


UDC: 004.9

PHYSICS-INFORMED NEURAL NETWORKS FOR INVERSE TASKS OF ONE-DIMENSIONAL WAVE PROPAGATION

Igor Kolych , Roman Shuvar 
Ivan Franko National University of Lviv,
50 Drahomanova St., UA-79005 Lviv, Ukraine

Kolych, I., Shuvar, R. (2025). Physics-Informed Neural Networks for Inverse Tasks of One-dimensional Wave Propagation. *Electronics and Information Technologies*, 31, 105–114.
<https://doi.org/10.30970/eli.31.9>

ABSTRACT

Background. Physics-informed neural networks (PINNs) are a family of learning methods that guide neural networks with the laws of physics, rather than relying only on data. PINNs demonstrated strong capabilities in solving forward and inverse problems for partial differential equations. In this study, the application of PINNs to single-pulse wave propagation in non-uniform media is explored, focusing on reconstructing velocity profiles from wavefield data. Specifically, we focus on reconstructing velocity profiles from wavefield data using PINNs and their convolutional extension, Physics-Informed Convolutional Neural Networks (PICNNs). The work is motivated by applications in seismology, acoustics, and biomedical imaging, where accurate velocity estimation is crucial.

Materials and Methods. We generate synthetic wave data using the finite element method (FEM) and use a Gaussian impulse so that the result of the neural models can be directly compared against the numerical benchmark. PINNs and PICNNs are applied to solve forward tasks (wavefield prediction) and inverse tasks (velocity reconstruction). Also, we use training data that contains varying levels of Gaussian noise.

Results and Discussion. For the forward task, both PINNs and PICNNs closely match the numerical simulations, with PICNNs reaching high accuracy faster. For inverse tasks, PICNNs demonstrated superior performance in reconstructing spatially varying velocity profiles, while PINNs struggled with convergence due to local maxima in the optimization landscape. The inclusion of a smoothness constraint in the loss function eliminated artifacts in the reconstructed velocities without increasing computational cost. The approach remains effective on testing cases and is robust to moderate noise levels in the input data.

Conclusion. PICNNs efficiently solve forward and inverse single-pulse propagation in non-uniform media, matching FEM in forward accuracy and outperforming standard PINNs in inverse recovery. These results indicate strong potential for practical sensing and imaging. Future work will explore the extension of these methods to multi-pulse scenarios.

Keywords: Physics-informed neural networks, PICNN, single-pulse, inverse problem, velocity reconstruction, non-uniform media

INTRODUCTION

Physics-informed neural networks (PINNs) are a class of neural networks that integrate the laws of physics, described by partial differential equations (PDEs), directly into the learning process. This revolutionary tool embeds the physics of the system directly into its loss functions [1]. Unlike traditional neural networks (NNs), which rely primarily on data, PINNs leverage physical laws, such as wave equations, to improve accuracy and enable the solution of inverse problems. This unique ability makes PINNs particularly



© 2025 Igor Kolych & Roman Shuvar. Published by the Ivan Franko National University of Lviv on behalf of Електроніка та інформаційні технології / Electronics and Information Technologies. This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

appealing for applications in wave propagation, where both the forward and inverse problems must often be solved simultaneously [2], [3]. These capabilities have made them especially valuable in complex problems across diverse fields such as seismology [4], fluid mechanics [5], and biomedical imaging, where accurate modeling of wave propagation and reconstruction of material properties is essential [6], [7].

Recent advancements in PINNs have extended their applicability to high-dimensional problems and multi-physics scenarios. In particular, hybrid approaches like Physics-Informed Convolutional Neural Networks (PICNNs) have gained attention for addressing some of the limitations of traditional PINNs. PICNNs combine the physics-driven constraints of PINNs with the computational efficiency and feature extraction of Convolutional Neural Networks (CNNs) [8], [9]. These advancements are critical in addressing challenges such as high-frequency wave propagation, sparse data scenarios, and complex inverse tasks for real noisy data.

To address these challenges, this study explores single-pulse wave propagation in a 1D non-uniform medium and evaluates the ability of both PINNs and their convolutional extension, PICNNs, to solve forward and inverse tasks, including scenarios affected by noise. Enhancements in the loss function, such as smoothness constraints for inverse tasks, are introduced to improve accuracy and mitigate artifacts.

This investigation serves as a foundation for addressing more complex wave propagation problems, such as those involving multiple overlapping pulses or higher-dimensional domains. Also, this study contributes to the growing body of research on physics-informed neural networks, offering insights and methodologies for their application in addressing challenging wave propagation problems.

MATERIALS AND METHODS

Governing Equations

The propagation of a single wave pulse in a non-uniform 1D medium is governed by the wave equation:

$$\frac{\partial^2 u(x, t)}{\partial t^2} = c(x)^2 \frac{\partial^2 u(x, t)}{\partial x^2}, \quad (1)$$

where $u(x, t)$ describes the displacement of the wave at position x and time t , and $c(x)$ is the spatially varying wave velocity.

Initial conditions for the problem are defined as:

$$u(x, 0) = g(x), \quad \frac{\partial u(x, 0)}{\partial t} = 0, \quad (2)$$

with $g(x)$ the initial wave profile, that is, a Gaussian pulse.

The boundary conditions (BCs) applied are absorbing boundary conditions (ABCs), designed to ensure that reflected waves do not appear. Specifically, Mur's ABCs [10] are used to simulate the wavefield at the edge of the computational domain. For a single simulation step, Mur's BC is expressed as

$$u(0, t + \Delta t) = u(\Delta x, t) + \frac{c(0)\Delta t - \Delta x}{c(0)\Delta t + \Delta x} (u(\Delta x, t + \Delta t) - u(0, t)), \quad (3)$$

where Δt is the time step and Δx is the position step. BC for the different side can be found by replacing zero-location with the length of the domain and flipping the sign of Δx .

Numerical Simulations

The data for training NNs was generated using the finite element method (FEM) [11]. Such an approach is widely used in the research community [12]. The single pulse propagation is conducted for simulation to determine the capabilities of different NNs architectures.

In this scenario, the initial conditions consisted of a Gaussian pulse starting at various x -locations. This gives ability to perform a comparison with results received by classical methods. The training set consists of 10 cases of pulse propagation that differ by the initial location of the pulse. An example of pulse propagation in the space-time domain is shown in **Fig. 1**. Note that location is measured in points and time is measured in steps for convenience.

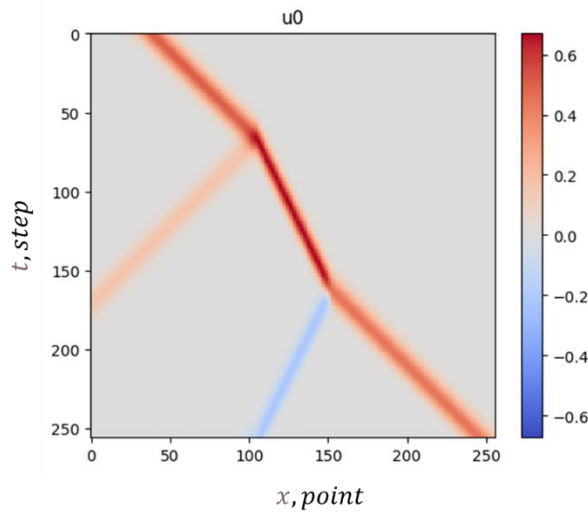


Fig. 1. Example of pulse propagation in the space-time domain.

A Gaussian pulse in **Fig. 1** is initialized at a random location x_0 and described by the equation:

$$u(x, 0) = A \exp\left(-\frac{x - x_0}{w^2}\right), \quad (4)$$

with amplitude $A = 1$, the initial location x_0 is limited to 0 and 255, and the width $w = 10$ points. These parameters are selected for simulation convenience. Note that zero time is shifted by 64 steps for training data to avoid an explicit relation between the initial point and the input data during the training procedure. This pulse propagates with a velocity $c(x)$ that was spatially varying (non-uniform medium) as shown in **Fig. 2**.

Forward tasks were modeled using different PINN architectures, while inverse tasks were solved to reconstruct $c^2(x)$ from wavefield data.

Also, the data was generated with different levels of Gaussian noise $N(0, \sigma)$ that can be described by the equation:

$$u_N(x, t) = u(x, t) + N(0, \sigma), \quad (5)$$

where 0 means the noise has zero mean and σ is its variance.

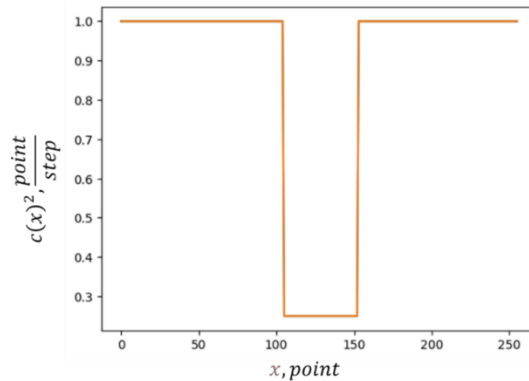


Fig. 2. Squared single pulse velocity measured in simulation mesh units.

Physics-Informed Neural Network Architectures

Different PINN types are considered to solve forward and inverse tasks. Classical PINN was fully connected NN (FNN) was used with 6 hidden layers as demonstrated in **Fig. 3**, and in more detail:

- The input layer has initial conditions x_0 and space-time domain investigation coordinates x, t .
- 6 hidden layers that are fully connected layers with 32, 64, 128, 64, and 32 neurons, using the Tanh activation function.
- The output layer has one neuron for representing $u(x, t)$

FNN in the bottom part of **Fig. 3** approximates velocity for different x-locations. This NN is a fully connected neural network that has 3 hidden layers:

- The input layer has an x-coordinate.
- 3 hidden layers that are fully connected layers with 16, 32, and 16 neurons, using the Tanh activation function.
- The output layer has one neuron for representing $c(x)$.

The PINN type for data with spatial relations is PICNN [5]. It uses CNN to store and process information. The current PICNN architecture is the following:

- The input layer has initial conditions x_0 .
- 8 hidden layers that are convolutional transpose two-dimensional layers, kernel sizes equal to 2, and the “stride” parameter is 2. The activation function in each layer is Tanh. The number of channels for each layer is in **Fig. 3**.
- The output layer produces a 256x256 matrix of $u(x, t)$ values.

CNN, which is in the bottom image of **Fig. 4**, is used for the velocity estimation task. Its architecture is similar to PICNN:

- The input layer uses a constant as velocity is not a function of any parameters.
- 8 hidden layers that are convolutional transpose one-dimensional layers, kernel sizes equal to 2, and the stride parameter is 2. The activation function in each layer is Tanh. The number of channels is smaller compared to PICNN and shown on **Fig. 4**.

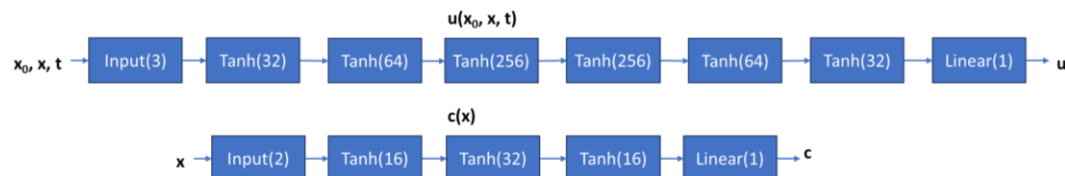


Fig. 3. Architecture of a fully connected neural network for describing pulse propagation and velocity.

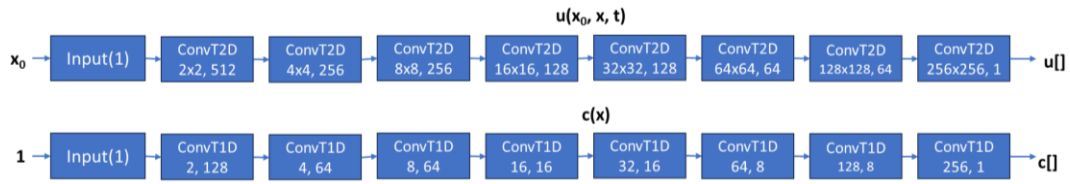


Fig. 4. Architecture of a convolutional neural network for describing pulse propagation and velocity.

- The output layer produces a vector of 256 elements that contain $c(x)$.

In the case of a forward task, the PINN and PICNN loss function used in training combines data loss and physics loss:

$$L = L_{data} + \lambda L_{physics}, \quad (6)$$

where L_{data} is the data error, which reflects the discrepancy between the model's predictions and the simulated data, and $L_{physics}$ is the physics error, which quantifies how well the model adheres to the governing PDEs.

The PINN and PICNN models were developed using PyTorch version 2.5.0 [9].

The training data consists of multiple simulated pulse propagations. This data was generated from the wave equations (1-4) using FEM.

The networks for the forward task are trained using the Adam optimizer with an initial learning rate of 0.001. After 5000 iterations, the learning rate was decreased by two times. The total number of training iterations is 20 000. MSE of the physics loss is scaled by 0.1 relative to MSE of the data loss to provide correct training.

For inverse tasks, the velocity field $c(x)$ is parameterized by FNN or CNN depending on used PINN type. The objective function for optimization velocity NN is physics loss only.

RESULTS AND DISCUSSION

Single Pulse Propagation and Inverse Task

The forward task for single-pulse wave propagation was solved successfully using both PINN and PICNN. Both architectures produced visually indistinguishable results with negligible mean square errors (MSE) relative to the FEM data (Fig. 1). This means these architectures are capable of approximate the simulated data. The convergence is expected for both PINNs, but PICNN demonstrates faster convergence speed as expected. This improvement can be attributed to the convolutional layers in PICNNs, which are inherently better at capturing spatial relationships and local features in the input data on a rectangular mesh.

In the inverse task, PINN and PICNN were evaluated for reconstructing the spatially varying velocity $c(x)$ (Fig. 2) from the observed wavefield data (as in Fig. 1). The results revealed a significant disparity in performance between the two architectures. PINN exhibited slow convergence and reconstruction the velocity field $c(x)$ was not reached, as shown in Fig. 5. The main problem is stopping convergence because of local maxima. Physical losses of different pulse configurations decrease locally (Fig. 5) which reduces total error and correspondingly the size of the gradient. Also, this may be caused by sharp changes of velocity, and the result $c(x)$ is quite smooth, so, decrease in error in one part $c(x)$ increases the error in the other part. As a result, out-of-the-box PINNs with some tunings are not able to solve this inverse task, but potentially this can be done after much better tuning and architecture search.

PICNNs, on the other hand, demonstrated good performance in this task. The inverse task convergence time is only four times larger than the forward task time. Also, it was able to capture local spatial patterns for the velocity estimation task. Despite their superior

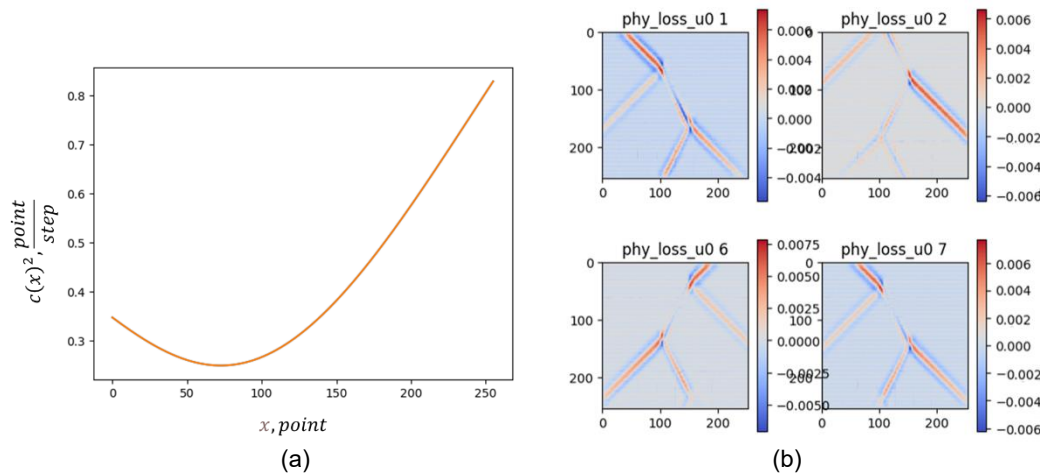


Fig. 5. Example of squared velocity (a) and physics loss (b) for a single pulse inverse task.

performance, initial PICNN reconstructions exhibited minor high-frequency ripples in the reconstructed velocity field (left graph in Fig. 6). There are multiple techniques in DSP to remove these ripples, but it may break physics that is embedded in PGE. To avoid this, the loss function was updated by adding one more term that adds a mathematical constraint for $c(x)$. A smoothness constraint was introduced into the loss function via an L1-norm of the velocity gradient

$$L_{\text{math}} = \frac{dc(x)}{dx}. \quad (7)$$

Also, it is possible to use the L2-norm of the derivative, but the L1-norm provides sufficient accuracy. As a result, the loss function used in training changes to:

$$L = L_{\text{data}} + \lambda L_{\text{physics}} + \gamma L_{\text{math}}, \quad (8)$$

where $\gamma = 10^{-5}$ is much smaller than λ and should cause its effect at the end of the training process.

This change of loss function gives a significant improvement (right graph in Fig. 6).

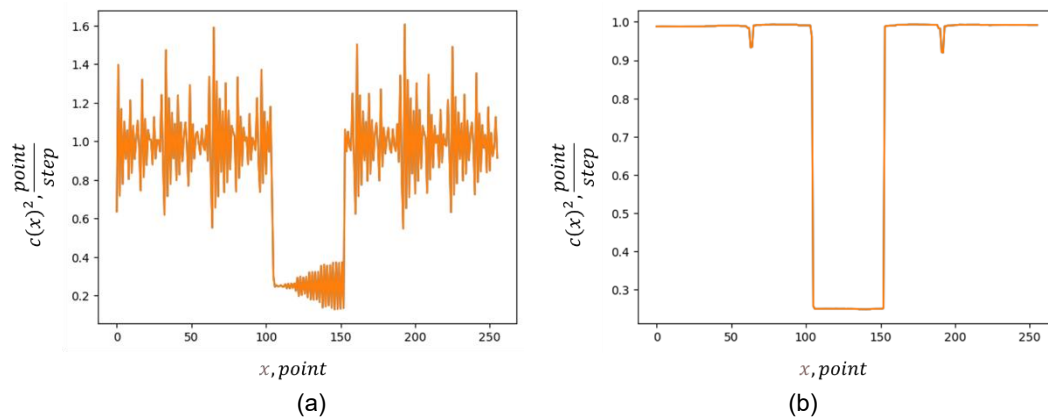


Fig. 6. Found velocity for a single pulse inverse task before (a) and after (b) the loss function change.

Importantly, the addition of this constraint did not increase training time, making it a cost-effective enhancement to the methodology.

Robustness to Noise in Wavefield Data for Inverse Task

To evaluate the robustness of PICNN for the inverse task, Gaussian noise is added to the wavefield data in varying noise-to-signal ratios: $\sigma/A = 0, 0.01, 0.02$, and 0.05 . **Fig. 7** illustrates the training loss curves, highlighting the significant challenges posed by higher noise levels. For $\sigma/A = 0.05$, the training loss increased by several orders of magnitude, making it difficult for the network to minimize the physics loss term effectively.

Also, it was found that the model cannot solve the inverse task if even a small noise is added. The root cause is in the ratio between data and physical losses. Noise increases data loss, and this affects physical loss. More precisely, it prevents decreasing physical loss because NN cannot fully approximate noisy data that significantly changes from one location to another. This noise behavior is typical for multiple applications, e.g., signals in wireless devices and ultrasound scanning. To mitigate this issue, it was proposed to use an increasing physics loss multiplier with each epoch, as described by the following equation:

$$\lambda = \frac{epoch}{S}, \quad (9)$$

where S is the scaler of physics loss multiplier, e.g., $S = 1000$ is sufficient to recover velocity in case of $\sigma/A = 0.01$. The scaler S can be empirically selected based on two principles: the physics loss should be smaller than data loss at the beginning of training (e.g., first 10 epochs); the physics loss should be much larger than data loss caused by noise at the end of training.

Using a changeable ratio between data and physical loss (9), it is possible to successfully reconstruct $c^2(x)$ across all noise levels. However, the behavior λ should be following. At the beginning of training procedure (epoch value is small), data loss should be dominant to approximate data as good as possible. The flattening of the loss curve in the right graph of **Fig. 7** shows this limit. This means NN has some gradient to update weights, but it cannot be realized because of the noise properties. Then physical loss should be larger to remove noise from data and provide $c^2(x)$ reconstruction. As a result, larger noise requires a smaller scaler S to increase λ . In case of $\sigma/A = 0.05$, S it should be decreased to 20.

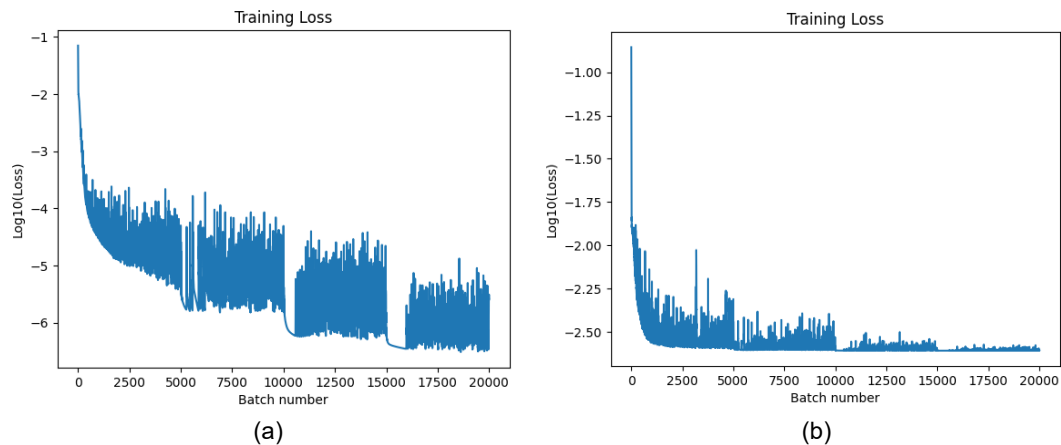


Fig. 7. Training loss for case w/o noise (a) and w/ noise (b) for $\sigma/A = 0.05$.

PICNN capabilities for the inverse task of velocity reconstruction are shown in Fig. 8. The graph in Fig. 8 shows the mean absolute error (MSE) of reconstruction $c^2(x)$ at different σ/A values and different λ . It can be seen that the scaler λ should be increased if noise is increased. This can be done in two ways. The first is to decrease the scaler S as is written above. The second is to increase the number of epochs that gets obvious from equation (9).

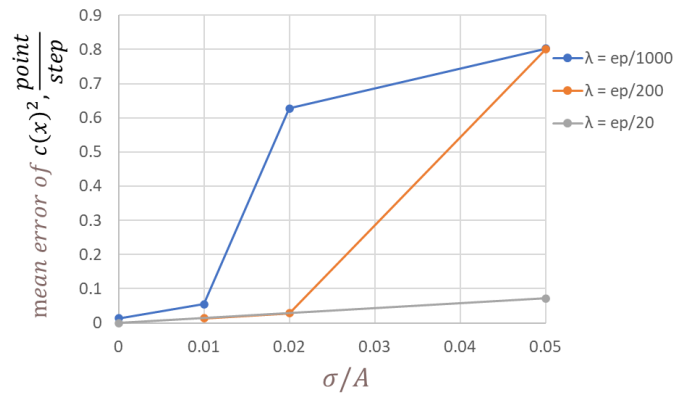


Fig. 8. Mean error for the reconstructed $c^2(x)$ for different noise-to-signal ratios and physics loss multipliers λ .

Also, there is a reconstruction error increase if the noise is larger. However, this is not caused directly by the noise. This is caused by the smaller contribution L_{math} in equation (8). The multiplier γ of the smoothness constraint L_{math} plays an important role in mitigating the noise effect too, and it should be increased λ to reduce artifacts in the reconstructed $c^2(x)$ profile.

CONCLUSION

This study demonstrates the effectiveness of PINNs and PICNNs for solving single-pulse wave propagation problems in non-uniform media, because we have good results for simplified tasks. PICNNs outperform PINNs in reconstructing velocity profiles, particularly under noisy conditions. Adding smoothness term to the loss function and using a variable multiplier of the physics loss significantly improves reconstruction quality without increasing computational cost. The ability of PICNNs to handle noisy data makes them suitable for real-world applications.

Future work will extend these methods to handle multi-pulse scenarios and higher-dimensional problems.

REFERENCES

- [1] Raissi M., Perdikaris P., & Karniadakis G. E. (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. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [2] Chen Y., Lu L., Karniadakis G. E., Negro L. D. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* 28, 11618-11633 <https://doi.org/10.1364/OE.384875>
- [3] Piao S., Gu H., Wang A. and Qin P. (2024) A Domain-Adaptive Physics-Informed Neural Network for Inverse Problems of Maxwell's Equations in Heterogeneous

- Media. IEEE Antennas and Wireless Propagation Letters, vol. 23, no. 10, pp. 2905–2909, <https://doi.org/10.1109/LAWP.2024.3413851>
- [4] Moseley B., Markham A., Nissen-Meyer T. (2020) Solving the wave equation with physics-informed deep learning. arXiv preprint <https://arxiv.org/pdf/2006.11894>
- [5] Arzani A., Wang J., D'Souza R. (2021) Uncovering near-wall blood flow from sparse data with physics-informed neural networks. arXiv preprint <https://arxiv.org/pdf/2104.08249>
- [6] Qi S. and Sarris C. D. (2024) Hybrid Physics-Informed Neural Network for the Wave Equation With Unconditionally Stable Time-Stepping. IEEE Antennas and Wireless Propagation Letters, 23(4), 1360–1365. <https://doi.org/10.1109/LAWP.2024.3355896>
- [7] Kolych, I. (2025). Physics-Informed Neural Networks for Narrowband Signal Propagation Modeling. Electronics and Information Technologies, 30, 113–120. <https://doi.org/10.30970/eli.30.9>
- [8] Gao H., Sun L., Wang J. (2020) Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. arXiv preprint <https://arxiv.org/pdf/2011.02364>
- [9] Tong J., Li H., Xu B. and Shi Y. (2025) Physics-Informed Convolutional Transposed Neural Network for 2-D Reconstruction of Hypersonic Plasma Wakes. IEEE Sensors Journal, 25(6) 10079–10086 <https://doi.org/10.1109/JSEN.2025.3538625>
- [10] Zheng G., Kishk A., Glisson A., and Yakovlev A. (2006) Implementation of Mur's absorbing boundaries with periodic structures to speed up the design process using finite-difference time-domain method. Progress In Electromagnetics Research 58, 101–114. <http://dx.doi.org/10.2528/PIER05062103>
- [11] Kochmann D. (2025) Introduction to Finite Element Analysis. Lecture Notes. ETH Zurich https://ethz.ch/content/dam/ethz/special-interest/mavt/mechanical-systems/mm-dam/documents/Notes/IntroToFEA_red.pdf
- [12] Pakravan A. (2024) One-Dimensional Elastic and Viscoelastic Full-Waveform Inversion in Heterogeneous Media Using Physics-Informed Neural Networks. IEEE Access, 12, 69922–69940, <https://doi.org/10.1109/ACCESS.2024.3402240>
- [13] PyTorch v2.5.0. Previous PyTorch Versions. URL: <https://pytorch.org/get-started/previous-versions/#v250>

ФІЗИЧНО-ІНФОРМОВАНІ НЕЙРОННІ МЕРЕЖІ ДЛЯ ОБЕРНЕНОЇ ЗАДАЧІ ПОШИРЕННЯ ОДНОВИМІРНИХ ХВИЛЬ

Ігор Колич^{ORCID}, Роман Шувар^{ORCID}

Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, м. Львів, 79005, Україна

АНОТАЦІЯ

Вступ. Фізично-інформовані нейронні мережі (ФІНМ) — це сімейство методів навчання, які спрямовують нейронні мережі законами фізики, а не покладаються лише на дані. ФІНМ продемонстрували високі можливості у розв'язанні прямих та обернених задач для диференціальних рівнянь з частинними похідними. У цій роботі досліджується застосування ФІНМ для моделювання поширення одиночного імпульсу у неоднорідному середовищі з акцентом на реконструкцію профілів швидкості з даних хвильового поля. Зокрема, увага приділяється реконструкції профілів швидкості з використанням ФІНМ і їхньої розширеної версії на основі згорткових нейронних мереж (ФІЗНМ). Робота мотивована застосуваннями у сейсмології, акустиці та біомедичній візуалізації, де точне визначення швидкості є вкрай важливим.

Матеріали та методи. Ми генеруємо синтетичні хвильові дані методом скінченних елементів (МСЕ) і використовуємо гаусовий імпульс, щоб результати нейронних моделей можна було безпосередньо порівняти з чисельним еталоном. Для вирішення прямих задач (прогнозування хвильового поля) та обернених задач (реконструкція швидкості) застосовуються ФІНМ і ФІЗНМ. Також ми використовуємо навчальний набір даних, що має різні рівні гаусівського шуму.

Результати. У прямій задачі і ФІНМ, і ФІЗНМ добре узгоджуються з числовими симуляціями, причому ФІЗНМ швидше досягають високої точності. У випадку обернених задач ФІЗНМ показали кращі результати у реконструкції просторово змінних профілів швидкості, тоді як ФІНМ стикалися з проблемами збіжності через локальні максимуми в оптимізаційному ландшафті. Включення обмеження на плавність у функцію втрат дозволило усунути артефакти у відновлених швидкостях без збільшення обчислювальних витрат. Крім того, правильне вагове співвідношення функцій втрат для даних та фізичних обмежень забезпечує стійкість до шуму.

Висновки. ФІЗНМ ефективно розв'язують прямі та обернені задачі поширення одиночного імпульсу в неоднорідному середовищі, забезпечуючи точність на рівні МСЕ для прямих задач і перевершуючи стандартні PINNs в оберненому відновленні. Ці результати свідчать про значний потенціал ФІЗНМ для практичних вимірювань та візуалізації. У майбутніх дослідженнях планується розширити ці методи для сценаріїв з багатокomпонентними імпульсами.

Ключові слова: Фізично-інформовані нейронні мережі, ФІЗНМ, одиночний імпульс, обернена задача, реконструкція швидкості, неоднорідне середовище