Physics-informed neural networks based on DeepONet for ocean wave propagation

Jiankai Wang Jeffrey Harris Marissa Yates

LHSV, ENPC, Institut Polytechnique de Paris, EDF R&D, Chatou, France

Congrès Interdisplinaire sur les Modèles Avancés de Vagues,
11-16 mai 2025 Aussois

Outline

Like a little baby growing up

What is Physics–Informed Neural Networks (PINNs) ?

1. Hybrid approach combining:
   - Data-driven Neural Networks
   - Physics-based constraints
2. Key innovation:
   - Physics equations as regularization in loss function
3. Advantages:
   - No mesh dependency
   - Enhanced interpretability
4. Challenges:
   - Balancing data fidelity and physical constraints
   - Training difficulties

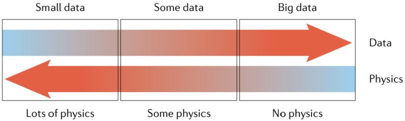
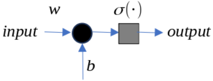


Fig. 1 Data and physics scenarios. (Karniadakis et al. 2021)



$$MSE = \lambda_{data} \underbrace{\frac{1}{N}\sum_{i=1}^{N}\left|\hat{u}(x_i) - u(x_i)\right|^2}_{MSE_{data}} + \lambda_{physics} \underbrace{\frac{1}{N}\sum_{i=1}^{N}\left|\mathcal{N}(u)\right|^2}_{MSE_{physics}}$$

$\mathcal{N}(\bullet)$ denotes a nonlinear differential operator

How it works ?

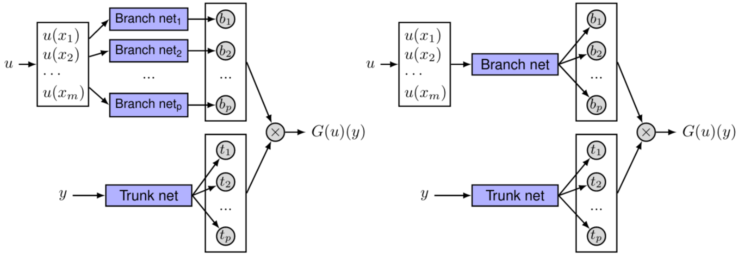


Fig. 2 Stacked and unstacked DeepONet structure.(Lu et al. 2021)

- Branch Net: extract features from input functions: $u_x \mapsto b$
- Trunk Net: extract spatial-temporal features and discretize the output function G(u) in space and time: $y \mapsto t$

$$G(u)(y) \approx \sum_{k=1}^{p} b_k . t_k$$

Why it works ?

Universal Approximation Theorem for Operator

Suppose that σ is a continuous non-polynominal function, X is a Banach Space, $K_1 \subset X$, $K_2 \subset \mathbb{R}$ are two compact sets in X and $\mathbb{R}$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p, m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $\omega_k \in \mathbb{R}$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$, and $j = 1, \ldots, m$ such that

$$\left| \left(G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{Branch} \underbrace{\sigma\left(\omega_k . y + \zeta_k\right)}_{Trunk} \right) \right| < \epsilon$$

holds for all $u \in V$ and $y \in K_2$.

Linear/Airy wave theory

In the case of a flat bathymetry,

Free surface elevation $\eta(x,t)$:

$$\eta(x,t) = \frac{H}{2} cos(kx - \omega t)$$

Under the condition that the linear dispersion relation is satisfied:

$$\omega^2 = gk\tanh(kh)$$

Analytical solution of the velocity potential $\phi(x,z,t)$:

$$\phi(x,z,t) = \frac{Ag}{\omega}\frac{cosh(k(z+h))}{cosh(kh)} sin(kx - \omega t)$$

Fully Nonlinear Potential Flow

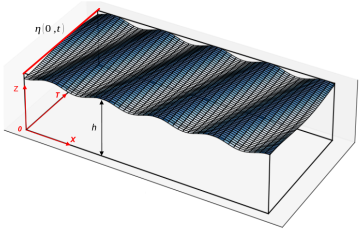


Fig. 3 Water wave propagation in (x, z, t).

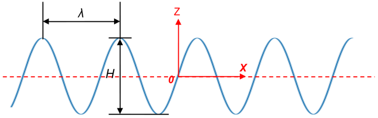


Fig. 4 Sinusoidal wave free surface.

Physical constraints

- The governing equation:

$$\nabla^2\phi = 0$$

- Bottom boundary:

$$u \cdot n = \frac{\partial \phi}{\partial n} = 0$$

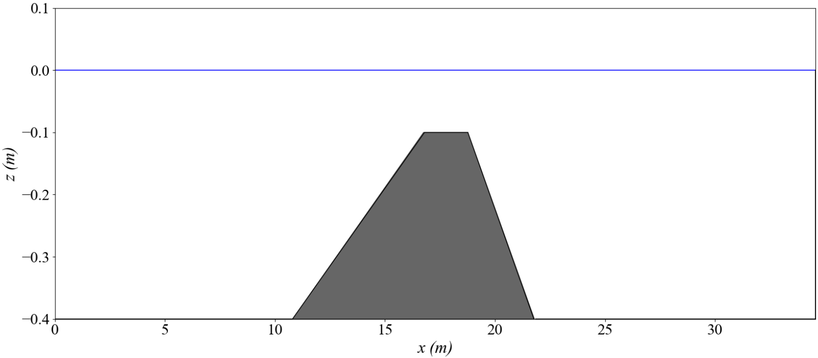
- Free surface boundary:

$$\frac{\partial \eta}{\partial t} = \frac{\partial \phi}{\partial z} - \frac{\partial \eta}{\partial x}\frac{\partial \phi}{\partial x}$$

$$\frac{\partial \phi}{\partial t} = -g\eta - \frac{1}{2}|\nabla\phi|^2 - \frac{p_a}{\rho}$$

Boundary elements method: Numerical Wave tank model (Grilli, Guyenne, and Dias 2001)

- Laplace equation is solved as a boundary integral equation derived from Green's second identity, discretized with a high-order BEM(Harris et al. 2022).



Ani. 1 Periodic wave propagation over a bar.

Network structure

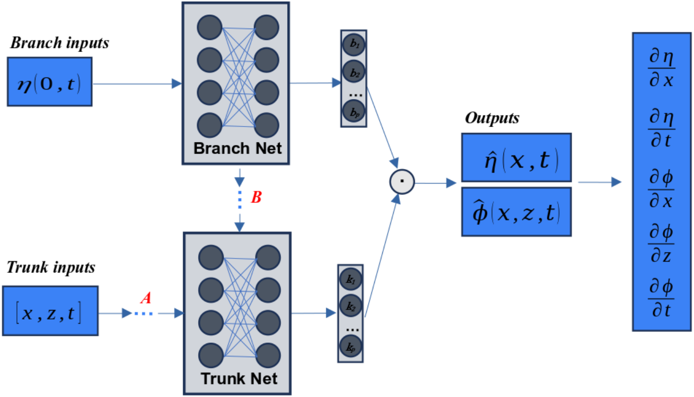


Fig. 5 Mapping from boundary condition $\eta(0, t)$ and spatial-temporal coordinates to the elevation of the free surface η and the velocity potential ϕ: *A* denotes the preprocessing of trunk net inputs and *B* denotes the residual connection.

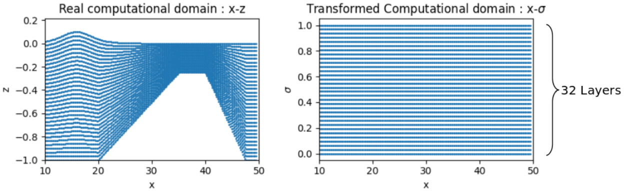
A: Preprocessing

$[x, z, t]$ —(σ' transformation)→ $[x, \sigma', t]$ —(Fourier features encoding)→ $[x, \sigma', t, \alpha_1 \cos(2^i x), \alpha_2 \sin(2^i x), \alpha_3 \cos(2^i t), \alpha_4 \sin(2^i t)]$

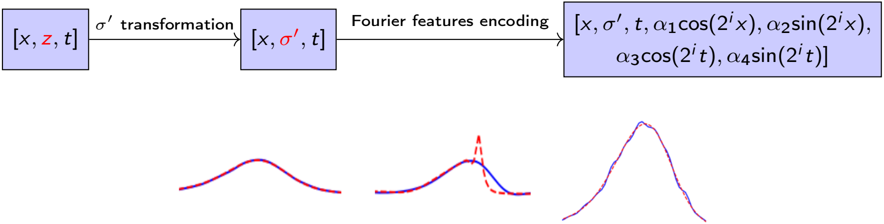
σ transformation

The time-varying computational domain is fixed, with bathymetry embedded in the variable σ'

$$\sigma' = \frac{z + h(x)}{h(x) + \eta(x, t)} \in [0, 1]$$



A: Preprocessing



Fourier features encoding

The input coordinates are extended using Fourier feature encoding to enhance the ability to predict high-frequency information. Where $i = 0, 1, 2, ..., n$, and α_i is a trainable variable that automatically controls the expression of different frequency components.

B: Residual/Skip connection(Peyvan and Kumar 2025)

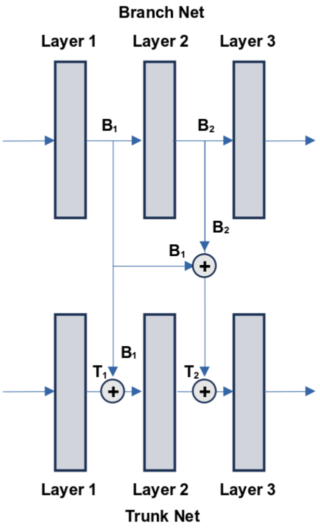


Fig. 6 Connection link

- Branch Net:
  - $B_1 = \textbf{Layer1}(\eta(0,t))$
  - $B_2 = \textbf{Layer2}(B_1)$
  - $B_{\text{output}} = \textbf{Layer3}(B_1)$
- Trunk Net:
  - $T_1 = \textbf{Layer1}(\text{coords}[x, \sigma', t, ...])$
  - $T_2 = \textbf{Layer2}(B_1 + T_1)$
  - $T_{\text{output}} = \textbf{Layer3}(B_1 + B_2 + T_2)$

How are physical constraints are embedded into the network ?

Due to automatic differentiation in neural networks, the gradients of the outputs with respect to the inputs, that is, the partial derivatives could be obtained.

Residuals of physical equations

$$\mathcal{L}_{\text{Laplace}} = \frac{\partial^2 \tilde{\phi}}{\partial x^2} + \frac{\partial^2 \tilde{\phi}}{\partial \sigma'^2} \left| \nabla \sigma' \right|^2 + 2 \frac{\partial^2 \tilde{\phi}}{\partial x \partial \sigma'} \frac{\partial \sigma'}{\partial x} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \nabla^2 \sigma'$$

$$\mathcal{L}_{\text{Kinematic}} = \frac{\partial \eta}{\partial t} - \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial z} + \frac{\partial \eta}{\partial x} \left(\frac{\partial \tilde{\phi}}{\partial x} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial x} \right)$$

$$\mathcal{L}_{\text{Dynamic}} = \frac{\partial \tilde{\phi}}{\partial t} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial t} + g\eta + \frac{1}{2} \left[\left(\frac{\partial \tilde{\phi}}{\partial x} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial x} \right)^2 + \left(\frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial z} \right)^2 \right]$$

$$\mathcal{L}_{\text{Bottom}} = \left(\frac{\partial \tilde{\phi}}{\partial x} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial x} \right) \frac{dh}{dx} + \frac{\partial \tilde{\phi}}{\partial \sigma'} \frac{\partial \sigma'}{\partial z}$$

How about data-driven components ?

Data-driven loss functions

$$\mathcal{L}_\eta = \frac{1}{N}\sum_{i=1}^{N}(\eta - \hat{\eta})^2$$

$$\mathcal{L}_\phi = \frac{1}{N}\sum_{i=1}^{N}\left(\phi - \hat{\phi}\right)^2$$

How to balance them ? (Wang, Yu, and Perdikaris 2022)

Total loss function

$$\mathcal{L}_{\text{total}} = \lambda_1 \cdot \mathcal{L}_\eta + \lambda_2 \cdot \mathcal{L}_\phi + \lambda_3 \cdot \mathcal{L}_{\text{Laplace}} + \lambda_4 \cdot \mathcal{L}_{\text{Kinematic}} + \lambda_5 \cdot \mathcal{L}_{\text{Dynamic}} + \lambda_6 \cdot \mathcal{L}_{\text{Bottom}}$$

Neural Tangent Kernel weighting

$$NTK_i = \|\nabla \mathcal{L}_i\|_2 = \left\| \frac{\partial \mathcal{L}_i}{\partial \theta} \right\|_2 \Rightarrow NTK_{max} = \max_{1 \leq i \leq 6} NTK_i$$

$$\Rightarrow \lambda_i = \frac{NTK_{max}}{NTK_i}$$

Trainable variables of network (→ θ)

How does the network learn ?

The dataset can consist of analytical solutions, numerical solutions, or even experimental data.

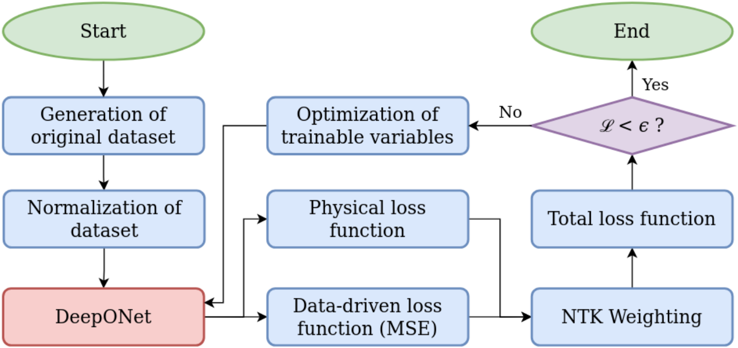
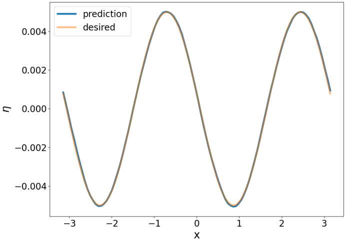


Fig. 7 Training flow chart

Reproducing a sinusoidal wave with a flat bottom.



Prediction of η

$\eta(x,t) = \frac{H}{2} cos(kx - \omega t)$

The predictions closely match the desired analytical solutions

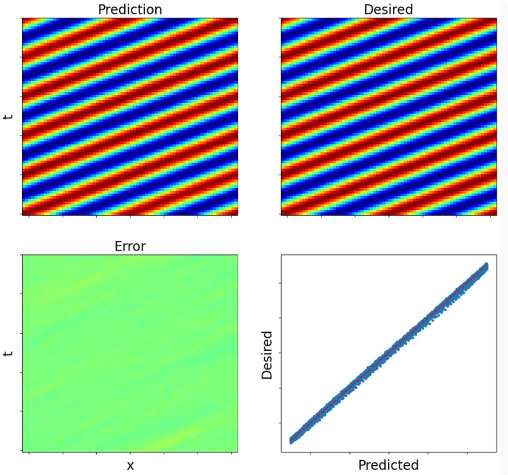
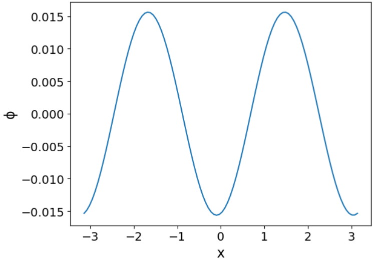


Fig. 8 Relative L2 error of η : 2.407%

Reproducing a sinusoidal wave with a flat bottom.



Prediction of ϕ

$$\phi(x,\eta,t) = \frac{Ag}{\omega}\frac{cosh(k(\eta+h))}{cosh(kh)}sin(kx-\omega t)$$

The predictions closely match the desired analytical solutions

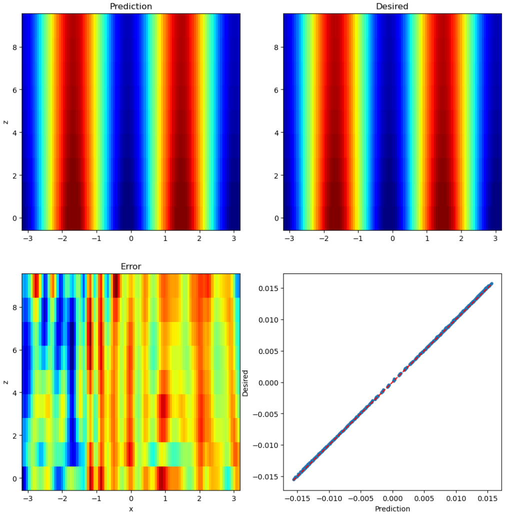


Fig. 9 Relative L2 error of ϕ at the free surface : 1.456%

Bathymetry and solitary wave properties

A trapezoidal bar with a front slope 1/20 and a rear slope 1/10 (Beji and Battjes 1993).

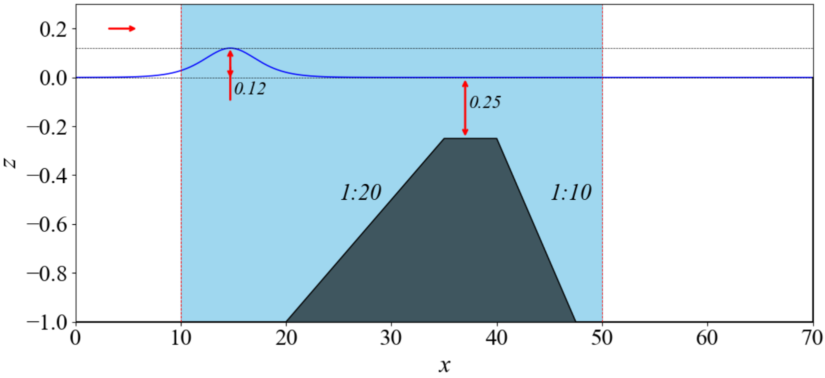


Fig. 10 Solitary wave with a height of 0.12 propagating over a bar

Prediction at the free surface

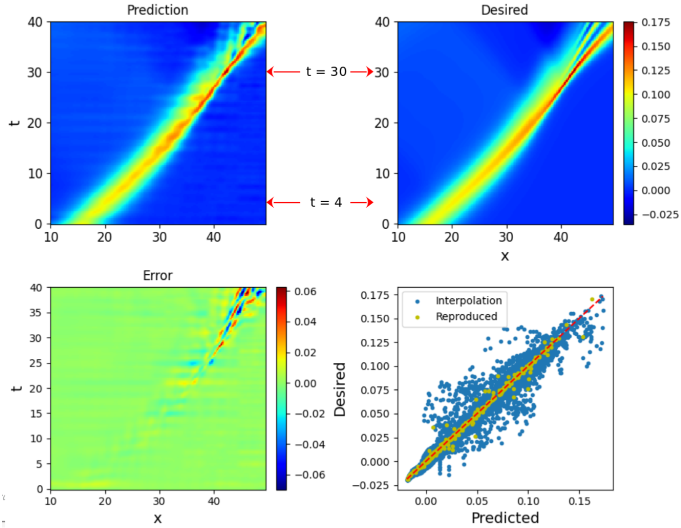


Fig. 11 Comparison of free surface elevation: Relative L2 error: 13.71% and 6.98%

Prediction at the free surface

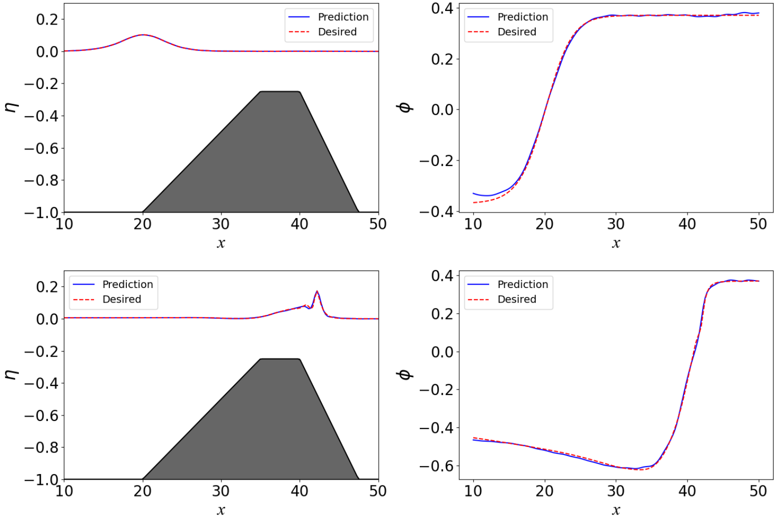
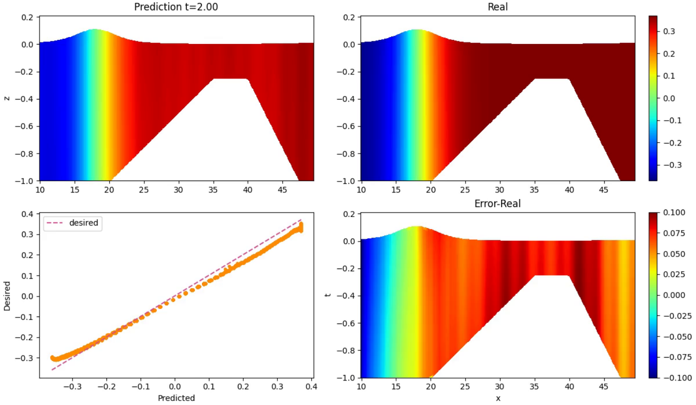


Fig. 12 Free surface elevation and potential at t = 4.0 and 30.0

Prediction of ϕ over the entire domain



Ani. 2 Solitary wave propagation over time.

Prediction of ϕ over the entire domain

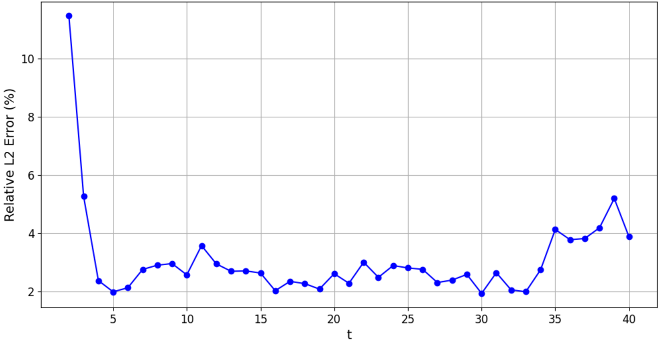
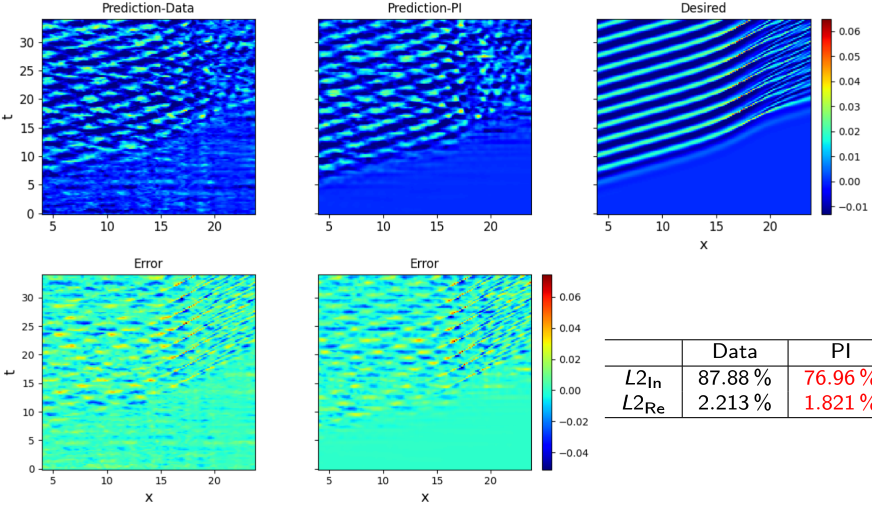


Fig. 13 Relative L2 error variation over time ($\bar{L2} = 3.08\%$)

Prediction at the free Surface



	Data	PI
$L2_{\mathrm{In}}$	87.88 %	76.96 %
$L2_{\mathrm{Re}}$	2.213 %	1.821 %

Fig. 14 Comparison of free surface elevation with and without physical constraints.

Prediction at the free surface

Currently, the neural network can accurately reconstruct the training data, but its generalization to unseen data is limited.

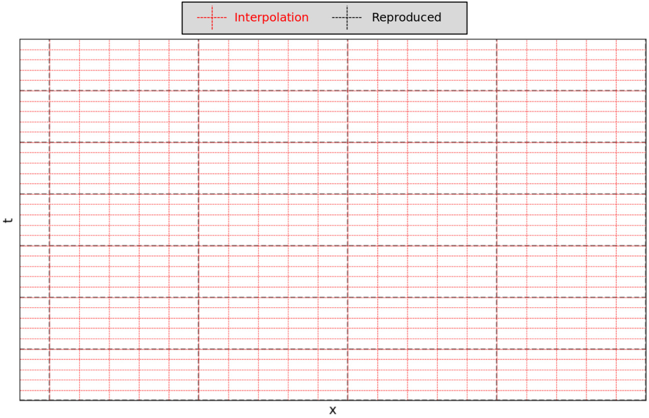


Fig. 15 Reproduced vs Interpolation

Prediction at the free surface

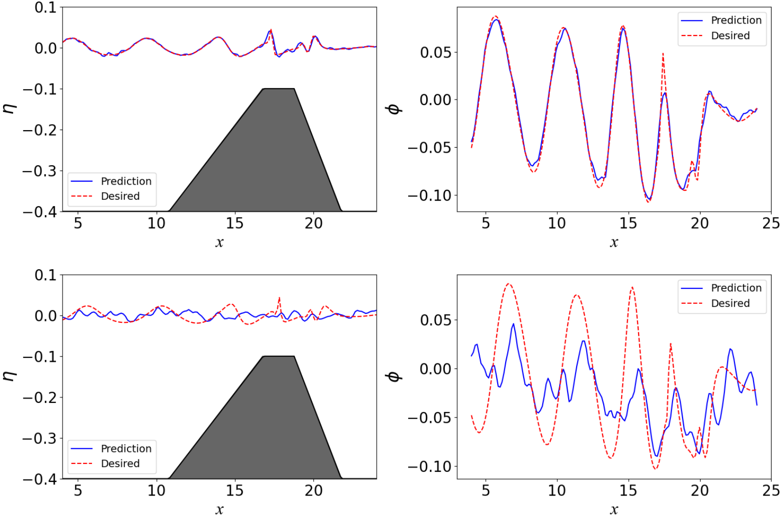
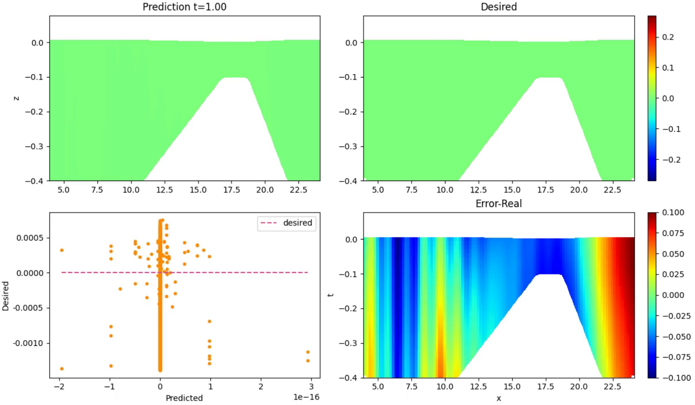


Fig. 16 Free surface elevation and potential at t = 18.0 and 18.5

Prediction of ϕ over the entire domain



Ani. 3 Sinusoidal wave propagation over time.

Prediction of ϕ over the entire domain

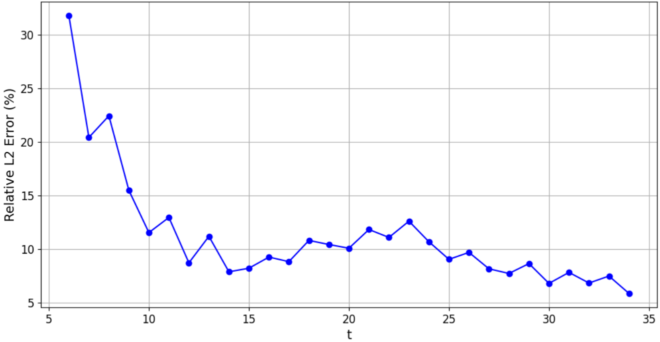


Fig. 17 Relative L2 error variation over time ($\bar{L2} = 11.17\%$)

Conclusion

1. The sinusoidal wave without bathymetry variation is accurately reproduced;
2. With bathymetry variation, the propagation of both solitary waves and periodic sinusoidal waves is also well reproduced;
3. However, both cases demonstrate limited predictive capability for unseen data

Future work

1. Enhancing the model's generalization;
2. Further improving its ability to learn features at different frequencies;
3. Training with multiple waves simultaneously...

Thank you for your attention !