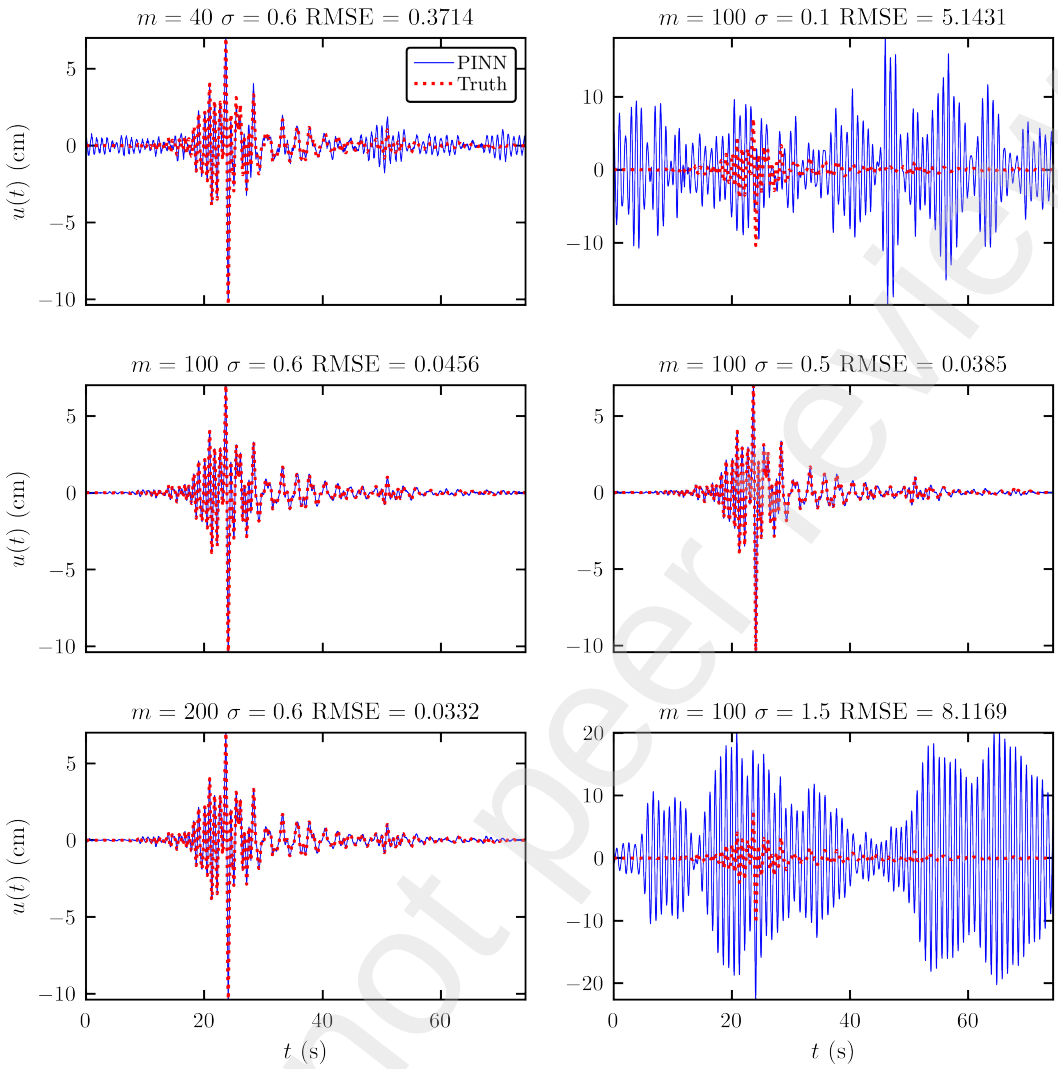

Figure 8: Influence of Fourier feature embedding parameters σ and m. The same perceptron as in Figure 7 is considered, and the PINN model is trained for 1000 iterations. The "Truth" is represented by the NB method's solution. Left column: increasing m enhances performance up to a point. In this case, an m value of 100 performs almost identically to 200. Right column: overly small and large σ values can degrade performance.

We recommend conducting a randomized search using a randomly split validation set to determine the optimal values for m and σ. Randomized search is a systematic method for optimizing hyperparameters. It involves randomly searching through a predefined set of hyperparameter values to identify the combination that yields an adequate performance according to a specified evaluation metric. Once optimal parameters are identified, the neural network should be retrained using

the entire dataset to avoid data loss from the validation split. Based on our assessments, the performance of PINNs tends to be more sensitive to σ than to m as illustrated by Figure 8.

4.2 Learning Rate

Training a neural network for traditional supervised tasks typically involves using a single loss function with a gradually decaying learning rate across iterations. In contrast, PINNs encounter more complex loss functions (Equation 13) due to components from the initial conditions and the residuals of the differential equations. Using a single learning rate becomes inadequate because the initial conditions and the residuals of the differential equations vary significantly in scale. To address this, each loss component is often weighted by a constant λ (Equation 13), but determining an appropriate λ can be challenging and require expert judgment. We recommend two methodologies to mitigate the impact of λ during PINN training. The first approach involves non-dimensionalizing the differential equations to normalize the scales of different loss components and enhance numerical stability. Non-dimensionalization entails normalizing both dependent and independent variables by characteristic scales. For example, the differential equation for a single-layer 1D seismic site response problem can be written as :

$$\ddot{u}(t) + 2\zeta\omega\dot{u}(t) + \omega^2 u(t) = -\ddot{u}_g(t) \tag{16}$$

where $\omega = \sqrt{\frac{K}{M}}$, and $\zeta = \frac{C}{2M\omega}$ with initial condition $\dot{u}(0) = u(0) = 0$. M, C, and K are the mass, damping coefficient, and stiffness of the soil layer. By considering $t = ay$ and $u = bz$, where $a = \frac{1}{w}$ and $b = -a^2$ are reference values and y and z are dimensionless quantity, the non-dimensionalized equation becomes:

$$z''(y) + 2\zeta z'(y) + z(y) = \ddot{u}_g(ay) \tag{17}$$

where $z' = \frac{dz}{dy}$ with $z'(0) = z(0) = 0$. This helps reduce the impact of the different scales of the differential equation residuals and initial conditions on the training of the PINN. Specifically, in the case of this study, we find that considering $\lambda = 1$ when the differential equation is non-dimensionalized gives satisfactory performance, as illustrated in the next section. In applications where λ has to be tuned, a more sophisticated method, discussed by Wang et al. (2022) could be considered. In this method, λ is dynamically adjusted based on the eigenvalues of the neural tangent kernel at different iterations. This adaptive calibration of λ ensures that the contributions of various loss components to gradients are balanced across iterations, thereby mitigating gradient

imbalances during training.

4.3 Other Hyperparameters

Regarding the initialization of PINN parameters, we recommend employing the Xavier or Kaiming initialization schemes, as they offer superior convergence properties compared to simple uniform or normal initialization methods. This is also described described in Narkhede et al. (2022). For activation functions, we examined different options, including Sigmoid, Tanh, ReLU, Leaky ReLU, and GELU (see Goodfellow et al. (2016) for their definitions) and recommend using GeLU or Tanh over other activation functions as their performance is adequate (performance discussions are presented in Section 6). In particular, we recommend not using ReLU because its second-order gradient is zero, potentially leading to saturation in the computation of differential equation residuals.

In terms of optimizers, we recommend using the Adam optimizer (Kingma, 2014) over other potential alternatives such as SGD, LBFGS, RMSprop, and Adagrad (see Paszke et al. (2019) for their definitions) as it showed an effective performance as evidenced by the results in Section 6. The Adam optimizer is a gradient descent algorithm that adjusts the learning rate for each parameter individually based on the first (mean) and second (uncentered variance) moments of the gradients, making it efficient and well-suited for many deep learning applications (Kingma, 2014). Regarding the width and depth of the PINN, our assessments indicated that narrow and shallow networks may result in limited performance. Conversely, wide and deep networks may encounter optimization difficulties due to issues with gradient propagation, a challenge consistent with findings reported by Goodfellow et al. (2016). Based on our assessments, we recommend network widths ranging from 50 to 500 and depths between 3 to 6 layers. Last, incorporating batch normalization improves training stability and speed by mitigating internal covariate shifts.

5 Final PINN Architecture

Figure 9 depicts the PINN architecture proposed in this study. The architecture considers feature embedding, three fully connected layers, batch normalization, and the Tanh activation functions. The workflow illustrating the training process and application of the PINN proposed in this study is shown in Figure 10. First, the ground motion acceleration and its corresponding timestamps are

divided into a training set (70%) and a validation set (30%) for hyperparameter optimization. The ordinary differential equations governing the problem are non-dimensionalized (i.e., Equation 13). In this study, λ was set to 1 without issues in the optimization process. The tree-structured Parzen (TPE) estimator (Ozaki et al., 2022) is used to obtain samples of hyperparameters during randomized hyperparameter search. Then, Fourier feature embedding is applied, and loss computations proceed. The ADAM optimizer with a learning rate of 0.01, reduced by a factor of 0.5 when the loss is not improved over the last 50 iterations, provided suitable performance. For Fourier feature embedding, the m and σ parameters are evaluated as part of the training process. The dashed box in Figure 10 highlights the hyperparameter tuning steps. Once these steps are completed, the optimized hyperparameters and the entire dataset (ground motion acceleration and timestamps) are used to retrain the PINN.

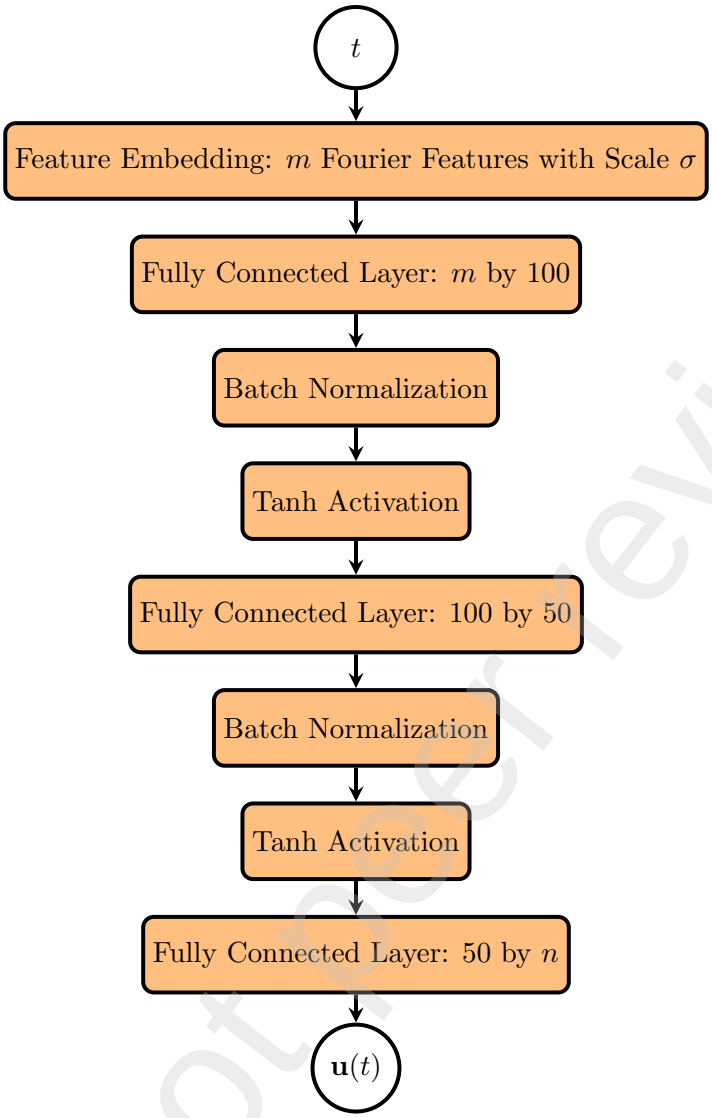

Figure 9: Architecture of the PINN proposed in this study for seismic site response analyses. The input t is the timestamp of the input ground motion time history. The output $\mathbf{u}(t)$ is the vector of nodal displacements for the n soil layers (for the single-layer case ($n = 1$), the PINN output is a scalar value of displacement at the surface of the soil layer).

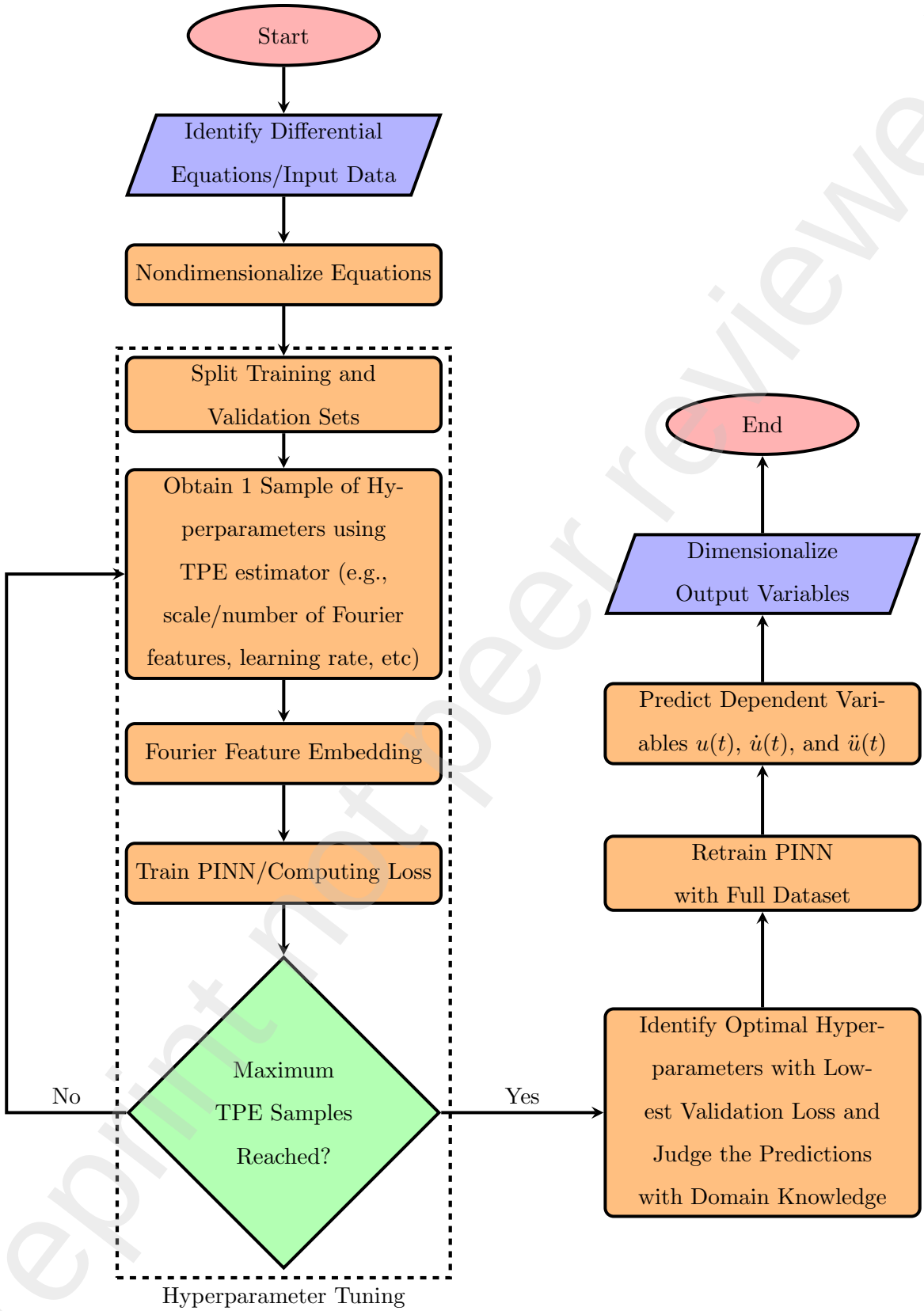

Figure 10: Flowchart illustrating the training and application of the PINN proposed for seismic site response in this study. The hyperparameters are sampled using the Tree-Structured Parzen Estimator (TPE) to facilitate the optimization. Each set of hyperparameters is evaluated based on validation loss until the maximum number of samples is reached.

6 Results

In this section, we first compare the results of the PINN with two traditional numerical algorithms for solving Equation 10 for the 1D, single-layer site response problem shown in Figure 5. Given that one of the main challenges of using the proposed PINNs for site response problems is the wide frequency range of the solutions, we evaluate the effectiveness of the proposed PINN by considering different soil properties and input ground motions encompassing a range of predominant frequencies. In addition to the soil profile and ground motion depicted in Figure 5, we conducted nine additional scenarios by varying the shear modulus (G, from very soft to very stiff) and selecting input ground motions with different mean period (T_m) values selected from the NGA-West2 ground motion database (Bozorgnia et al., 2014). The considered nine scenarios are summarized in Table 1. For brevity, we detail only the results obtained from the configuration in Figure 5, with additional results available in the electronic supplement.

Table 1: Different soil properties and input ground motions considered in this study.

case	G (kPa)	T_m (s)
1	5,000	0.2
2	5,000	0.5
3	5,000	1.0
4	50,000	0.2
5	50,000	0.5
6	50,000	1.0
7	200,000	0.2
8	200,000	0.5
9	200,000	1.0

The traditional algorithms used to compare with PINNs are the explicit Runge-Kutta method of order 5 (RK45) and the Newmark-beta (NB) method (Chopra, 2007). The RK45 algorithm uses an adaptive step size approach, dynamically adjusting the integration steps based on local solution behavior to balance accuracy and computational efficiency. With its automatic time-stepping feature, RK45 is commonly used in scientific computing (Butcher, 1996). The NB method is an implicit, unconditionally stable method commonly used for solving problems in earthquake engineering that involve the motion equation (Chopra, 2007).

Figure 11 presents the computed acceleration, velocity, displacement, and acceleration response spectrum (PSA) at the top soil layer surface using the proposed PINN against the RK45 and NB methods . The results demonstrate close agreement among the three methods. Notably, the presence of high-frequency ground motions influences the differential equation solution, resulting in highly oscillatory responses. Without Fourier feature embedding, learning such functions would pose substantial challenges for a conventional multi-layer perceptron, as already demonstrated in Figures 6 and 7.

It's noteworthy that the PINN learns a continuous and differentiable representation of the solution, facilitating the direct computation of $\ddot{u}(t)$ and $\dot{u}(t)$ by differentiating the PINN output ($u(t)$) using automatic differentiation with analytic gradients stored in the computational cycle (Paszke et al., 2019). In contrast, RK45 and NB methods rely on discrete integration, necessitating additional steps to estimate $\ddot{u}(t)$ (by solving Equation 10 after substituting estimated $u(t)$ and $\dot{u}(t)$), since performing a finite difference differentiation on $u(t)$ or $\dot{u}(t)$ to obtain $\ddot{u}(t)$ may cause numerical instabilities.

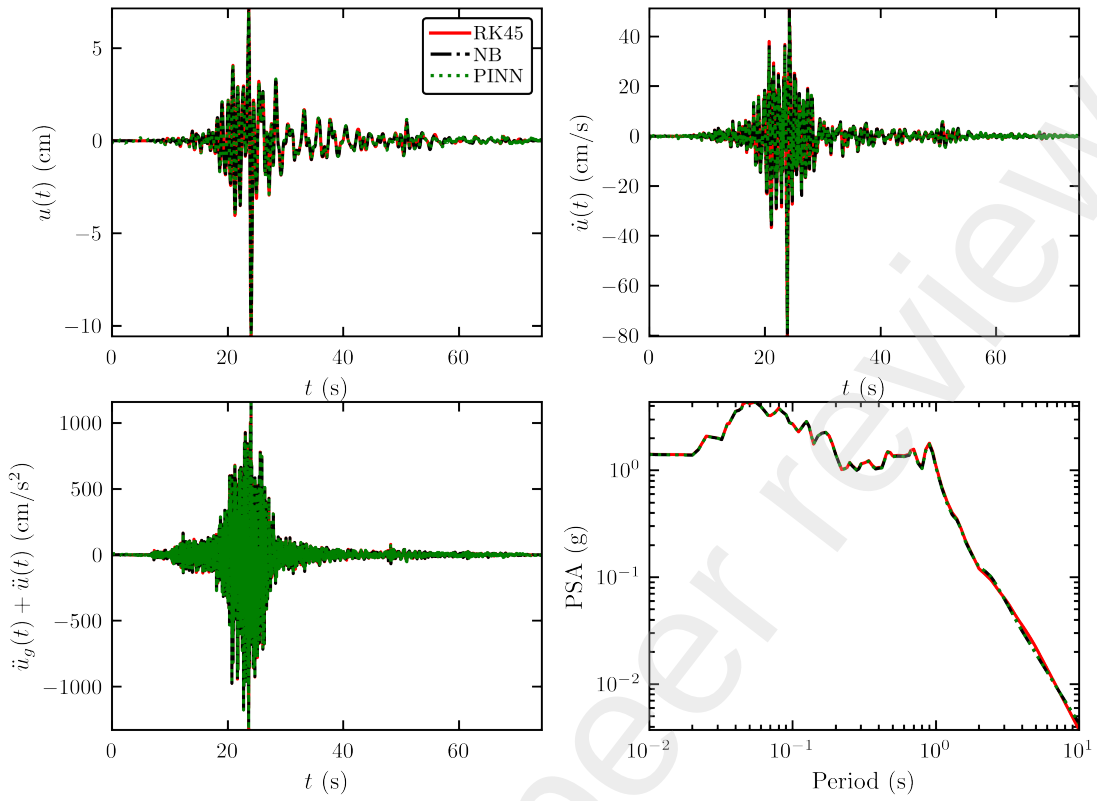

Figure 11: Comparisons of solutions obtained with the proposed PINN and the RK45 and NB methods for the configuration in Figure 5. The results are, for practical purposes, identical.

Table 2 provides the RMSE for the three methods considered. The RMSE values are calculated relative to the RK45 and NB methods. The results indicate nearly identical performance with negligible errors comparable to floating-point precision. Figure 12 shows the distribution of RMSE values, considering the nine additional cases listed in Table 1 (additional plots can be found in the electronic supplements). The results demonstrate that the solutions obtained from the PINN are, for practical purposes, identical to those from the traditional numerical methods. This alignment is expected, as the proposed PINN is designed to capture the embedded physics by satisfying the equations of motion and initial conditions. As a result, its efficacy is not dependent or sensitive to variations in soil properties or input ground motions, given that the PINN is properly trained using the proposed strategies outlined in the previous section. Notably, the RMSE for $\ddot{u}(t)$ is higher than for $\dot{u}(t)$, and the RMSE for $\dot{u}(t)$ is higher than for $u(t)$. This occurs due to error propagation in the estimation process. Both PINN and traditional methods first estimate $u(t)$, followed by $\dot{u}(t)$, and then $\ddot{u}(t)$. Consequently, differences in estimating $u(t)$ propagate to $\dot{u}(t)$ and $\ddot{u}(t)$, leading to

the observed pattern; however, the differences are still negligible.

Table 2: RMSE of $u(t)$, $\dot{u}(t)$, $\ddot{u}(t)$ estimated using PINN with respect to the RK45 and NB methods.

$u(t)$ (cm)		$\dot{u}(t)$ (cm/s)		$\ddot{u}(t)$ (cm/s^2)	
RK45	NB	RK45	NB	RK45	NB
5.35×10^{-8}	3.43×10^{-8}	5.79×10^{-7}	5.26×10^{-7}	3.57×10^{-6}	3.12×10^{-6}

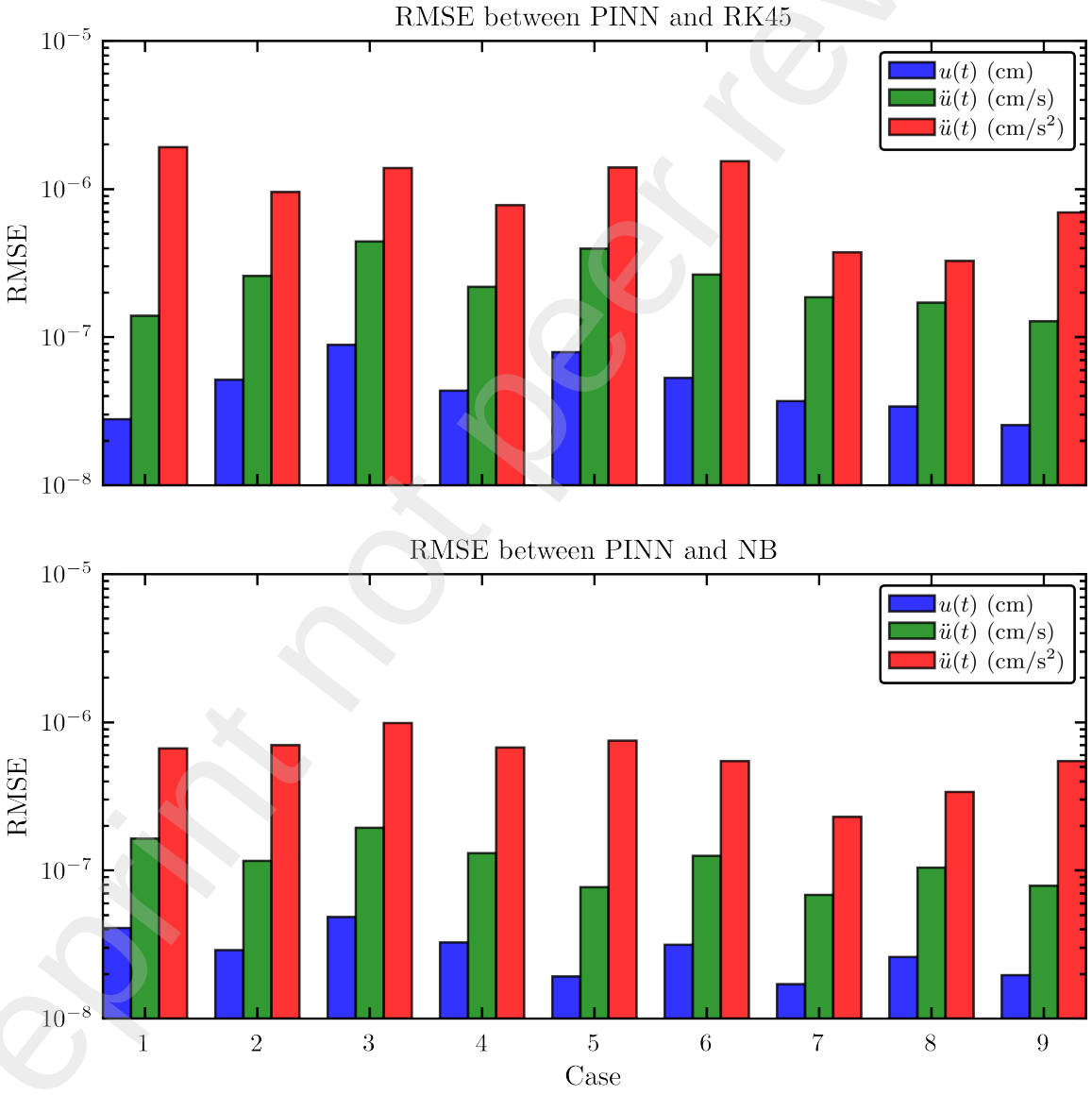

Figure 12: RMSE of the PINN with respect to the RK45 and NB methods considering the cases shown in Table 1.

In addition to the single-layer site response example, we also evaluate the effectiveness of the proposed PINN in solving multi-layer site response problems. For illustration, we consider a three-layer scenario, as depicted in Figure 3 (i.e., $n = 3$ in this case). The ground motion is the same as in Figure 5(b), with soil properties characterized by layer thicknesses of 10, 5, and 5 m; shear moduli (G) of 10,000, 5,000, and 15,000 KPa; unit weights of 15, 17, and 15 kN/m^3; and damping ratios (ζ) of 0.1, 0.1, and 0.2 for the three layers, respectively. Figure 13 presents the estimated acceleration ($\ddot{u}(t) + \ddot{u}_g(t)$) and the PSA response spectrum obtained from the proposed PINN and the RK45 and NB methods. Similar to the single-layer case, the results from the three procedures are consistent, underscoring the efficacy and generalization ability of the proposed PINN framework for multi-layer seismic site response problems.

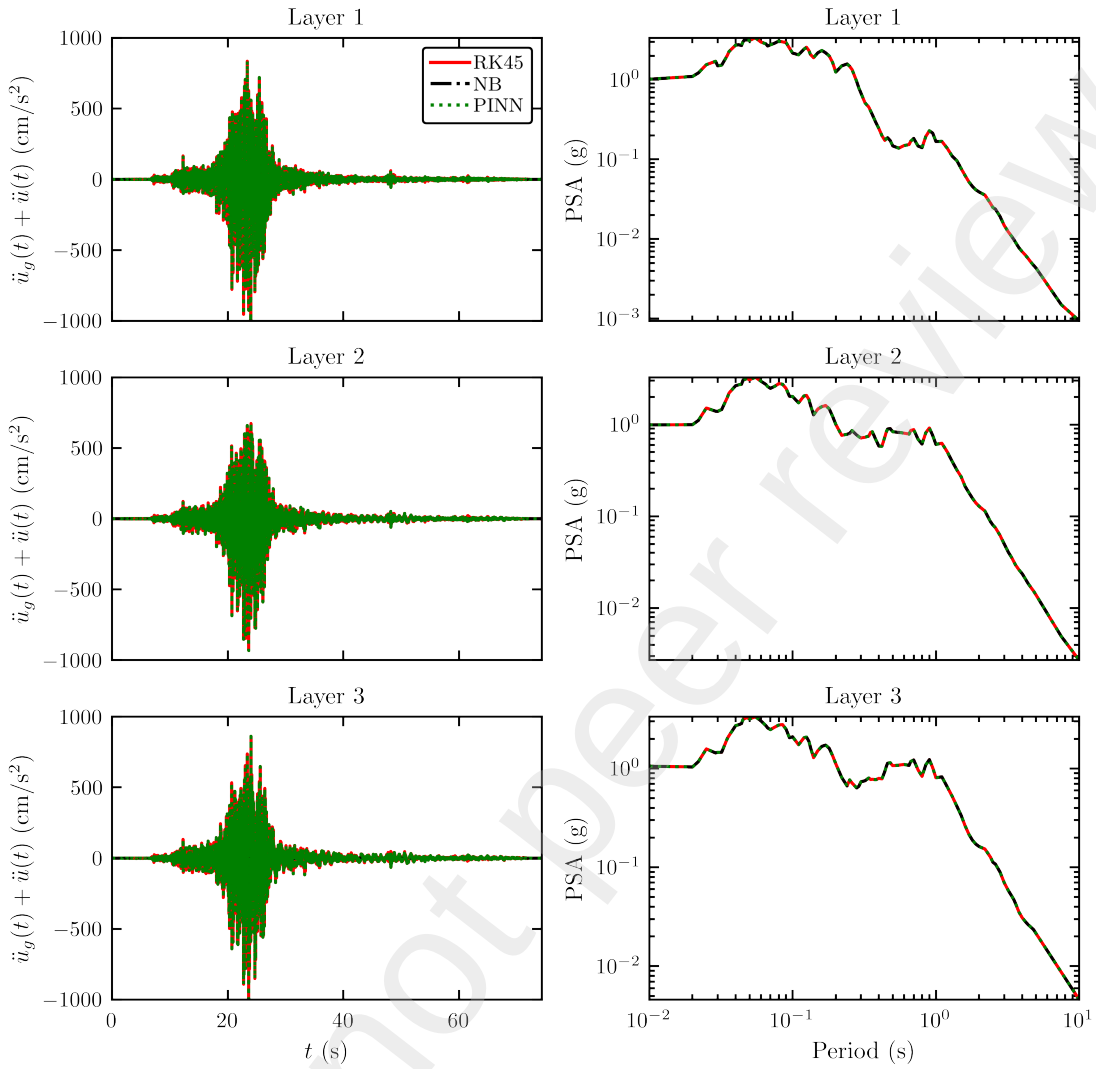

Figure 13: Comparisons of solutions obtained with the proposed PINN and the RK45 and NB methods considering a 3-layer system (i.e., $n = 3$ in Figure 3) with layer thicknesses of 10, 5, and 5 m; shear moduli (G) of 10,000, 5,000, and 15,000 KPa; unit weights of 15, 17, and 15 kN/m^3; and damping ratios of 0.1, 0.1, and 0.2, respectively. The results are, for practical purposes, identical.

7 Discussion

Although the discussions on the PINN-based framework for seismic site response analyses proposed in this study focused on linear analyses, applications to equivalent-linear analyses are straightforward. This is the case because equivalent-linear analyses are essentially a series of linear analyses;

hence, the proposed framework can readily be applied. Moreover, extensions to fully non-linear analyses can also be explored as part of future efforts. In this scenario, the differential equations are the same; thus, the framework proposed in this study is expected to work; however, additional information on the non-linearity of soil behavior would need to be incorporated into the training process.

The encouraging results from this study - i.e., the proposed PINN provides essentially the same results as established numerical solutions - suggest that there is also potential for general applications in GEE that can be explored in future efforts, with the added value that PINNs could bring potential advantages. For example, adapting PINNs to different differential equations usually requires minor modifications to their architecture, primarily by adjusting the number of parameters and loss functions (Cai et al., 2021). This contrasts with traditional numerical methods, which are often tailored to specific applications. For instance, it is known that numerical solutions in finite element solutions of GEE problems can find challenges in convergence or stability depending on the specific application (McKenna, 2011). Moreover, PINNs can also be less sensitive to mesh resolution compared to traditional numerical methods (Sharma and Shankar (2022); Wu and Lu (2022); Abueidda et al. (2021)), and the treatment of boundary conditions as highlighted by Cai et al. (2021) and discussed in Chandan et al. (2024) for the heat distribution problem. In general, the flexibility of PINNs to adapt to various problems by changing loss functions, the network width, and depth (e.g., see applications in Raissi et al. (2019)) has the potential to change the paradigms on how the governing equations in GEE applications are solved. Moreover, PINNs can combine observed data with physics-based knowledge to solve differential equations (e.g., Gao et al. (2023)). This is particularly useful in inversion problems in GEE, such as site response analysis, where ground motion data is available, but the system properties are unknown, as is often the case in earthquake scenarios (e.g., Moug et al. (2024)).

In closing, as discussed by Wang et al. (2019), it is also relevant to highlight that PINNs can leverage modern computing architectures such as GPUs and TPUs, benefiting from highly vectorized and parallelized operations. This makes PINNs scalable and well-suited for high-performance computing. PINNs are advancing rapidly, fueled by ongoing progress in deep learning research (Karniadakis et al., 2021). While traditional numerical methods have matured, PINNs continue to evolve, benefiting from the latest innovations in neural network architectures, optimization techniques, and computational resources. As deep learning research progresses, PINNs hold significant potential to become a fundamental tool in the numerical solution of differential equations.

8 Conclusions

This study is focused on assessing the potential of PINNs on 1D seismic site response analyses. Towards this end, existing challenges have been discussed, and a framework for incorporating PINNs addressing the identified challenges has been proposed. One of the key challenges is that the wide spectrum of frequencies present in the solutions of seismic site response differential equations necessitates tailored approaches in PINN architectures to expedite convergence during training. To address this, we propose employing a new technique, i.e., Fourier feature embedding, originally formulated for imaging processing. Additionally, we discuss other strategies that improve the robustness of the proposed PINN architecture, including learning rate adjustments, non-dimensionalization of differential equations, parameter initialization, and choice of activation functions, which enhance training stability and performance. The proposed PINN architecture is illustrated in Figure 9, and the overall framework is outlined in Figure 10. Specifically, hyperparameter tuning should focus on adjusting the parameters m and σ. A generally recommended approach is to explore m values within the range of 50 to 200, and σ values between 0.1 and 2. It is important to note that the performance of PINNs tends to be more sensitive to the choice of σ. The robustness of the discussed framework has been assessed by comparing results obtained with the proposed PINNs and numerical methods traditionally used for site response analyses, namely the Runge-Kutta and Newmark-beta methods. The results show that the proposed PINNs provide results identical to those obtained from the considered numerical methods. The results are encouraging as they highlight the PINN potential for GEE applications. In a general perspective, given the momentum of deep learning and physics-informed machine learning research, PINNs have the potential to become a new paradigm in GEE problems as they can bring advantages, including (1) the ability to incorporate data and physical principles to enhance performance, (2) the capability to learn a differentiable representation of solutions without domain discretization, (3) flexibility in handling boundary conditions and types of differential equations, (4) high parallelizability for efficient, high-performance computing, and (5) promising prospects for ongoing advancements, which contrast with traditional numerical approaches in GEE that have reached a plateau after $\tilde{8}0$ years of research.

References

Abueidda DW, Lu Q and Koric S (2021) Meshless physics-informed deep learning method for three-dimensional solid mechanics. *International Journal for Numerical Methods in Engineering* 122(23): 7182–7201.

Alimoradi A and Beck JL (2015) Machine-learning methods for earthquake ground motion analysis and simulation. *Journal of Engineering Mechanics* 141(4): 04014147.

Bozorgnia Y, Abrahamson NA, Atik LA, Ancheta TD, Atkinson GM, Baker JW, Baltay A, Boore DM, Campbell KW, Chiou BSJ et al. (2014) Nga-west2 research project. *Earthquake Spectra* 30(3): 973–987.

Brandolin F, Ravasi M and Alkhalifah T (2024) Pinnslope: seismic data interpolation and local slope estimation with physics informed neural networks. *Geophysics* 89(4): 1–61.

Butcher JC (1996) A history of runge-kutta methods. *Applied numerical mathematics* 20(3): 247–260.

Cai S, Mao Z, Wang Z, Yin M and Karniadakis GE (2021) Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica* 37(12): 1727–1738.

Chandan K, Saadeh R, Qazza A, Karthik K, Varun Kumar R, Kumar RN, Khan U, Masmoudi A, Abdou MMM, Ojok W et al. (2024) Predicting the thermal distribution in a convective wavy fin using a novel training physics-informed neural network method. *Scientific Reports* 14(1): 7045.

Chopra AK (2007) *Dynamics of structures*. Pearson Education India.

Dupuis M, Schill C, Lee R and Bradley B (2023) A deep-learning-based model for quality assessment of earthquake-induced ground-motion records. *Earthquake spectra* 39(4): 2492–2517.

Fukushima R, Kano M and Hirahara K (2023) Physics-informed neural networks for fault slip monitoring: Simulation, frictional parameter estimation, and prediction on slow slip events in a spring-slider system. *Journal of Geophysical Research: Solid Earth* 128(12): e2023JB027384.

Gao Y, Qian L, Yao T, Mo Z, Zhang J, Zhang R, Liu E and Li Y (2023) An improved physics-informed neural network algorithm for predicting the phreatic line of seepage. *Advances in Civil Engineering* 2023(1): 5499645.

Goodfellow I, Bengio Y and Courville A (2016) *Deep learning*. MIT press.

Guo XY and Fang SE (2023) Structural parameter identification using physics-informed neural networks. *Measurement* 220: 113334.

Idriss IM and Seed HB (1968) Seismic response of horizontal soil layers. *Journal of the Soil Mechanics and Foundations Division* 94(4): 1003–1031.

Idriss IM and Sun JI (1992) User's manual for shake91. *Center for Geotechnical Modeling, Department of Civil Engineering, University of California, Davis* .

Ilhan O, Harmon JA, Numanoglu OA and Hashash YM (2019) Deep learning-based site amplification models for central and eastern north america. In: *Earthquake Geotechnical Engineering for Protection and Development of Environment and Constructions*. CRC Press, pp. 2980–2987.

Jacot A, Gabriel F and Hongler C (2018) Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems* 31.

Ji K, Zhu C, Yaghmaei-Sabegh S, Lu J, Ren Y and Wen R (2023) Site classification using deep-learning-based image recognition techniques. *Earthquake engineering & structural dynamics* 52(8): 2323–2338.

Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S and Yang L (2021) Physics-informed machine learning. *Nature Reviews Physics* 3(6): 422–440.

Kazemi F, Asgarkhani N and Jankowski R (2023) Machine learning-based seismic response and performance assessment of reinforced concrete buildings. *Archives of Civil and Mechanical Engineering* 23(2): 94.

Kingma DP (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kramer SL (1996) *Geotechnical earthquake engineering*. Pearson Education India.

Laubscher R (2021) Simulation of multi-species flow and heat transfer using physics-informed neural networks. *Physics of Fluids* 33(8).

Lee YG, Kim SJ, Achmet Z, Kwon OS, Park D and Di Sarno L (2023) Site amplification prediction model of shallow bedrock sites based on machine learning models. *Soil Dynamics and Earthquake Engineering* 166: 107772.

Liu C and Macedo J (2022) Machine learning-based models for estimating seismically-induced slope displacements in subduction earthquake zones. *Soil Dynamics and Earthquake Engineering* 160: 107323.

Liu C and Macedo J (2024) Machine learning-based models for estimating liquefaction-induced building settlements. *Soil Dynamics and Earthquake Engineering* 182: 108673.

Macedo J, Liu C and Soleimani F (2021) Machine-learning-based predictive models for estimating seismically-induced slope displacements. *Soil Dynamics and Earthquake Engineering* 148: 106795.

Markham CS, Bray JD, Macedo J and Luque R (2016) Evaluating nonlinear effective stress site response analyses using records from the canterbury earthquake sequence. *Soil Dynamics and Earthquake Engineering* 82: 84–98.

Maurer BW and Sanger MD (2023) Why "ai" models for predicting soil liquefaction have been ignored, plus some that shouldn't be. *Earthquake Spectra* 39(3): 1883–1910.

McKenna F (2011) Opensees: a framework for earthquake engineering simulation. *Computing in Science & Engineering* 13(4): 58–66.

Moug DM, Bray JD, Bassal P, Macedo J, Ulmer K, Cetin KÖ, Kendır SB, Şahin A, Arnold C and Bikçe M (2024) Liquefaction-induced ground and building interactions in iskenderun from the 2023 kahramanmaraş earthquake sequence. *Earthquake Spectra* 40(2): 913–938.

Mousavi SM, Ellsworth WL, Zhu W, Chuang LY and Beroza GC (2020) Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications* 11(1): 3952.

Narkhede MV, Bartakke PP and Sutaone MS (2022) A review on weight initialization strategies for neural networks. *Artificial intelligence review* 55(1): 291–322.

Ozaki Y, Tanigaki Y, Watanabe S, Nomura M and Onishi M (2022) Multiobjective tree-structured parzen estimator. *Journal of Artificial Intelligence Research* 73: 1209–1250.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al. (2019) Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32.

Rahaman N, Baratin A, Arpit D, Draxler F, Lin M, Hamprecht F, Bengio Y and Courville A (2019) On the spectral bias of neural networks. In: *International conference on machine learning*. PMLR, pp. 5301–5310.

Raissi M, Perdikaris P and Karniadakis GE (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378: 686–707.

Rasht-Behesht M, Huber C, Shukla K and Karniadakis GE (2022) Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth* 127(5): e2021JB023120.

Rodríguez-Marek A, Bray JD and Abrahamson NA (2001) An empirical geotechnical seismic site response procedure. *Earthquake spectra* 17(1): 65–87.

Rumelhart DE, Hinton GE and Williams RJ (1986) Learning representations by back-propagating errors. *nature* 323(6088): 533–536.

Sharma R and Shankar V (2022) Accelerated training of physics-informed neural networks (pinns) using meshless discretizations. *Advances in Neural Information Processing Systems* 35: 1034–1046.

Tancik M, Srinivasan P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron J and Ng R (2020) Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems* 33: 7537–7547.

Tomar A and Burton HV (2021) Active learning method for risk assessment of distributed infrastructure systems. *Computer-Aided Civil and Infrastructure Engineering* 36(4): 438–452.

Van Nguyen D, Choo Y and Kim D (2024) Deep learning application for nonlinear seismic ground response prediction based on centrifuge test and numerical analysis. *Soil Dynamics and Earthquake Engineering* 182: 108733.

Wang S, Yu X and Perdikaris P (2022) When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* 449: 110768.

Wang YE, Wei GY and Brooks D (2019) Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701* .

Wu S and Lu B (2022) Inn: Interfaced neural networks as an accessible meshless approach for solving interface pde problems. *Journal of Computational Physics* 470: 111588.

Wu Y, Yin Z, Zhang H and Geng W (2023) Prediction of nonlinear seismic response of underground structures in single-and multi-layered soil profiles using a deep gated recurrent network. *Soil Dynamics and Earthquake Engineering* 168: 107852.

Xie Y, Ebad Sichani M, Padgett JE and DesRoches R (2020) The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra* 36(4): 1769–1801.

Zhu C, Cotton F, Kawase H and Nakano K (2023) How well can we predict earthquake site response so far? machine learning vs physics-based modeling. *Earthquake Spectra* 39(1): 478–504.