Laplace HypoPINN: physics-informed neural network for hypocenter localization and its predictive uncertainty

To cite this article: Muhammad Izzatullah *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 045001

View the article online for updates and enhancements.

MACHINE LEARNING
Science and Technology

PAPER

Laplace HypoPINN: physics-informed neural network for hypocenter localization and its predictive uncertainty

Muhammad Izzatullah[1,*] , Isa Eren Yildirim[1], Umair Bin Waheed[2] and Tariq Alkhalifah[1]

[1] King Abdulllah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
[2] King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia
* Author to whom any correspondence should be addressed.

E-mail: muhammad.izzatullah@kaust.edu.sa

Abstract

Several techniques have been proposed over the years for automatic hypocenter localization. While those techniques have pros and cons that trade-off computational efficiency and the susceptibility of getting trapped in local minima, an alternate approach is needed that allows robust localization performance and holds the potential to make the elusive goal of real-time microseismic monitoring possible. Physics-informed neural networks (PINNs) have appeared on the scene as a flexible and versatile framework for solving partial differential equations (PDEs) along with the associated initial or boundary conditions. We develop *HypoPINN*—a PINN-based inversion framework for hypocenter localization and introduce an approximate Bayesian framework for estimating its predictive uncertainties. This work focuses on predicting the hypocenter locations using HypoPINN and investigates the propagation of uncertainties from the random realizations of HypoPINN's weights and biases using the Laplace approximation. We train HypoPINN to obtain the optimized weights for predicting hypocenter location. Next, we approximate the covariance matrix at the optimized HypoPINN's weights for posterior sampling with the Laplace approximation. The posterior samples represent various realizations of HypoPINN's weights. Finally, we predict the locations of the hypocenter associated with those weights' realizations to investigate the uncertainty propagation that comes from those realizations. We demonstrate the features of this methodology through several numerical examples, including using the Otway velocity model based on the Otway project in Australia.

1. Introduction

In recent years, seismicity induced by anthropogenic activities including underground mining, geothermal exploitation, hydrofracturing, CO_2 geologic sequestration, and hydrocarbon production has resulted in a sharp increase in the number of earthquakes observed in historically quiet tectonic areas. In addition to causing considerable economic losses, such events are increasingly becoming a threat to public safety. A traffic light system (TLS) is typically implemented to manage and mitigate the associated hazard by reducing or suspending operations in case of observed seismicity beyond preset thresholds [1]. The success of a TLS protocol relies on real-time capabilities of detecting and locating these events.

Several techniques have been proposed over the years for automatic hypocenter localization (see [2] for an overview). Early methods relied on picked arrival times and estimated the unknown source location coordinates by using time difference as the objective function [3, 4]. These methods construct different objective functions to obtain absolute [5] or relative [6] locations. More recent approaches use full seismic waveforms to image the source location using a migration-type method [7–9]. Using full seismic waveforms to locate and image microseismic events allows for an automatic process bypassing the picking procedure that utilizes the full wavefield. However, this approach faces incredible nonlinearity due to the unknown source locations and it also uses mainly the transmission energy, which results in poor illumination and low

resolution. While these approaches have pros and cons that trade-off computational efficiency and the susceptibility of getting trapped in local minima, an alternate approach is needed that allows robust localization performance and holds the potential to make the elusive goal of real-time microseismic monitoring possible.

The confluence of ultrafast computers, rapid advancements in machine learning algorithms, and increasing availability of large datasets place seismology at the threshold of dramatic progress. Therefore, it is no surprise that several localization approaches have recently been proposed to harness the potential of supervised machine learning. These methods typically train a convolutional neural network (CNN) using historical or synthetically generated datasets [10, 11]. Once the CNN model is trained, it can be used to infer locations in real-time. Nevertheless, these methods typically require a huge amount of training data that may not be readily available. More importantly, due to the black-box nature of these models, it is not easy to gain insights into the features learned by the model.

Physics-informed neural networks (PINNs) have appeared on the scene as a flexible and versatile framework for solving partial differential equations (PDEs), along with any initial or boundary conditions [12]. Recently, researchers have explored the potential of PINNs as a fast travel time modeling engine for hypocenter localization [13, 14]. However, such approaches require source-receiver pairs as an input for PINN thus limits its applicability when the source locations are unknown. For other seismic applications, while [15, 16] used PINNs as a solver for the wave equation [17], proposed an approach for inverting wavefilled to retrieve the velocity model. Moreover [18], utilized PINNs to provide a framework to solve the eikonal equation and [19] extended the framework for seismic tomography. An important component of these solutions, especially when using the data as a boundary condition, is our confidence in their accuracy. There has been little study of PINN accuracy as an inversion tool. Neural networks naturally embed stochasticity through the random realization of weights and biases, thus, propagating uncertainties into its predictive solutions [20].
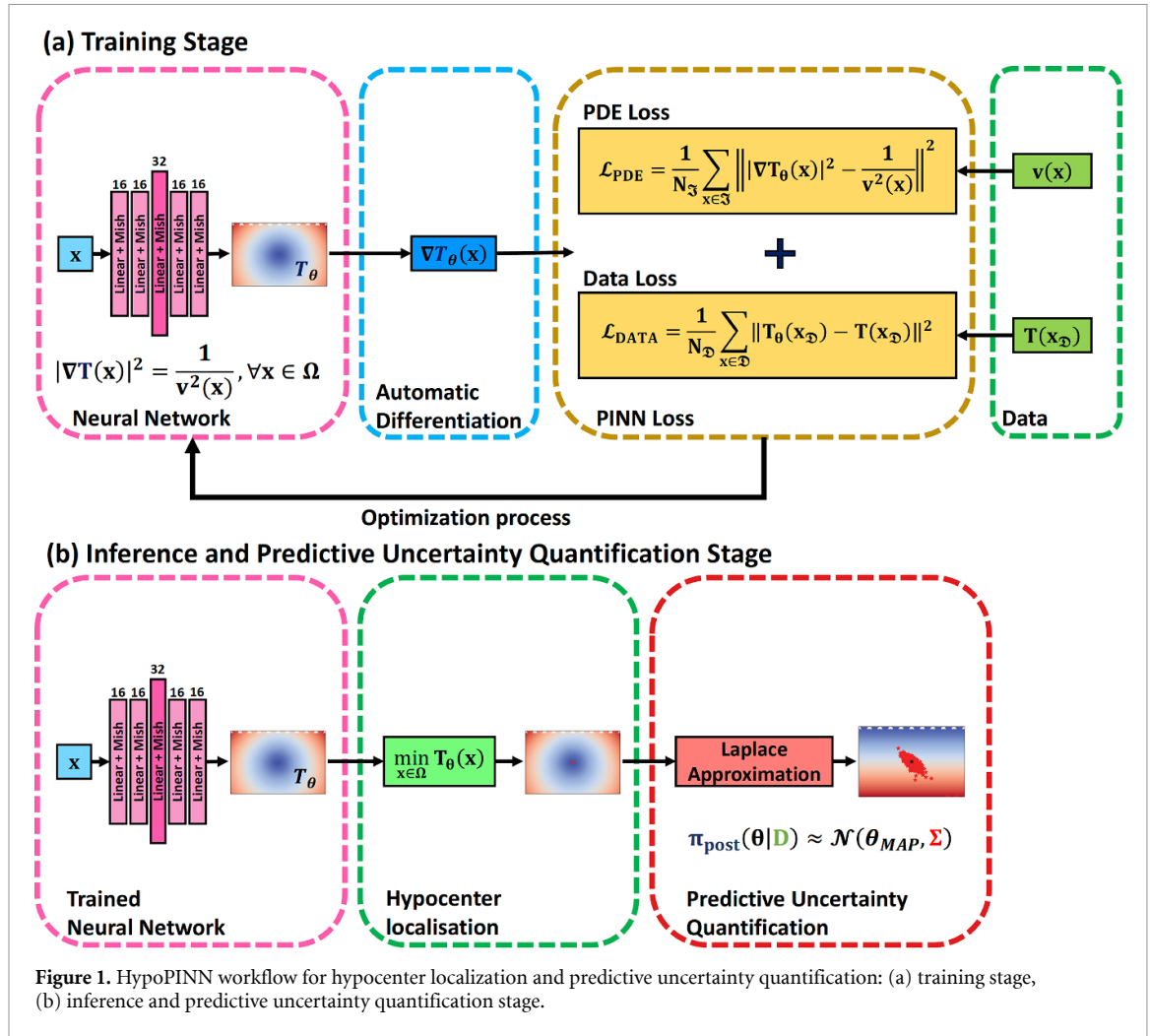
The contributions of this work to the field under study are as listed below.

- Inspired by the advancement of PINN, we develop a direct inversion framework for hypocenter localization using PINNs. Given picked arrival times for an event, we train a PINN model by minimizing a loss function formed by the misfit of observed and predicted travel times and the residual of the eikonal equation and obtain a travel time map for the entire computational domain. The hypocenter locations are then obtained by finding the location of the minimum travel time, representing the focusing location within the domain. As a result, we refer to this inversion framework as *HypoPINN*.
- We introduce an approximate Bayesian framework based on the Laplace approximation [21, 22] for estimating the predictive uncertainties of HypoPINN or, simply, forward modeling uncertainty in the context of HypoPINN. The Laplace approximation is arguably the simplest family of approximations for the intractable posteriors of deep neural networks. We approximate the covariance matrix at the optimized Hypo-PINN's weights for posterior sampling with the Laplace approximation. The posterior samples represent various realizations of HypoPINN's weights. Finally, we predict the locations of the hypocenter associated with those weights' realizations for predictive uncertainties quantification.
- We introduce a neural network architecture with an expansion-contraction design for HypoPINN in predicting the eikonal solution and localizing the hypocenter locations which provides more degrees of freedom in efficiently transforming the inputs (i.e. spatial coordinates) which belong to the euclidean space into polar coordinates space which represents the traveltimes map. This architecture is intuitively inspired by the emerging field of *geometric deep learning* [23].
- We analyze the effects of prior distribution in HypoPINN's weights realizations to investigate the uncertainty propagation that comes from the weights' realizations to the HypoPINN's solution.

The outline of the rest of the paper is as follows. First, we introduce the problem formulation of estimating hypocenter locations through PINN, which is the basis of HypoPINN and its predictive uncertainty studies. Next, we discuss the theoretical framework of the proposed approximate Bayesian framework for estimating predictive uncertainties in HypoPINN based on Laplace approximation. Then, we demonstrate the features of this methodology through several numerical examples, including using the Otway velocity model. Finally, we discuss the limitations and possible improvements of the proposed method before concluding the study.

2. Theoretical framework

We focus on developing a direct inversion framework for hypocenter localizations directly using PINNs by solving the eikonal equation and investigating its predictive uncertainties. Several works have shown

Figure 1. HypoPINN workflow for hypocenter localization and predictive uncertainty quantification: (a) training stage, (b) inference and predictive uncertainty quantification stage.

promising results in solving the eikonal equation through PINN [18, 19, 24, 25] by leveraging the factorization idea of eikonal equation [26]. However, this approach could not be extended to our work because the factorized eikonal equation requires the source locations to be known. Thus, we utilize the original eikonal equation together with the traveltime observations recorded at the surface to predict a traveltime map and by taking its minimum, we can directly localize the hypocenter locations.

In this section, we first introduce the eikonal equation and its PINN representation for the hypocenter localization problem. This is followed by a brief overview of PINN in the Bayesian framework and the Laplace approximation as an approximator for the intractable posterior distribution in quantifying its predictive uncertainty. Finally, putting these pieces together, we present the proposed algorithm for solving the hypocenter localization problem through PINN, which we refer to as *HypoPINN*. The full HypoPINN's workflow is illustrated in figure 1.

2.1. HypoPINN for hypocenter localization

Here, we outline the HypoPINN formulation for hypocenter localization through the eikonal equation. The eikonal equation is a non-linear, first-order, hyperbolic PDE of the form:

$$|\nabla \mathbf{T}(\mathbf{x})|^2 = \frac{1}{v^2(\mathbf{x})}, \quad \forall \mathbf{x} \in \Omega, \tag{1}$$

where Ω is a domain in $\mathbb{R}^d$ with d as the space dimension, $\mathbf{T}(\mathbf{x})$ is the travel time from the point-source $\mathbf{x}_s$ to any point $\mathbf{x}$, $v(\mathbf{x})$ is the velocity defined in Ω, and ∇ denotes the spatial differential operator. HypoPINN can be formulated as an optimization problem for the learnable PINN's parameters θ in approximating the traveltimes and estimating the hypocenter. The loss function for solving HypoPINN can be constructed using a mean-squared error (MSE) loss as:

$$\mathcal{L}(\theta) = \frac{1}{N_{\mathcal{I}}} \sum_{\mathbf{x}^* \in \mathcal{I}} |||\nabla \mathbf{T}_\theta(\mathbf{x}^*)|^2 - \frac{1}{v^2(\mathbf{x}^*)}||^2 + \frac{1}{N_{\mathcal{D}}} \sum_{\hat{\mathbf{x}} \in \mathcal{D}} ||\mathbf{T}_\theta(\hat{\mathbf{x}}) - \mathbf{T}(\hat{\mathbf{x}})||^2, \tag{2}$$

where $\mathbf{T}_\theta(\mathbf{x})$ represents the neural networks for the eikonal solution $\mathbf{T}(\mathbf{x})$. The first term on the right side of equation (2) imposes the validity of the eikonal equation as in equation (1) on a given set of training points $\mathbf{x}^* \in \mathcal{I}$, with $N_\mathcal{I}$ as the number of sampling points. The second term acts as data loss on a given set of travel time data at the receiver locations $\hat{\mathbf{x}} \in \mathcal{D}$, with $N_\mathcal{D}$ representing the number of receivers. We minimize the loss in equation (2) to obtain a good approximation of the eikonal solution and hypocenter location. Once the network is trained, the solution (microseismic hypocenter location) is basically the minimum of the computed travel time function, evaluated within the computational domain:

$$\mathbf{T}_{\theta^*_{\min}} = \min_{\hat{\mathbf{x}} \in \Omega} \{ \mathbf{T}_{\theta^*}(\hat{\mathbf{x}}) \}. \tag{3}$$

2.2. Laplace approximation for Bayesian PINNs

Bayesian framework for PINNs can be formulated through *unnormalized* Bayes' Theorem [20] which can be evaluated up to its normalizing constant as:

$$\mathbf{p}(\theta|\mathbf{D}) \propto \mathbf{p}(\mathbf{D}|\theta)\mathbf{p}(\theta) \approx \exp\left(-\mathcal{L}(\mathbf{D};\theta) \right), \tag{4}$$

where θ denotes the learnable PINN's parameters and $\mathbf{D}$ represents the dataset associated with PINN's training, e.g. observed data collected by seismic stations. Bayesian framework stated above introduces the notion of prior probability density and the likelihood function. The prior probability density $\mathbf{p}(\theta)$ encodes the confidence of the prior information on the unknown learnable PINN's parameters θ, whereas the likelihood function $\mathbf{p}(\mathbf{D}|\theta)$ describes the conditional probability density for the learnable PINN's parameters to generate observed data collected by seismic stations. Based on Bayes' theorem, we obtain the posterior probability density of the learnable PINN's parameters given the observed data, $\mathbf{p}(\theta|\mathbf{D})$, by combining the prior probability density and the likelihood function. The last term in equation (4) is known as the *Gibbs distribution* [27], a probability distribution which takes the form of exponential of negative loss function $\mathcal{L}(\mathbf{D};\theta)$. Due its form, we can transform the *Gibbs distribution* in equation (4) into a representation similar to the loss function in a deterministic setting by reformulating it in the *log-posterior* as follows:

$$\log \mathbf{p}(\theta|\mathbf{D}) \propto -\log \mathbf{p}(\mathbf{D}|\theta) - \log \mathbf{p}(\theta) \approx \mathcal{L}(\mathbf{D};\theta). \tag{5}$$

This transformation links the Bayesian framework with optimization procedure; thus, simplify the *Maximum-A-Posteriori (MAP)* computation. By minimizing equation (5), we obtain the MAP solution that we consider as the center of our Laplace approximation. The Laplace approximation uses a second-order expansion (Taylor expansion) of $\mathcal{L}(\mathbf{D};\theta)$ around θ_{MAP} to approximate $\mathbf{p}(\theta|\mathbf{D})$. We consider:

$$\mathcal{L}(\mathbf{D};\theta) \approx \mathcal{L}(\mathbf{D};\theta_{MAP}) + \frac{1}{2}(\theta - \theta_{MAP})^T \left(\nabla_\theta^2 \mathcal{L}(\mathbf{D};\theta_{MAP}) \right)(\theta - \theta_{MAP}), \tag{6}$$

and identify the Laplace approximation for $\mathbf{p}(\theta|\mathbf{D})$ as:

$$\mathbf{p}(\theta|\mathbf{D}) \approx \mathcal{N}(\theta_{MAP}, \Sigma), \quad \text{with} \quad \Sigma = -\left(\nabla_\theta^2 \mathcal{L}(\mathbf{D};\theta_{MAP}) \right)^{-1}. \tag{7}$$

Note that a naive implementation of the covariance matrix in equation (7) is infeasible, and it scales quadratically with the number of learnable PINN's parameters, θ. This work focuses on the diagonal approximation for the covariance matrix. Interested readers may refer to [21] for a detailed review on the scalable Laplace approximation for Bayesian neural networks. The diagonal approximation of the covariance matrix based on the Fisher information matrix $\mathbf{F}$ can be computed efficiently using automatic differentiation. It is simply the expectation of the squared gradients with respect to the network parameters θ:

$$\mathbf{H} \approx \text{diag}(\mathbf{F}) = \text{diag}\left(\mathrm{E}\left[\nabla_\theta \mathcal{L}(\mathbf{D};\theta) \nabla_\theta \mathcal{L}(\mathbf{D};\theta)^T \right] \right) = \text{diag}\left(\mathrm{E}\left[(\nabla_\theta \mathcal{L}(\mathbf{D};\theta))^2 \right] \right),$$

where 'diag' extracts the diagonal of a matrix. Note that, even if the expansion in equation (6) is accurate, this approximation will, unfortunately, place probability mass in low probability regions of the true posterior if some of the PINN's parameters θ exhibit high covariance. Despite the fact, it has been used successfully in neural network weights pruning and transfer learning [28]. Based on the diagonal approximation, we can approximate our covariance by:

$$\Sigma \approx \mathbf{H}^{-1} = \frac{1}{\text{diag}(\mathbf{F})}. \tag{8}$$

To apply the Laplace approximation for uncertainty estimation, we first minimize equation (5) to obtain the θ_{MAP}. We can transform the loss function in equation (2) into Bayesian framework as in equation (5) by reformulating it in the *log-posterior* form with a chosen log-prior distribution $\log \mathbf{p}(\theta)$ that commonly acts as a regularizer in a deterministic setting. Next, we approximate the covariance matrix at θ_{MAP} and construct the Laplace approximation of the posterior distribution as in equation (7). The posterior samples represent various realizations of PINN's weights, θ. Finally, we predict the solutions associated with those weights' realizations θ to investigate the uncertainty propagation that comes from those realizations.

3. Numerical examples

This section demonstrates the proposed methodology on numerical examples with a vertically varying velocity model of $2 \times 3 \, \text{km}^2$ that varies with depth and the Otway velocity model. In all numerical examples, we specially crafted a neural network architecture with an expansion-contraction design for HypoPINN in predicting the eikonal solution and localizing the hypocenter locations. The neural network contains 5 fully connected hidden layers and 16 neurons per layer except for the third layer with 32 neurons. Intuitively, by doubling the neurons in the middle layer provides the network with more degrees of freedom in efficiently transforming the inputs (i.e. spatial coordinates) which belong to the euclidean space into polar coordinates space which represents the traveltimes map. We consider *Mish* [29] as the activation function and Adam optimizer [30] as the optimization algorithm. The network architecture is illustrated in figure 1(a).

3.1. The vertically varying velocity model

We consider a vertically varying velocity model of $2 \times 3 \, \text{km}^2$ that varies with depth. The velocity at zero depth is $2 \, \text{km s}^{-1}$, and it increases linearly with a gradient of $0.5 \, \text{s}^{-1}$. We place a point-source at $(1, 1.5 \, \text{km})$ representing the microseismic event. The model is illustrated in figure 2(a) with the black star depicting the point-source location. The model is discretized on a 101×151 grid with a grid spacing of $20 \, \text{m}$ along both axes. The true traveltime map is computed analytically [31].

First, we investigate the effect of random initial weights initialization from four different prior distributions as a motivation towards understanding the hypocenter location prediction performance and the predictive uncertainties of HypoPINN. We consider the modified normal and uniform distributions introduced by [32, 33] for the initial weights. We randomly sample 2500 points in the domain and collect the travel time value at 11 receivers on the surface as data for training the HypoPINN. We minimize equation (5) along with equation (2) as the log-likelihood term and Gaussian prior (Tikhonov regularization). We perform the minimization for 3,000 epochs and predict the eikonal solution with the last epoch's weights, thus localizing the hypocenter location based on equation (3). The network took 59.93 s and 10.81 microseconds on average for training and inference, respectively.

In figure 3, we observe that the microseismic localizations from the eikonal solutions visually match the true location shown in figure 2(b) for all distributions except for the Xavier uniform prior. Its solution is biased towards observed data on the surface. We also localize the hypocenter location by taking the location of the eikonal solutions where its value is minimum, and relatively accurate hypocenter locations are estimated for the three previously mentioned prior distributions as denoted by the green star in the same figure. The hypocenter location coordinates are tabulated in table 1. Based on this experiment, we observed the effect of random weights initialization on the HypoPINN prediction as the eikonal solutions and hypocenter locations vary significantly with different initial weights realization. For example, the losses history in figure 4 for all the initial weights are showing convergence. Although the initial weights sampled from Xavier uniform distribution recorded the lowest total loss value, it failed to predict the eikonal solution correctly, thus localizing the hypocenter at the wrong location. This signifies the ill-posedness of this problem; HypoPINN with such prior weight distribution leads the training process to a bad local minimum, which belongs to the null space of the HypoPINN's solution space despite having the lowest total loss value.

Next, we repeat the experiment to study the HypoPINN's predictive uncertainties based on the Laplace approximation. Based on previous results, we consider Kaiming normal distribution as the prior distribution, and we initialize our initial weights by sampling from it. Similarly, in figure 5(a), we observe that the eikonal solution and its hypocenter location visually matched the analytical one. Based on this result, we take the last epoch's weights as our θ_{MAP} for the uncertainty analysis using the Laplace approximation.

To study the uncertainties propagations from the HypoPINN's weights to the predictive solution, we construct the Laplace approximation to the posterior distribution as in equation (7). This work considers the diagonal approximation of the covariance matrix described in the previous section with 2737 learnable network parameters. With this approximation, we sample 1000 weights' realizations and perform the eikonal
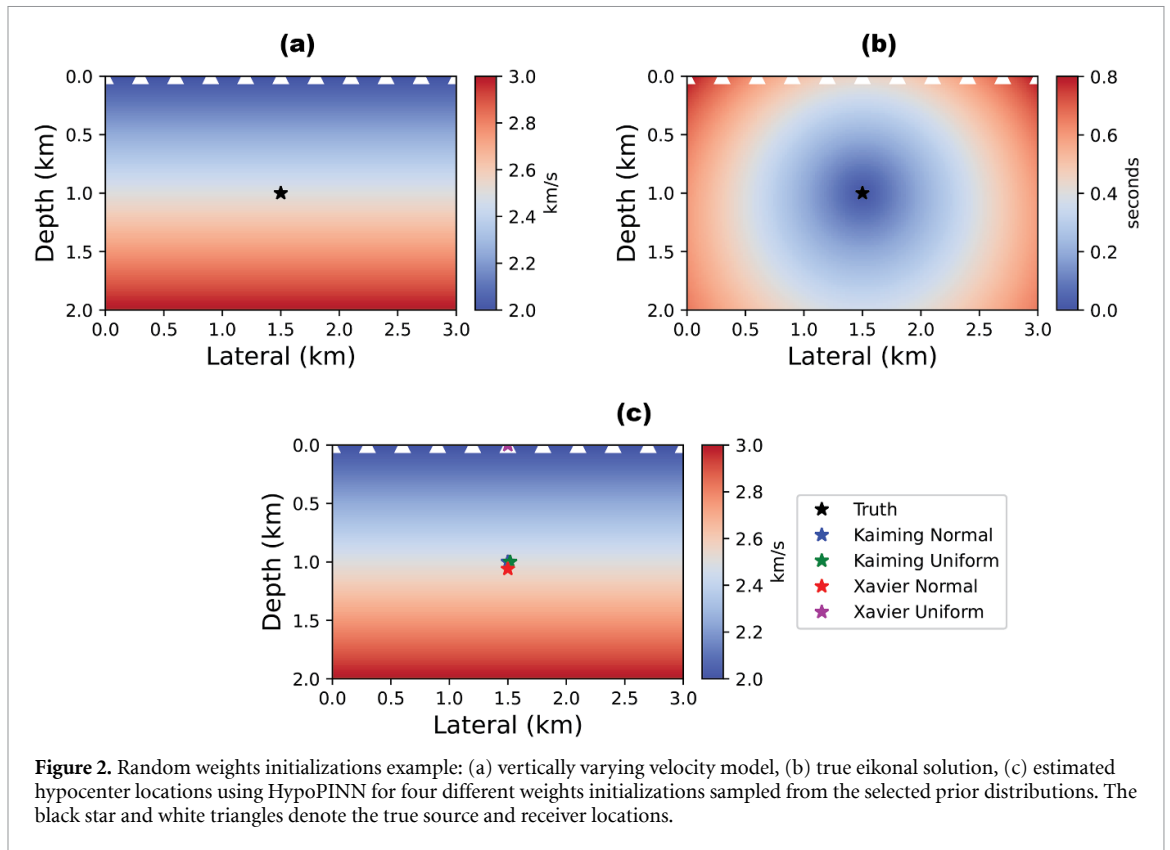
Figure 2. Random weights initializations example: (a) vertically varying velocity model, (b) true eikonal solution, (c) estimated hypocenter locations using HypoPINN for four different weights initializations sampled from the selected prior distributions. The black star and white triangles denote the true source and receiver locations.
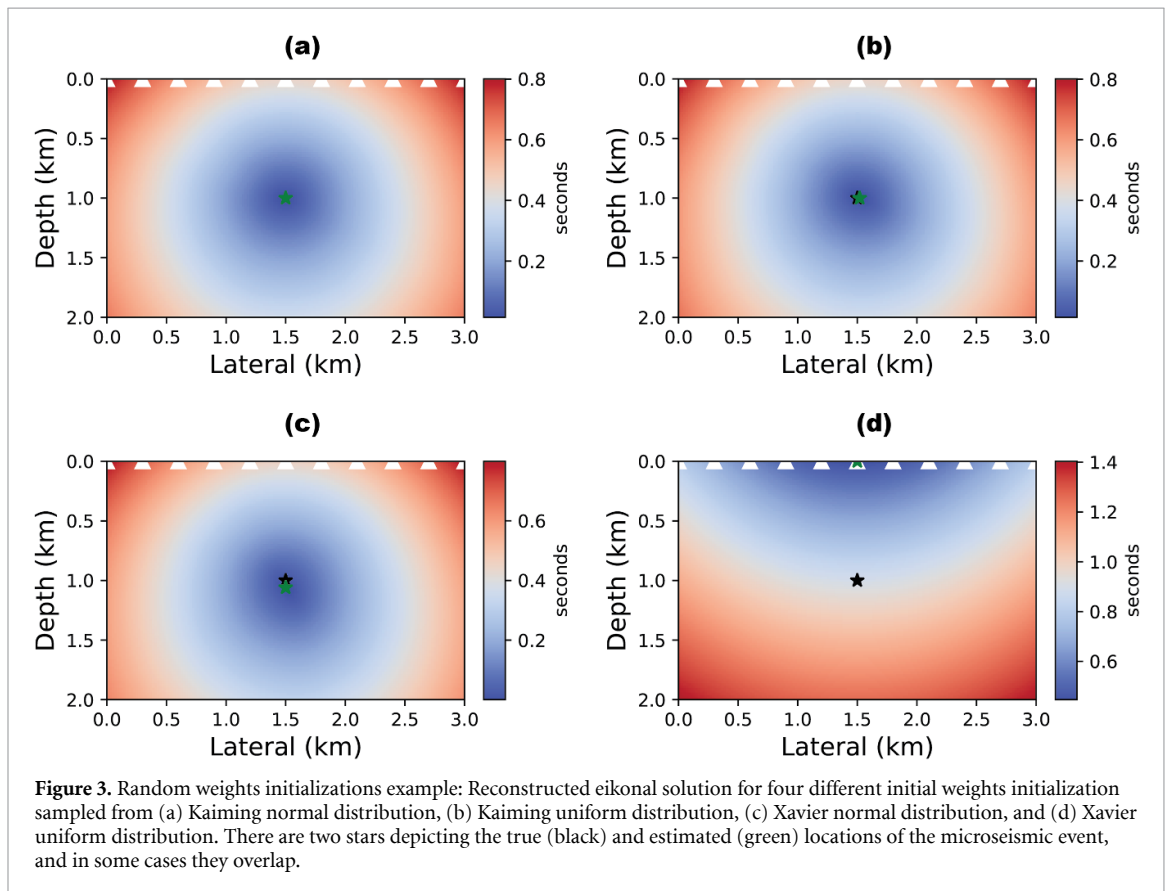
Figure 3. Random weights initializations example: Reconstructed eikonal solution for four different initial weights initialization sampled from (a) Kaiming normal distribution, (b) Kaiming uniform distribution, (c) Xavier normal distribution, and (d) Xavier uniform distribution. There are two stars depicting the true (black) and estimated (green) locations of the microseismic event, and in some cases they overlap.

solution predictions by realizing HypoPINN with those respective weights. Based on those realizations, we have 1000 predicted eikonal solutions, and we obtain the predicted hypocenter locations from those solutions. The results for this predictive uncertainty are illustrated in figure 5(b) and table 2. We observe that

Table 1. Random weights initializations example: hypocenter coordinates for four different initial weights initialization.

Prior weights initialization	Depth (km)	Lateral (km)
Truth	**1.000**	**1.500**
Kaiming Normal	1.000	1.500
Kaiming Uniform	1.000	1.520
Xavier Normal	1.060	1.500
Xavier Uniform	0.000	1.500

Figure 4. Random weights initializations example: convergence history of (a) data loss, (b) PDE loss, (c) total loss.
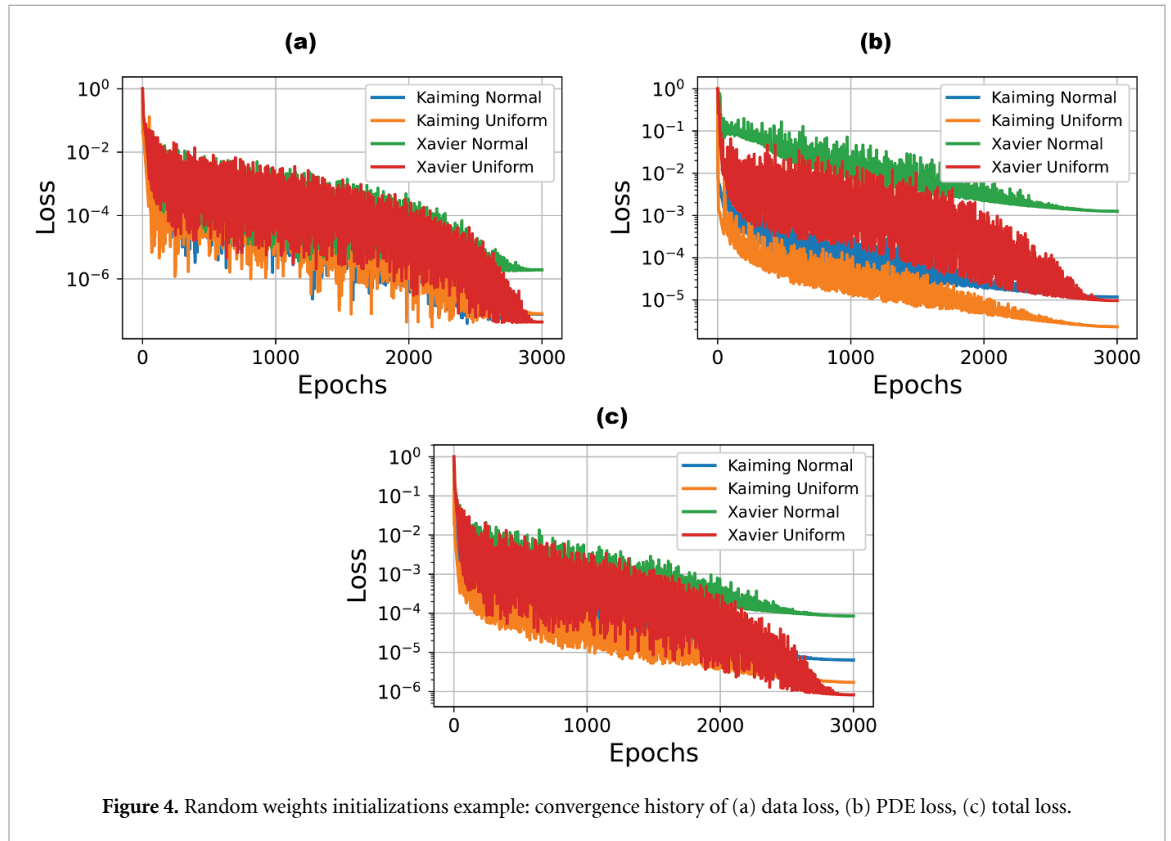
the predictions of the eikonal solution and the hypocenter locations vary significantly with different HypoPINN's weights realization, as shown in the previous example. This shows that the uncertainty in the HypoPINN's weights propagates into the predictive solution and significantly influences the prediction. In figure 5(b), we also observe the uncertainty of the hypocenter is larger in the depth direction as we use surface recordings. However, the predictive uncertainty depends on the loss landscape of the hypoPINN. A different loss landscape where the MAP solution lands will give a significantly different predictive uncertainty. For example, we could observe that several predicted hypocenter locations are completely far from the true location and no longer reflect the physical constraint. This shows us that the predictive uncertainty is sensitive to the hypoPINN's loss landscape. Interested readers may refer to [34, 35] to learn more about the loss landscape of neural networks and its effects on the network's predictive uncertainties. The examples shown here highlight critical features of this predictive uncertainty on the seismic hypocenter location.

3.2. Otway velocity model

This example considers the Otway velocity model which belongs to Stage 2C of the Otway project by CO2CRC Limited in Australia with 1.28 km depth and extended laterally for 1.70 km [36]. Multiple point-source locations at depth and lateral positions ranging from (0.7, 1.0 km), (0.75, 1.1 km), and (0.65, 0.9 km) are considered, respectively. The model is illustrated in figure 6 with the black stars depicting the point-source locations. The model is discretised on a 246×305 grid with a grid spacing of 8 m along both axes. We use the Fast Marching Method [37] to compute the eikonal solutions for each source location. The observed data are obtained from 6 regularly spaced receivers at the surface of the model.

For each source location in the model, an independent HypoPINN network with 2737 learnable network parameters is trained to estimate the hypocenter locations. The initial HypoPINN's weights are initialized by sampling from the Kaiming normal distribution. We randomly sample 5000 points in the domain for
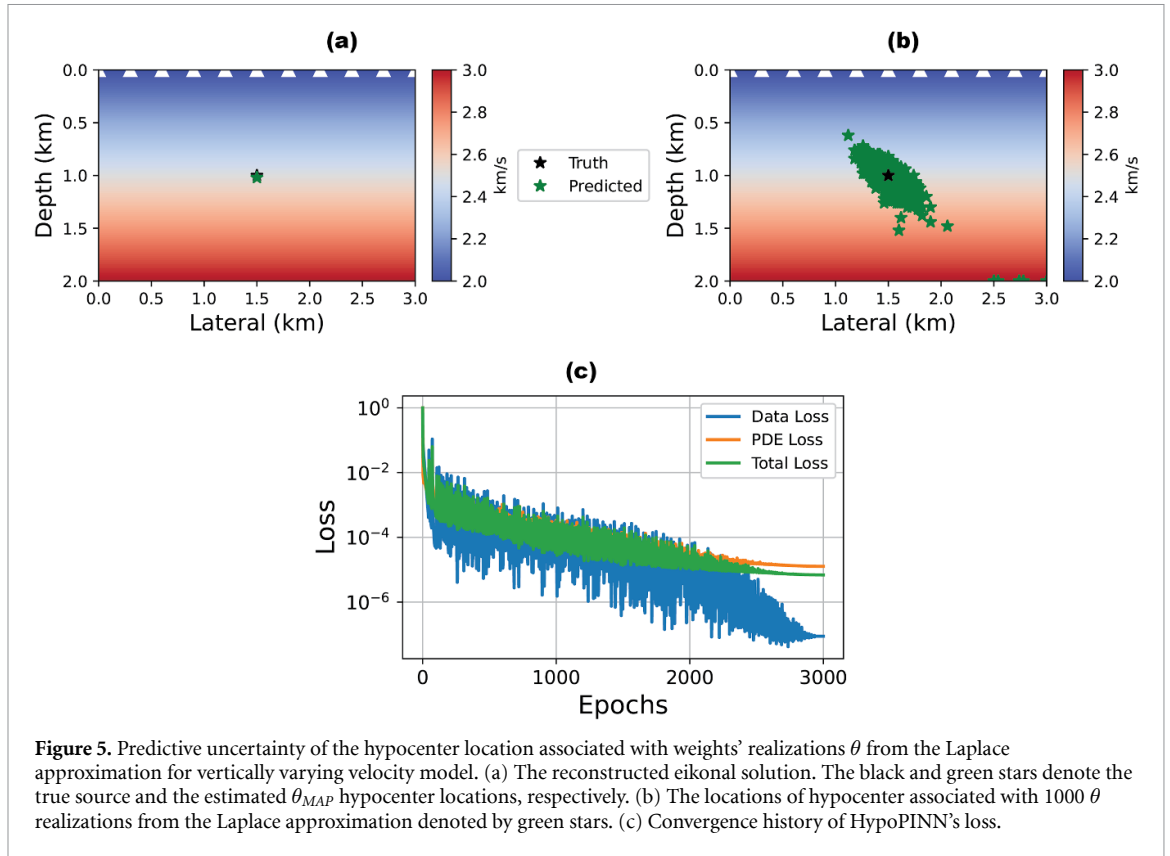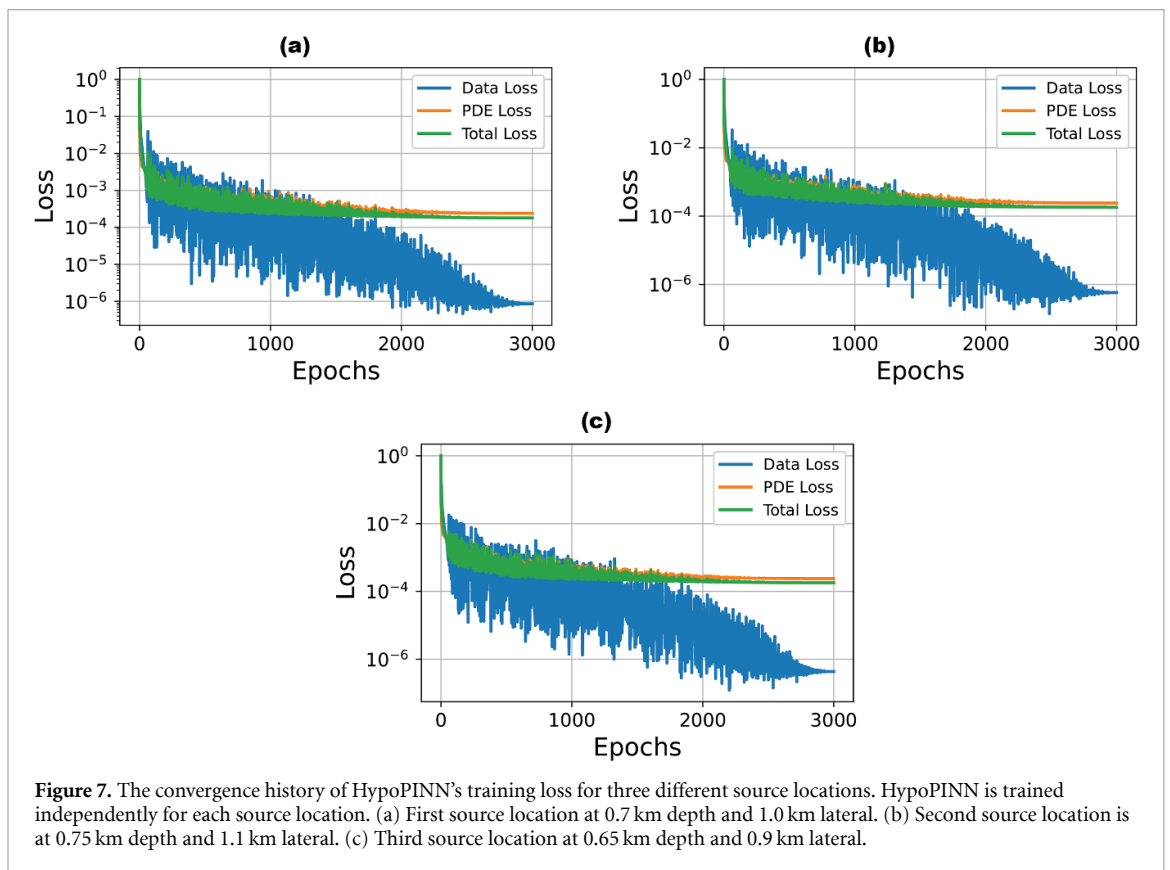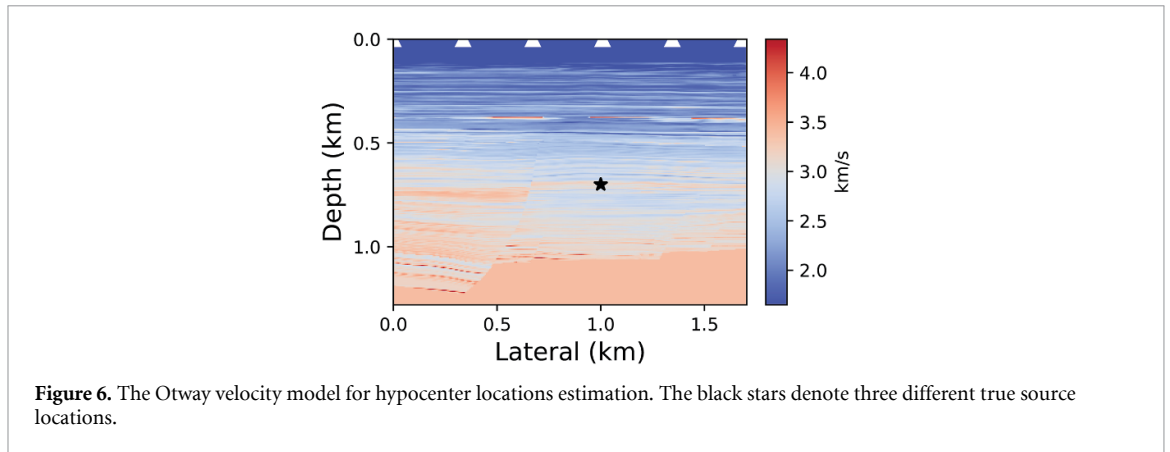
Figure 5. Predictive uncertainty of the hypocenter location associated with weights' realizations θ from the Laplace approximation for vertically varying velocity model. (a) The reconstructed eikonal solution. The black and green stars denote the true source and the estimated θ_{MAP} hypocenter locations, respectively. (b) The locations of hypocenter associated with 1000 θ realizations from the Laplace approximation denoted by green stars. (c) Convergence history of HypoPINN's loss.

Table 2. Statistics of predictive uncertainty of the hypocenter location associated with weights' realizations θ from the Laplace approximation for vertically varying velocity model.

Statistics	Depth (km)	Lateral (km)
Truth	**1.000**	**1.500**
Mean/MAP	1.015	1.500
Standard deviation	0.127	0.145

training the HypoPINN. Similar to the previous example, we minimize equation (5) with equation (2) as the log-likelihood term and the Gaussian prior (Tikhonov regularization). We perform the minimization for 3000 epochs and consider the last epoch's weights as θ_{MAP} for the eikonal solution prediction, thus, estimating the hypocenter locations. The network took 60.52 s and 45.82 μs on average for training and inference, respectively. The convergence of HypoPINN's training for all three scenarios is shown in figure 7. Also, in figure 8, overall predictions of eikonal solutions and the hypocenter locations are relatively accurate except for the slight deviation observed in figure 8(d) for the second scenario.

We proceed with the predictive uncertainties study by constructing the Laplace approximation to the posterior distribution as in equation (7) independently for each scenario in this example. Similar to the previous example, we consider the diagonal approximation of the covariance matrix for the Laplace approximation around the θ_{MAP}. We sample 1000 weights' realizations and perform the eikonal solution predictions by realizing HypoPINN with those weights for each scenario. The statistics of predictive uncertainty of the hypocenter location are tabulated in table 3. For the first and second scenarios in figures 9(b) and (c), we observe the uncertainty of the hypocenter is larger in the depth direction. Yet, in the second scenario, the uncertainty of the hypocenter is biased towards deeper depth due to the inaccurate prediction of its eikonal solution and hypocenter location. In the third scenario, as depicted in figure 9(d), we observe the predictive uncertainty for the hypocenter locations is well-constrained in depth and the lateral direction. As discussed earlier in the previous example, the predictive uncertainty of HypoPINN depends on the loss landscape surrounding the MAP solution location. This example demonstrates this phenomenon through three independent HypoPINN training for three different source locations. Nevertheless, with a limited number of observed data available and the usage of surface recordings, HypoPINN can recover relatively accurate prediction of the eikonal solution and its hypocenter locations with well-defined predictive uncertainty.

Figure 6. The Otway velocity model for hypocenter locations estimation. The black stars denote three different true source locations.

Figure 7. The convergence history of HypoPINN's training loss for three different source locations. HypoPINN is trained independently for each source location. (a) First source location at 0.7 km depth and 1.0 km lateral. (b) Second source location is at 0.75 km depth and 1.1 km lateral. (c) Third source location at 0.65 km depth and 0.9 km lateral.
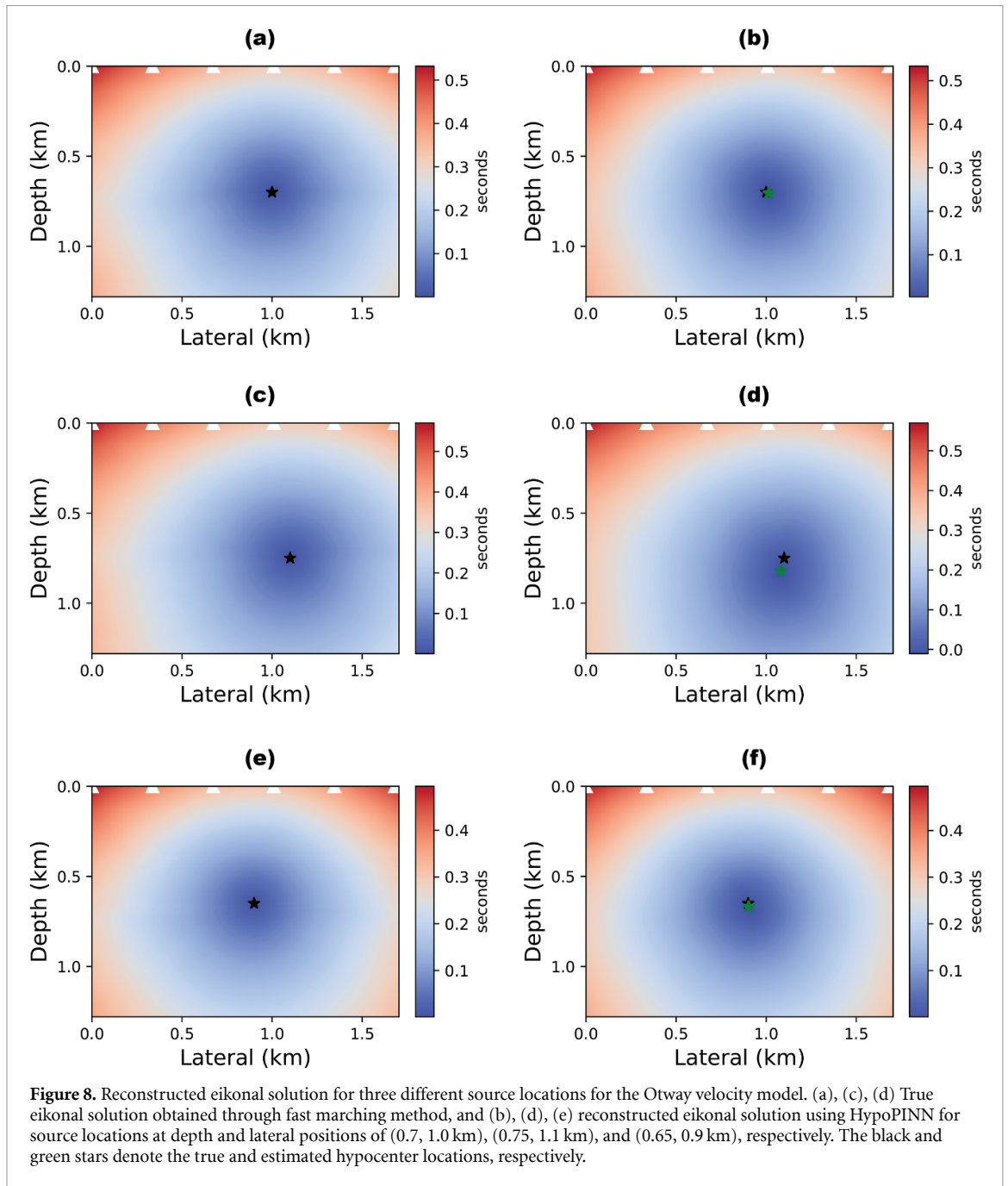
4. Discussion

This work focuses on developing an inversion framework for hypocenter localizations directly using PINNs and investigating its predictive uncertainties or, simply, forward modeling uncertainties in the context of HypoPINN. This predictive uncertainty is different from the physical model uncertainty, in which the quantity of interest is the physical quantity, e.g. hypocenter locations, velocity, etc. Our quantity of interest here is the PINN's network parameters θ. In the numerical examples, we demonstrated the efficacy of the proposed methodology in obtaining robust hypocenter location. However, localizing hypocenter locations alone is insufficient as many factors influence the predictive uncertainty of a neural network; thus, representing uncertainty and understanding its source is crucial for decision-making. This discussion will generally focus on the challenges and limitations of HypoPINN in localizing hypocenter locations and Laplace approximation in representing its predictive uncertainties.

4.1. Factored vs. unfactored eikonal equation for HypoPINN

Solving the eikonal equation using PINN was first introduced by [18] where the authors leveraged the factored eikonal equation [26] to avoid the singularity due to the point source. Formulating the problem as

Figure 8. Reconstructed eikonal solution for three different source locations for the Otway velocity model. (a), (c), (d) True eikonal solution obtained through fast marching method, and (b), (d), (e) reconstructed eikonal solution using HypoPINN for source locations at depth and lateral positions of (0.7, 1.0 km), (0.75, 1.1 km), and (0.65, 0.9 km), respectively. The black and green stars denote the true and estimated hypocenter locations, respectively.
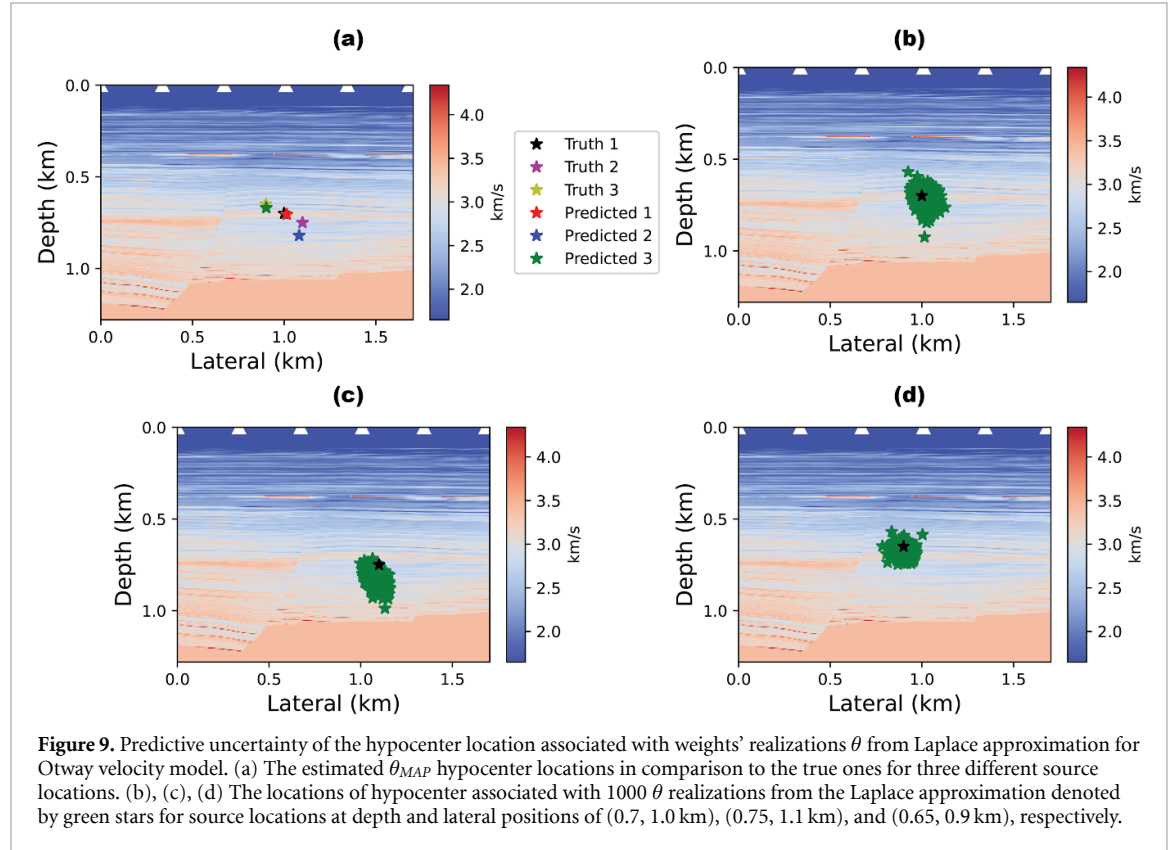
such constrains the null space and regulates the ill-posedness of a neural network as it only requires learning a scaling factor. Also, the factored eikonal formulation provides *a priori* information regarding the geometry of the traveltime as an assistant to the assigned neural network to learn and solve the eikonal equation accurately. However, this approach requires the source locations to be known, limiting its applicability in solving the hypocenter location problem.

HypoPINN is leveraging the unfactored or original eikonal equation in solving the hypocenter locations estimation. The main challenge for PINN in solving an eikonal equation and thus estimating hypocenter locations is the need to learn the eikonal equation as a whole by optimizing a large unconstrained solution space due to no prior information in assisting its learning. This makes the neural network learning process ill-posed as it is limited by large null-space solutions. This phenomenon is observed in our first numerical example where HypoPINN failed to predict the eikonal solution for the initial weights sampled from Xavier uniform distribution despite its loss history showing convergence. Nevertheless, these challenges can be alleviated by several strategies such as designing a task-specific network architecture, different initial weights' prior distribution, optimal hyperparameters tuning, etc. However, such strategies will influence the neural network's loss landscape, thus affecting its predictive uncertainties [34, 35, 38].

Table 3. Statistics of predictive uncertainty of the hypocenter location associated with weights' realizations θ from the Laplace approximation for Otway velocity model. The coordinates are presented in (depth, lateral) format.

Source location	Truth (km)	Mean/MAP (km)	Std (km)
1	(0.700, 1.000)	(0.705, 1.015)	(0.033, 0.032)
2	(0.750, 1.100)	(0.820, 1.082)	(0.033, 0.028)
3	(0.650, 0.900)	(0.669, 0.902)	(0.024, 0.028)

Figure 9. Predictive uncertainty of the hypocenter location associated with weights' realizations θ from Laplace approximation for Otway velocity model. (a) The estimated θ_{MAP} hypocenter locations in comparison to the true ones for three different source locations. (b), (c), (d) The locations of hypocenter associated with 1000 θ realizations from the Laplace approximation denoted by green stars for source locations at depth and lateral positions of (0.7, 1.0 km), (0.75, 1.1 km), and (0.65, 0.9 km), respectively.

4.2. Predictive uncertainties estimation with Laplace approximation

Laplace approximation is arguably the simplest family of approximations for the intractable posteriors of deep neural networks. It forms a Gaussian approximation to the exact posterior. Its mean equals the MAP estimate of a neural network, and its covariance equals the negative inverse Hessian (i.e. approximation thereof) of the loss functions evaluated at the MAP estimate. Due to its simplicity, the Laplace approximation can be applied to any pre-trained neural network in a cost-efficient post hoc manner. The main ingredient of the Laplace approximation is its covariance estimation, and there are various approaches to approximate the covariance [21, 22]. A good covariance approximation is important in predicting the uncertainty embedded in the neural network training as it affects the quality of uncertainty representation. However, good covariance approximation is commonly computationally limited by the size of learnable network parameters. This work considers the diagonal approximation, which is the simplest approximation approach in studying the HypoPINN's predictive uncertainties. Thus, such representation of uncertainties might be either under- or over-estimated. Also, Laplace approximation relies on crude approximations of the posterior distribution as the posterior is intractable due to the neural networks' size and nonlinearity.

4.3. HypoPINN for real-time microseismic monitoring

To have an ability to perform real-time analysis giving an upper hand in prompt decision making in real-time microseismic monitoring. However, obtaining such an ability is far from trivial. To realize the goals of real-time microseismic monitoring, the methodology and its algorithm should be computationally efficient and robust in handling large volumes of data with a poor signal-to-noise ratio (SNR). To make matter worse, a poor-calibrated real-time velocity model induces errors that are carried forward to the hypocenter locations detection; thus, limiting the possibility of real-time microseismic monitoring.

HypoPINN is the first step in achieving real-time microseismic monitoring. However, the main limitation is it assumes the velocity model is accurate. This assumption is critical as an accurate velocity

model is practically hard to achieve. Nevertheless, as a proof of concept, HypoPINN has shown computational efficiency in locating the hypocenter locations accurately throughout the numerical examples. There are two key unified advancements to alleviate this limitation in realizing real-time microseismic monitoring. First, to unify the hypocenter localization and velocity model reconstruction processes simultaneously, a bi-networks architecture similar to [25] could be introduced. Although this approach looks promising, the main challenge of this bi-networks approach is that the learning process becomes strongly ill-posed as it is limited by large null-space solutions that require both networks to compensate each other to settle in a *sweet-spot* in the solution space—similar to training a generative adversarial network (GAN) [39]. Secondly, the Bayesian framework [40] plays a major role in practically advancing HypoPINN for real-time microseismic monitoring through the concept of *knowledge-adaptation priors* [41], where the posterior distribution of HypoPINN at the previous time-frame can be re-implemented as a prior distribution for the later time-frame training. This concept is similar in spirit to data assimilation in weather prediction [42]. Note that, *knowledge-adaptation priors* is different from transfer learning through the usage of distributions. However, this approach is limited only to the exponential family of distributions due to its parametric nature. The advancements discussed here will be our main focus in the future in achieving the aforementioned objectives.

5. Conclusion

We developed HypoPINN—a PINN-based algorithm for hypocenter localization. We trained a neural network by minimizing a loss function formed by the misfit of the observed and predicted traveltimes and the eikonal residual term. Given picked arrival times for an event, this provides predicted traveltimes in the entire computational domain. The hypocenter location is then given by the location of the minimum travel time. We also introduced an approximate Bayesian framework for estimating predictive uncertainty or, simply, forward modeling uncertainty in the context of HypoPINN. We investigated the uncertainties propagation from the random realizations of HypoPINN's weights and biases using the Laplace approximation to the predicted solutions. This generally opens up new pathways in investigating the training dynamics of PINN, especially in the PINN's weights initialization to obtain a correct and robust solution. Through numerical examples, we demonstrated the efficacy of the proposed methodology in obtaining robust hypocenter locations equipped with predictive uncertainty estimation, even in the case where sparse traveltime observations are available. Through the Bayesian framework, a trained neural network and its covariance could be used as a prior distribution in obtaining new hypocenter locations in almost real-time. This allows for computationally efficient and robust hypocenter localization that may enable the goal of real-time microseismic monitoring.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/izzatum/laplace-hypopinn.

Acknowledgment

ORCID iD

Muhammad Izzatullah ⬤ https://orcid.org/0000-0001-6349-3103

References

[1] Mignan A, Broccardo M, Wiemer S and Giardini D 2017 Induced seismicity closed-form traffic light system for actuarial decision-making during deep fluid injections *Sci. Rep.* **7** 1–10
[2] Foulger G R, Wilson M P, Gluyas J G, Julian B R and Davies R J 2018 Global review of human-induced earthquakes *Earth-Sci. Rev.* **178** 438–514
[3] Lienert B R, Berg E and Frazer L N 1986 Hypocenter: an earthquake location method using centered, scaled and adaptively damped least squares *Bull. Seismol. Soc. Am.* **76** 771–83
[4] Pugh D J, White R S and Christie P A F 2016 A Bayesian method for microseismic source inversion *Geophys. J. Int.* **206** 1009–38
[5] Bancroft J C, Wong J and Han L 2010 *SEG Technical Program Expanded Abstracts 2010* pp 2191–5
[6] Waldhauser F and Ellsworth W L 2000 A double-difference earthquake location algorithm: method and application to the Northern Hayward fault, California *Bull. Seismol. Soc. Am.* **90** 1353–68

[7] Zhang P, Han L, Gao H and Sun H 2015 *SEG Technical Program Expanded Abstracts 2015* pp 2650–4

[8] Wang H and Alkhalifah T 2018 Microseismic imaging using a source function independent full waveform inversion method *Geophys. J. Int.* **214** 46–57

[9] Song C, Wu Z and Alkhalifah T 2019 Passive seismic event estimation using multiscattering waveform inversion *Geophysics* **84** KS59–KS69

[10] Perol T, Gharbi Mel and Denolle M 2018 Convolutional neural network for earthquake detection and location *Sci. Adv.* **4** e1700578

[11] Wang H, Alkhalifah T, Waheed U B and Birnie C 2021 Data-driven microseismic event localization: an application to the Oklahoma Arkoma basin hydraulic fracturing data *IEEE Trans. Geosci. Remote Sens.* **60** 4506212

[12] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707

[13] Grubas S, Yaskevich S and Duchkov A 2021 Localization of microseismic events using the physics-informed neural-network for traveltime computation *82nd EAGE Annual Conf. & Exhibition* vol 2021 (European Association of Geoscientists & Engineers) pp 1–5

[14] Smith J D, Ross Z E, Azizzadenesheli K and Muir J B 2022 HypoSVI: hypocentre inversion with stein variational inference and physics informed neural networks *Geophys. J. Int.* **228** 698–710

[15] Song C, Alkhalifah T and Waheed U B 2021 A versatile framework to solve the Helmholtz equation using physics-informed neural networks *Geophys. J. Int.* **228** 1750–62

[16] Alkhalifah T, Song C, Waheed U B and Hao Q 2021 Wavefield solutions from machine learned functions constrained by the Helmholtz equation *Artif. Intell. Geosci.* **2** 11–19

[17] Song C and Alkhalifah T 2022 Wavefield reconstruction inversion via physics-informed neural networks *IEEE Trans. Geosci. Remote Sens.* **60** 1–12

[18] Waheed U B, Haghighat E, Alkhalifah T, Song C and Hao Q 2021 PINNeik: eikonal solution using physics-informed neural networks *Comput. Geosci.* **155** 104833

[19] Waheed U bin, Alkhalifah T, Haghighat E, Song C and Virieux J 2021 PINNtomo: seismic tomography using physics-informed neural networks (arXiv:2001.04536)

[20] Yang L, Meng X and Karniadakis G E 2021 B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data *J. Comput. Phys.* **425** 109913

[21] Ritter H, Botev A and Barber D 2018 A scalable Laplace approximation for neural networks *6th Int. Conf. on Learning Representations, ICLR 2018-Conf. Track Proc.* vol 6 (Int. Conf. on Representation Learning)

[22] Daxberger E, Kristiadi A, Immer A, Eschenhagen R, Bauer M and Hennig P 2021 Laplace redux-effortless Bayesian deep learning *Advances in Neural Information Processing Systems* vol 34

[23] Bronstein M M, Bruna J, Cohen T and Veličković P 2021 Geometric deep learning: grids, groups, graphs, geodesics, and gauges (arXiv:2104.13478v2)

[24] Waheed U B, Alkhalifah T, Haghighat E and Song C 2021 A holistic approach to computing first-arrival traveltimes using neural networks (arXiv:2101.11840)

[25] Waheed U B, Alkhalifah T, Haghighat E, Song C and Virieux J 2021 PINNtomo: seismic tomography using physics-informed neural networks (arXiv:2001.04536)

[26] Fomel S, Luo S and Zhao H 2009 Fast sweeping method for the factored Eikonal equation *J. Comput. Phys.* **228** 6440–55

[27] Murphy K P 2023 *Probabilistic Machine Learning: Advanced Topics* (Cambridge, MA: MIT Press)

[28] Kirkpatrick J *et al* 2017 Overcoming catastrophic forgetting in neural networks *Proc. Natl Acad. Sci.* **114** 3521–6

[29] Misra D 2019 Mish: a self regularized non-monotonic activation function (arXiv:1908.08681)

[30] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)

[31] Geyer R A 1959 Lessons in Seismic Computing (Houston, TX: Society of Exploration Geophysicists) (https://doi.org/10.1190/1.9781560802563)

[32] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics* (JMLR Workshop and Conf. Proc.) pp 249–56

[33] He K, Zhang X, Ren S and Sun J 2015 Delving deep into rectifiers: Surpassing human-level performance on imagenet classification *Proc. IEEE Int. Conf. on Computer Vision* pp 1026–34

[34] Li H, Xu Z, Taylor G, Studer C and Goldstein T 2017 Visualizing the loss landscape of neural nets. (arXiv:1712.09913)

[35] Garipov T, Izmailov P, Podoprikhin D, Vetrov D P and Wilson A G 2018 Loss surfaces, mode connectivity and fast ensembling of DNNs *Advances in Neural Information Processing Systems* vol 31

[36] Glubokovskikh S, Pevzner R, Dance T, Caspari E, Popik D, Shulakova V and Gurevich B 2016 Seismic monitoring of CO_2 geosequestration: CO2CRC Otway case study using full 4d FDTD approach *Int. J. Greenhouse Gas Control* **49** 201–16

[37] Sethian J A 1996 A fast marching level set method for monotonically advancing fronts *Proc. Natl Acad. Sci.* **93** 1591–5

[38] Wilson A G and Izmailov P 2020 Bayesian deep learning and a probabilistic perspective of generalization *Advances in Neural Information Processing Systems* vol 33 pp 4697–708

[39] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks (arXiv:1406.2661)

[40] Osawa K, Swaroop S, Khan M E E, Jain A, Eschenhagen R, Turner R E and Yokota R 2019 Practical deep learning with Bayesian principles *Advances in Neural Information Processing Systems* vol 32

[41] Khan M E E and Swaroop S 2021 Knowledge-adaptation priors *Advances in Neural Information Processing Systems* vol 34 pp 19757–70

[42] Law K, Stuart A and Zygalakis K 2015 *Data Assimilation* vol 214 (Cham: Springer)