



# OPEN Rayleigh-wave dispersion data selection and model fine-tuning based on uncertainty estimation

Xijun Feng<sup>1✉</sup>, Fen Zhang<sup>1✉</sup>, Wen Peng<sup>2</sup> & Fei Deng<sup>1</sup>

Rayleigh-wave inversion is a reliable approach for obtaining subsurface shear-wave velocity structures and holds significant importance in seismic risk assessment, resource exploration, and geotechnical engineering. Numerous studies have demonstrated the great potential of deep learning (DL) in Rayleigh-wave inversion; however, existing DL methods still suffer from limited generalization, strong dependence on training data, and slow convergence. To address these issues, this study proposes a representative data selection and model optimization strategy. Specifically, we identify high-uncertainty samples based on the inconsistency of predictions from multiple pretrained models trained in parallel. An automatic differentiation-driven inversion method is then used to generate high-confidence pseudo-labels for the selected data, which are subsequently employed to fine-tune the original model. This workflow requires no borehole information and significantly improves the prediction accuracy and robustness of the model in the target area. Both synthetic and field experiments validate the effectiveness of the proposed method, demonstrating enhanced adaptability and performance in complex geological environments with relatively small additional cost.

**Keywords** Neural networks, Rayleigh wave dispersion, Inverse problems, Uncertainty estimation

Shear wave velocity of the subsurface directly reflects the stiffness of underground materials and plays a pivotal role in groundwater exploration, engineering geology, and environmental studies<sup>1</sup>. Based on the dispersive propagation characteristics of surface waves in heterogeneous media, Surface wave methods—such as SASW(Spectral analysis of surface waves) and MASW(Multichannel analysis of surface waves)—can derive subsurface shear-wave velocity profiles by analyzing Rayleigh-wave dispersion curves. Owing to their high efficiency, low cost, and minimal environmental impact, these techniques have become the mainstream approach for obtaining shear wave velocity structures<sup>2–4</sup>. The complete surface wave analysis workflow comprises three core stages: field data acquisition, dispersion characteristic analysis, and dispersion-curve inversion. In particular, inverting the Rayleigh wave dispersion curve is a key step in surface wave analysis<sup>5</sup>, as the accuracy of this inversion directly determines the precision and reliability of the resulting Vs model.

Early Rayleigh wave dispersion curve inversion techniques relied predominantly on optimization algorithms, which can be classified into two broad categories: linear local methods and nonlinear global methods. Common linear approaches include Damped Least Squares<sup>6</sup>, Singular Value Decomposition, and Occam's inversion. Linear local optimization methods iteratively approximate the optimal solution by linearizing the forward model in the vicinity of an initial guess, relying heavily on accurate parameter derivatives and well-chosen starting models. This dependence on initialization and precise gradient computation significantly limits their practicality. Nonlinear global strategies—such as Genetic Algorithms<sup>7</sup>, Simulated Annealing<sup>8</sup>, Particle Swarm Optimization, and Sparrow Search<sup>9</sup>—offer broader search capabilities with reduced dependence on initialization. However, they suffer from excessive computational demands, low convergence efficiency, and a tendency to become trapped in locally optimal solutions, rendering them impractical for large scale dispersion curve inversion<sup>10</sup>. In summary, traditional optimization methods for Rayleigh wave inversion are generally constrained by low computational efficiency and strong non uniqueness.

To overcome the limitations of traditional optimization algorithms in Rayleigh wave dispersion curve inversion, more efficient and robust deep learning methods have garnered extensive attention in recent years. Once fully trained, a deep neural network can produce accurate predictions in a single forward pass, effectively balancing computational speed and inversion precision. Early efforts predominantly utilized fully connected

<sup>1</sup>College of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China.

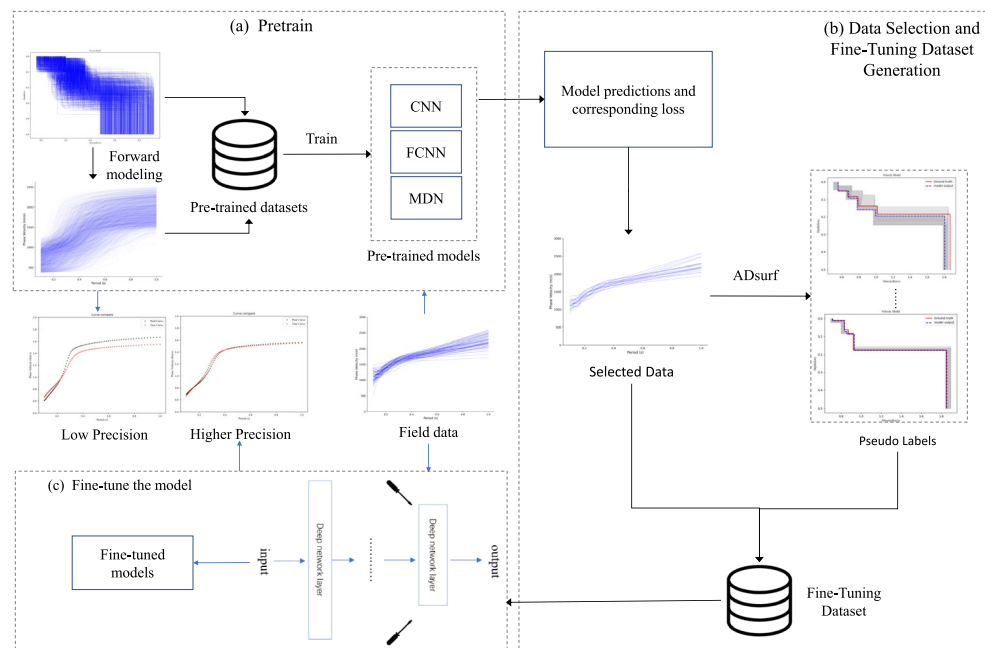
<sup>2</sup>Exploration Geophysics Technology Research Center, BGP Inc., CNPC, Zhuozhou 072750, China. ✉email: 2023020949@stu.cdut.edu.cn; zhangfen@cdut.cn

neural networks (FCNNs), with multiple successful applications demonstrating their substantial potential<sup>11,12</sup>. As the field has advanced, various enhancements have emerged. For example, Earp et al.<sup>13</sup> and Yang et al.<sup>14</sup> employed mixture density networks (MDNs) to infer shear wave velocity structures, enhancing prediction reliability through probabilistic modeling; He et al.<sup>15</sup> were the first to apply convolutional neural networks (CNNs) to field datasets, validating the suitability of CNNs for dispersion curve inversion; and Chen et al.<sup>16</sup> improved the loss function and incorporated geological priors into the synthetic data generation process, enabling CNNs to capture local geological variations in the target region, thereby mitigating inversion non uniqueness and improving predictive accuracy under complex geological settings.

Although deep learning method has demonstrated high computational efficiency and prediction accuracy in Rayleigh wave dispersion curve inversion, it still faces significant challenges in practical applications, particularly limited generalization ability and strong dependence on its training data. Specifically, as a data driven approach, its predictive performance closely depends on the training dataset and often fails to predict out of distribution samples accurately. Consequently, when the application context changes, prediction accuracy on target-region data typically degrades substantially<sup>10</sup>. The common remedy involves reconstructing a geological parameter search space based on prior knowledge of the target area, randomly generating numerous shear wave velocity ( $V_s$ ) models within that space, computing the corresponding dispersion curves through forward modeling and then retraining the network using these synthetic datasets—a process that is both time consuming and labor intensive<sup>17</sup>. Moreover, accurate geological information for the target region is rarely available in practice, which forces the use of overly broad search spaces to ensure coverage of all plausible subsurface scenarios. However, such broad spaces dilute the proportion of field relevant samples, thereby degrading the model's predictive accuracy on target data. Notably, Yang et al.<sup>18</sup> showed that training a model with only a small number of high quality synthetic samples that closely resemble field measurements can significantly improve performance while dramatically reducing data requirements, emphasizing that sample quality is more important than quantity. Inspired by these findings and by concepts from uncertainty sampling, and active learning<sup>19,20</sup>, this paper proposes a method that selects representative field data via multi model prediction uncertainty and uses them for model fine tuning to enhance prediction accuracy in the target region.

## Related work

Fig. 1 illustrates the overall workflow of our approach, which integrates multi-model fusion with uncertainty-driven sample selection to enable targeted model fine-tuning. The method comprises three main stages. First, multiple models are pretrained in parallel on a large synthetic dataset. Second, these models are applied to field data and their prediction discrepancies are evaluated to identify high-uncertainty samples. Third, high-confidence pseudo-labels are generated for the selected samples. The high-uncertainty samples and their pseudo-labels are aggregated into a fine-tuning subset, and only the last two linear layers of each model are



**Fig. 1.** Workflow of the proposed method. During pretraining, a large synthetic dataset—consisting of randomly generated subsurface shear-wave velocity models and their corresponding dispersion curves computed via forward modeling—is used to train multiple model architectures. Next, samples exhibiting high predictive uncertainty are identified from field data based on discrepancies across the pretrained models, and corresponding pseudo-labels are generated using the ADsurf inversion method to form a fine-tuning subset. Finally, all models undergo targeted fine-tuning to enhance prediction accuracy in the target region.

fine-tuned. This targeted fine-tuning substantially improves prediction accuracy and stability on complex target-region samples while preserving overall generalization capability.

### Dispersion-curve inversion model

In this study, we leverage a parallel training approach to develop multiple models that differ in architecture, neuron count, and parameter initialization. This approach is designed to improve overall predictive accuracy and to supply high-quality, diverse initial solutions for subsequent pseudo-label generation. Specifically, we incorporate three mainstream deep-learning architectures applied to Rayleigh wave dispersion curve inversion: fully connected neural networks (FCNNs), convolutional neural networks (CNNs), and mixture density networks (MDNs).

#### (1) FCNN

An FCNN consists of multiple dense (fully connected) layers, each followed by a nonlinear activation function. Due to its simplicity and ease of implementation, FCNN is commonly applied to regression problems. In this study, the FCNN takes a sequence of phase velocities (sampled from the dispersion curve) as input and predicts the subsurface shear wave velocity model (excluding thickness for the half space). During training, we used mean squared error (MSE) as the loss function. Compared to mean absolute error (MAE), MSE is more sensitive to outliers and has been widely applied and validated in regression problems<sup>10</sup>.

$$MSE = \frac{1}{n} \sum_{i=1}^n \frac{\Delta v_i}{v_i} + \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{\Delta h_i}{h_i} \quad (1)$$

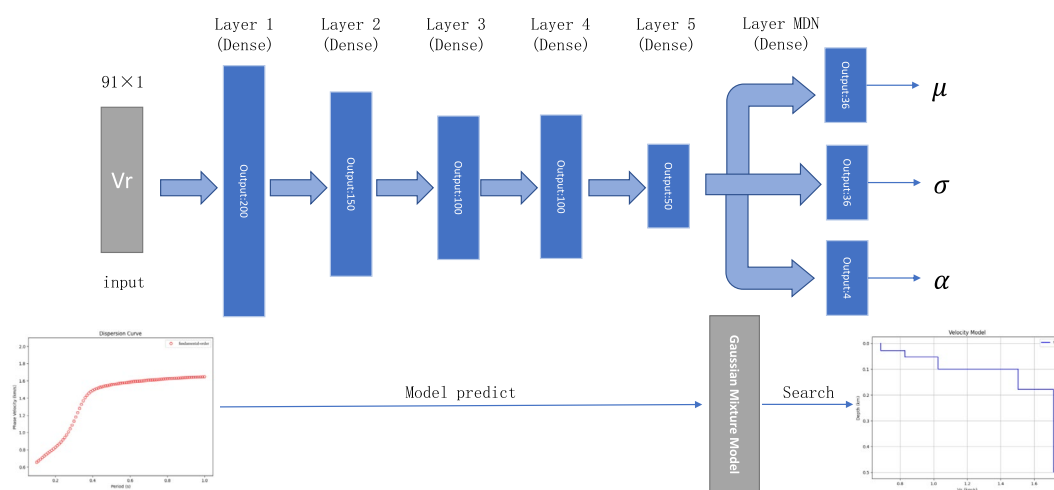
In the formula,  $n$  denotes the number of training samples,  $\Delta v_i$  and  $\Delta h_i$  represent the differences between the predicted and true shear wave velocity and layer thickness for the  $i$ th layer, respectively, while  $v_i$  and  $h_i$  are the true shear wave velocity and thickness of the  $i$ th layer. Since the bottommost layer is modeled as a half space with infinite thickness, its thickness term is excluded from the loss function.

#### (2) MDN

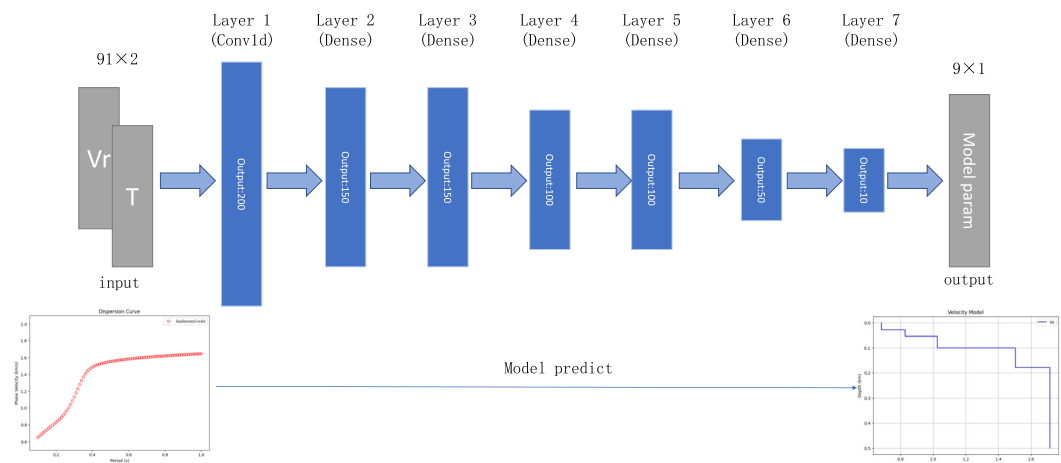
An MDN is a neural network model designed to capture complex conditional probability distributions. Compared to traditional deterministic models, it can represent underlying uncertainty and better reflect physical reality. In this study, the MDN takes the phase-velocity sequence sampled from dispersion-curve as input and outputs the Gaussian mixture model (GMM) parameters: mixture weights, means, and standard deviations (the network structure is shown in Fig. 2).

1. mixture weights ( $\alpha$ ) indicate the contribution of each Gaussian component and satisfy  $\sum_{i=1}^k \alpha_i = 1$ , where  $k$  is the number of components.
2. means ( $\mu$ ) represent the central values of the Gaussian components.
3. standard deviations ( $\sigma$ ) characterize the spread of each Gaussian and quantify uncertainty.

To obtain the most probable subsurface shear-wave velocity model, we employ a grid-search strategy within predefined physical constraints (for example, restricting the first layer's shear-wave velocity to 10–3000 m/s with a 1 m/s sampling interval). For each candidate value, we compute its probability density under the MDN's output



**Fig. 2.** Architecture of the MDN model. Arrows indicate the flow of data through the network. The final MDN layer consists of three dense sublayers that compute the mixture weights ( $\alpha$ ), means ( $\mu$ ), and standard deviations ( $\sigma$ ), respectively; all other dense layers use the Tanh activation function.



**Fig. 3.** Architecture of the CNN network. The input consists of two sequences—period and phase velocity—sampled at 91 points along the dispersion curve. A Conv1d layer with output dimension 1 and kernel size 1 merges these two sequences into a single 1D feature array, which is then passed to the subsequent dense layers. Both the CNN and FCNN models use the ReLU activation function in their dense layers.

	Hidden Layers and Nodes	Activations	Gaussian Components(K)
FCNN	[150, 150, 100, 100, 50]	ReLU	-
MDN	[200, 150, 100, 100, 50]	Tanh	4
CNN	[200, 150, 150, 100, 100, 50]	ReLU	-

**Table 1.** Selected values for the hyperparameters and activations.

and select the value of highest likelihood as the optimal solution for that layer. This process is repeated iteratively for successive layers until the full velocity model is reconstructed.

The MDN is optimized during training using the negative log-likelihood (NLL) loss function (Eq. 2). Here,  $N$  denotes the number of training samples.  $\hat{P}_{X|Y=y_i}(x_i)$  represents the posterior probability density of the true shear-wave velocity label  $x_i$  given the input phase-velocity sequence  $y_i$ , calculated as the weighted sum of Gaussian component densities. Specifically,  $\hat{p}_j(x_i)$  is the probability density of under the  $j$ th Gaussian component, and  $\alpha_j$  is its mixture weight. The parameter  $k$  indicates the number of Gaussian components in the mixture model<sup>14</sup>.

$$\text{NLL} = - \sum_{i=0}^{N-1} \log(\hat{P}_{X|Y=y_i}(x_i)) = - \sum_{i=0}^{N-1} \log\left(\sum_{j=1}^k \alpha_j \hat{p}_j(x_i)\right) \quad (2)$$

### (3) CNN

Chen et al. introduced a one dimensional convolutional layer (Conv1d) preceding a FCNN. This Conv1d layer fuses the sampled period sequence and phase velocity sequence of the dispersion curve into a single channel 1D feature vector, which is then fed into the following fully connected layers to emulate complex matrix operations<sup>16</sup>. Both the CNN and FCNN employ mean squared error (MSE) as the loss function (see Eq. 1). In contrast to an FCNN that takes only phase velocity values at fixed periods as inputs, this CNN architecture automatically integrates period and phase velocity information via the Conv1d layer. It thus eliminates the need for prior time alignment and provides richer time–frequency features, enhancing the network’s representational capability for complex dispersion data. (The detailed network architecture is depicted in Fig. 3)

### (4) Model pretraining

During training, multiple models are pretrained in parallel on the synthetic dataset with a learning rate of  $10^{-3}$  and an L2 regularization coefficient of  $10^{-4}$ . Model hyperparameters were chosen according to the well-known validation-set approach (see Table 1). The weights of the FCNN and CNN models were initialized using the Kaiming scheme, while the weights of the MDN model, except for those in the output layer, were initialized using the Xavier method. To enhance model diversity and improve the robustness of subsequent ensemble predictions, each model was trained three times with different random initialization parameters. During the model inference stage, we select the prediction that yields the smallest loss as the final output from the ensemble of model predictions. Compared with schemes that merge predictions by weighted averaging, this “minimum-loss selection” strategy offers two practical advantages. First, it incurs lower computational overhead: there is no need to estimate or update ensemble weights during pretraining or the subsequent rounds of fine-

tuning, which substantially reduces computational cost and improves overall efficiency<sup>21</sup>. In contrast, Qu et al.<sup>22</sup> compute approximate model weights at each training epoch using a Hessian trace-based approach, which substantially increases computational complexity. Second, it preserves the physical self-consistency of single-model predictions and avoids the non-physical smoothing or spurious intermediate solutions that can arise when averaging outputs. Consequently, in scenarios where model architectures differ substantially or where maintaining physical consistency is critical, the minimum-loss selection strategy is generally more stable and reliable than weighted fusion<sup>23</sup>.

### High-Uncertainty data selection

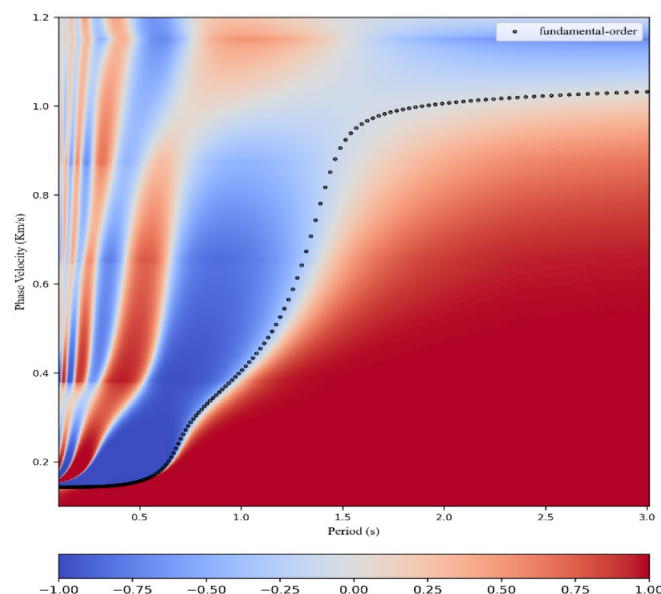
In the absence of true subsurface information, we evaluate model performance using the misfit function proposed by Ernst<sup>24,25</sup>. This misfit function calculates the mean absolute value of the determinant of the dispersion function  $F(t_i, c_i^{obs}, x)$  for a given velocity model  $x$  at the observed dispersion data points  $(t_i, c_i^{obs})$ , without requiring prior mode identification:

$$L(f, c, m) = \frac{1}{N} \sum_{i=1}^N |F(t_i, c_i^{obs}, x)| \quad (3)$$

In Eq. 3,  $N$  denotes the number of dispersion curve sampling points. The model prediction  $x$  comprises estimates of both layer thickness and shear wave velocity. The determinant  $F$  is calculated via forward simulation using the predicted model  $x$ . Each coordinate  $(t_i, c_i^{obs})$  corresponds to the  $i$ th observed dispersion sampling point's period and phase velocity. Fig. 4 presents an example of the determinant based misfit function in action. The determinant  $F$  can be computed via the frequency–Bessel (F–J) transform or phase shift methods. The theoretical dispersion curve corresponds to zeros of  $F$ ; hence, if the predicted model is accurate, all observed sampling points should lie within the white troughs (zero loci) of the determinant image, resulting in a misfit value of zero.

Building on this, we introduce the coefficient of variation (CV) as a metric for quantifying the uncertainty of multi-model predictions<sup>26</sup>. Specifically, for each sample, we first calculate the forward misfit value of the prediction produced by each pretrained model, then compute the mean  $\mu$  and standard deviation  $\sigma$  of these misfit values, and substitute them into Eq (4). The result represents the ensemble's predictive uncertainty for that sample. A larger CV indicates more significant disagreement among models, suggesting that the sample is more likely to belong to regions insufficiently covered by the training set or to lie outside the training distribution. We choose to compute CV from the models' misfit values because the raw outputs of different models are multi-dimensional and the output dimensionality or parameterization may vary when applied to field data from different regions, making it difficult to compute a CV directly from the model predictions themselves.

$$CV = \left( \frac{\sigma}{\mu} \right) \times 100\% \quad (4)$$



**Fig. 4.** An example of the misfit function defined above. The figure displays the determinant  $F$  distribution computed from the forward-modeled predictions, where white areas precisely delineate the theoretical dispersion curve. Black dots mark all observed dispersion sampling points, and the mean of the absolute determinant values at these points corresponds to the loss defined in Eq. 3.



The CV is defined as the ratio of the standard deviation to the mean, thereby eliminating the unit of measurement of the standard deviation and intuitively reflecting the relative dispersion of the data, regardless of differences in units or scales across datasets. This dimensionless property enables the CV to be applicable to data with varying signal-to-noise ratios.

### Pseudo-label construction

After identifying high-uncertainty samples using the coefficient of variation, reliable pseudo-labels must be generated to support model fine-tuning. To achieve this, we employ an automatic differentiation-driven iterative inversion method, ADsurf, to generate trustworthy labels for the selected data. ADsurf by default initializes with velocity models derived from empirical formulas and can generate multiple perturbed versions of initial model within a local neighborhood, thereby enhancing the diversity of initial solutions. During each iteration, the loss function defined in Eq. 3 is minimized using the forward-determined misfit computed via the Dunkin<sup>27</sup> and Herrmann & Ammon<sup>28</sup> enhanced Haskell–Thomson propagator. Because this forward modeling is differentiable everywhere, it allows gradient calculations via automatic differentiation (AD) and gradient-based optimization to iteratively refine the initial guesses toward realistic subsurface models<sup>29</sup>.

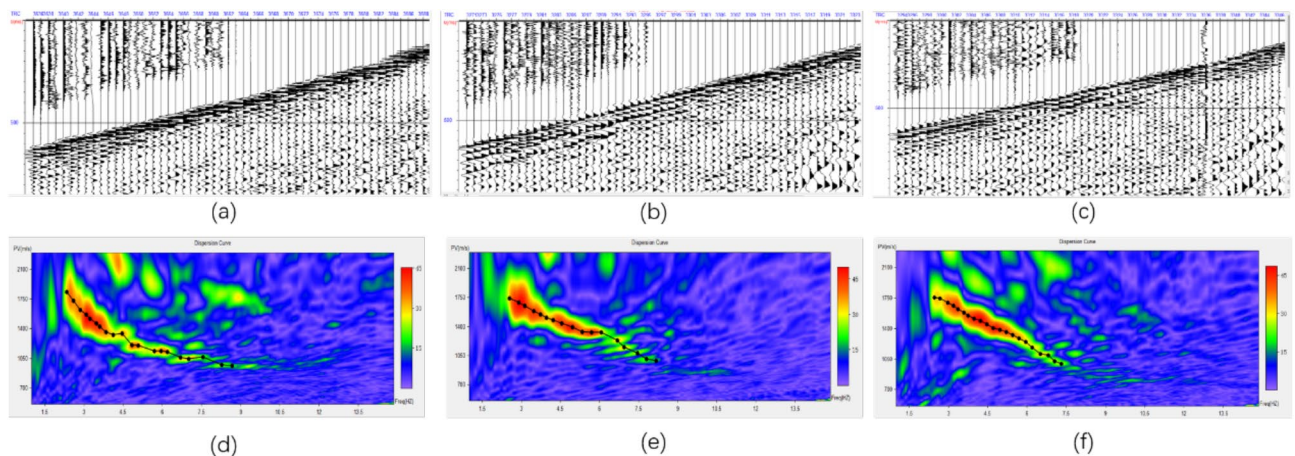
However, velocity models derived from empirical formulas often deviate considerably from the true subsurface structure. Using such models as initial solutions may cause ADsurf to exhibit slow loss reduction, unstable convergence, or even gradient explosion during iteration, which represents a major challenge for its practical application. To address this, we construct more plausible initializations using predictions from multiple pretrained models, and introduce geological prior constraints during iteration to guide convergence toward realistic solutions. Specifically, for each high-uncertainty sample, the prediction of each pretrained model is used as an independent initialization for ADsurf inversion, which is performed under the guidance of prior constraints. Each inversion produces an optimal candidate solution, and among these candidates, the one with the smallest misfit is selected as the final inversion result and used as the pseudo-label for subsequent model fine-tuning.

### Experiment

Field seismic data were acquired at an industrial site in southwest China using 62 receiver channels with an average channel spacing of 2m. The time sampling interval was 2ms, and each trace had a duration of 2.002s (1,001 time samples). After converting the seismic records into dispersion-energy spectrograms via the phase-shift method, dispersion curves were manually picked (examples are provided in Fig. 5). The resulting discrete picks were then interpolated and resampled onto a standardized period range (0.10–1.00s with a 10ms interval) to eliminate sampling nonuniformity introduced by manual picking, thereby facilitating input into the deep learning model.

Since the field data lack borehole ground-truth labels and the geological conditions of the acquisition area remain uncertain, traditional inversion algorithms often yield unstable results. To comprehensively validate the proposed method's accuracy and robustness, we adopt a two-step strategy:

1. Use randomly generated shear-wave velocity models and their corresponding dispersion curves to quantitatively assess the effectiveness of the proposed optimization scheme.
2. After validating the method with synthetic data, directly apply it to the 224 real-world dispersion curves to evaluate applicability in actual geological conditions.



**Fig. 5.** Observed seismic data are shown in panels (a)–(c), and the corresponding dispersion-energy spectrograms obtained via the phase-shift method are shown in panels (d)–(f). Black dots indicate the manually picked dispersion curves.

Layer	Vs(m/s)	Thick(m)
1	400-1000	10-100
2	$\max(600, V_{stop})-1400$	10-100
3	$\max(800, V_{stop})-1600$	10-100
4	$\max(900, V_{stop})-1700$	10-100
5	1700-2400	-

**Table 2.** Search space for synthetic pretraining dataset.

Layer	Vs(m/s)	Thick(m)
1	400-700	10-50
2	700-1000	10-50
3	800-1100	10-50
4	900-1300	50-100
5	1500-2000	-

**Table 3.** Search space for synthetic test dataset.

Mean loss	Mean relative error	Coefficient of Variation			
		> 0.7	> 0.6	> 0.5	> 0.4
0.1782	1.9516	8	18	45	154

**Table 4.** Pretrained model performance on synthetic data.

### Creating the pretraining Dataset

To construct the pretraining dataset, we define a wide parameter search space based on prior knowledge of the field data collection area (see Table 2), ensuring comprehensive coverage of plausible subsurface structures. During the generation of synthetic shear-wave velocity models, we enforce that the topmost layer has the minimum velocity while the bottommost layer attains the maximum velocity. This constraint guarantees that the synthetic models robustly produce a fundamental-mode Rayleigh-wave dispersion curve via forward modeling<sup>30</sup>.

Based on surface wave sensitivity analyses<sup>31</sup>, shear wave velocity ( $V_s$ ) exerts the most significant control on Rayleigh wave dispersion curves, followed by layer thickness. In contrast, compressional wave velocity ( $V_p$ ) and density have relatively minor effects on the computed dispersion curves. Accordingly, we compute  $V_p$  using a fixed  $V_p/V_s$  ratio of 2.45, and derive density  $\rho$  via Brocher's empirical relationship<sup>32</sup>, which relates density to  $V_p$ .

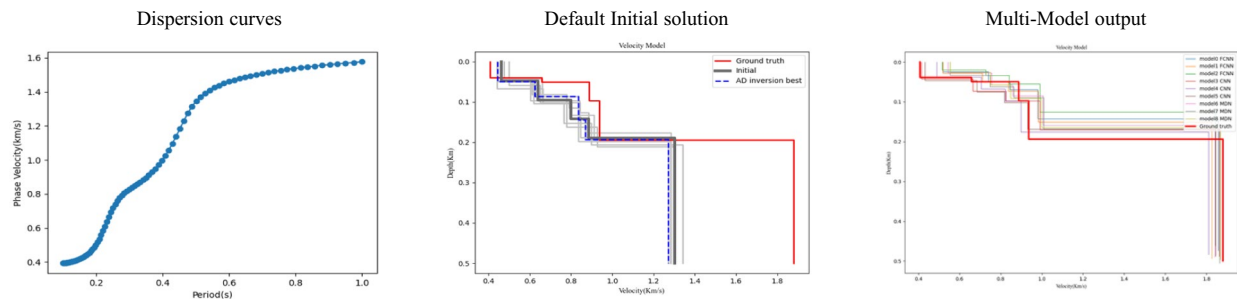
$$\rho = 1.74V_p^{0.25} \quad (5)$$

We use the `disba` Python package to generate the fundamental-mode Rayleigh-wave dispersion curves. This package implements a subset of the “Computer Programs in Seismology” (CPS) codebase<sup>28</sup> in pure Python and accelerates execution using `numba` just in time compilation, enabling efficient and convenient dispersion-curve computation. Given randomly generated velocity models, we computed phase velocities for fundamental-mode Rayleigh waves over the 0.10–1.00s period range with a sampling interval of 10ms. In total, 25,000 synthetic datasets were generated and split into training and validation subsets at a 4:1 ratio. The training subset was used for multi-model pretraining, while the validation subset was used to monitor model performance and prevent overfitting. Pretraining was conducted using the Adam optimizer, a learning rate of  $10^{-3}$ , and an L2 regularization coefficient of  $10^{-4}$ .

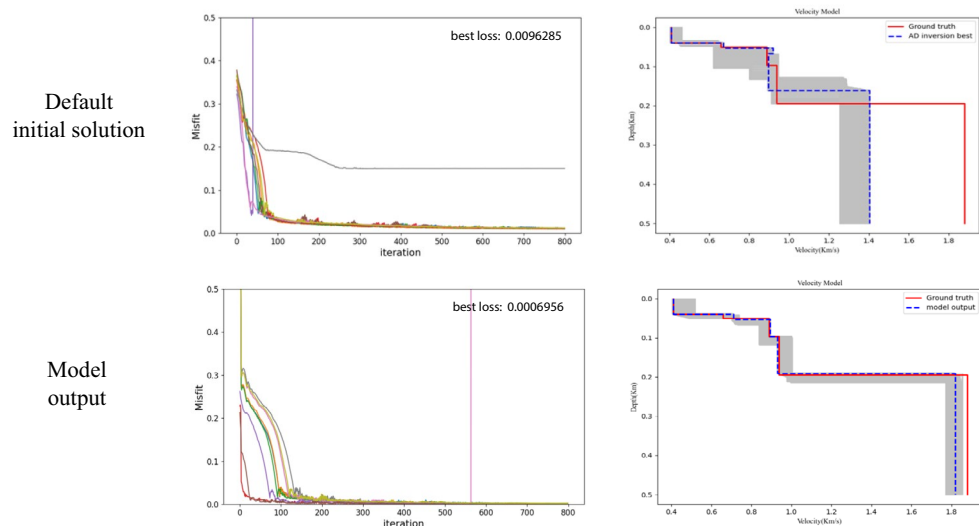
### Synthetic data

We first validate the proposed method using theoretical, noise-free data. Based on the previously described dataset generation process and the parameter space defined in Table 3, we generate 400 synthetic test cases to evaluate model performance before and after optimization.

To quantify ensemble model uncertainty, we use the CV of prediction losses across models as the evaluation metric. We set threshold values of 0.7, 0.6, 0.5, and 0.4. Whenever the CV of prediction loss for a specific sample exceeds the threshold, the ensemble's prediction for that data point is deemed highly uncertain. For the identified high-uncertainty samples, we apply the `ADsurf` package for iterative inversion. The resulting velocity models and dispersion curves are used to fine-tune the base model. The pretrained models' predictive performance on synthetic data is shown in Table 4:



**Fig. 6.** On the left are the dispersion-curve data; in the center are nine randomly generated initial models produced by the ADSurf package based on the input data; and on the right are the predictions from nine pretrained models. The red segments indicate the true labels of the data.



**Fig. 7.** In the unconstrained scenario, results are shown for both the empirically initialized solution and the initialization based on multi-model predictions, each iterated 800 times via ADSurf. The left panel displays the loss value evolution during the iteration process. The right panel shows inversion outputs: the blue dashed line represents the solution corresponding to the minimum loss, the red solid line indicates the true label, and the shaded gray region depicts how the initial solution changes through iterations.

#### Inversion results with noise-free data

To investigate the effect of the initialization strategy and constraint application on ADSurf inversion outcomes, two initialization schemes are compared in this study (see Fig. 6):

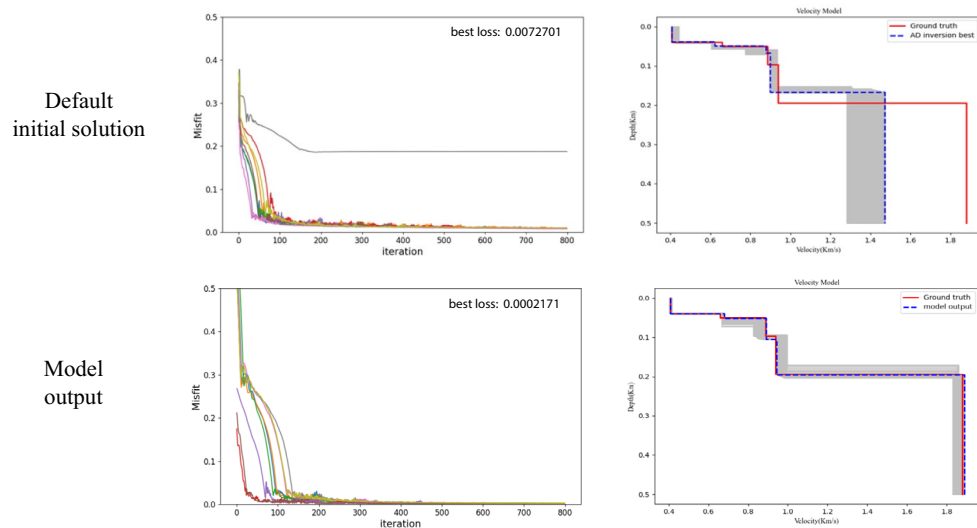
1. Predictions from multiple pretrained models;
2. The default initialization from the ADSurf package derived from empirical formulas.

Using the Adam optimizer (initial learning rate  $\eta = 10^{-3}$ , decayed by 25% every 100 iterations), each initialization strategy undergoes 800 iterations under both constrained and unconstrained settings to analyze the variations in final inversion results.

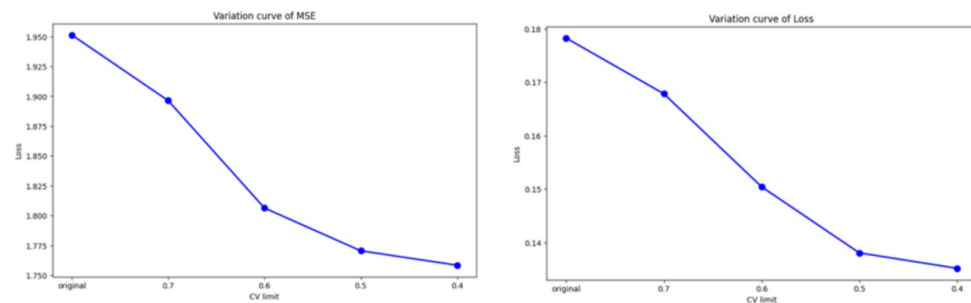
Under the assumption of constant Poisson's ratio and density, the ADSurf package generates an initial layered velocity model from observed dispersion data (period–phase velocity pairs). Specifically, each layer's thickness is calculated as  $w_{max}/depth\_factor$ , where  $w_{max}$  is the maximum observed wavelength and  $depth\_factor$  is set to 2.5. An empirical relation links Rayleigh wave wavelength to subsurface depth, with the maximum penetration depth assumed to be 0.65 times the wavelength. Shear wave velocity ( $V_s$ ) is then estimated layer-by-layer using the approximation  $V_s \approx C_{phase}/0.92$ , where the  $C_{phase}$  corresponds to the phase velocity that penetrates each layer. The computed model serves as the central solution, and ten additional initial models are randomly sampled within a small neighborhood around this solution to provide initialization diversity.

The comparison results (Fig. 7) show that, in the unconstrained scenario, using model predictions as the initial solution yields superior convergence characteristics compared to the empirical-formula-based initialization.





**Fig. 8.** Under constrained conditions, the results of 800 ADsurf iterations starting from both the empirical-formula initialization and the ensemble-model predicted initialization.



**Fig. 9.** The curves show the prediction performance of fine-tuned models under different coefficient-of-variation thresholds. The left plot depicts the variation in relative error between model predictions and true labels, while the right plot illustrates the change in prediction loss values.

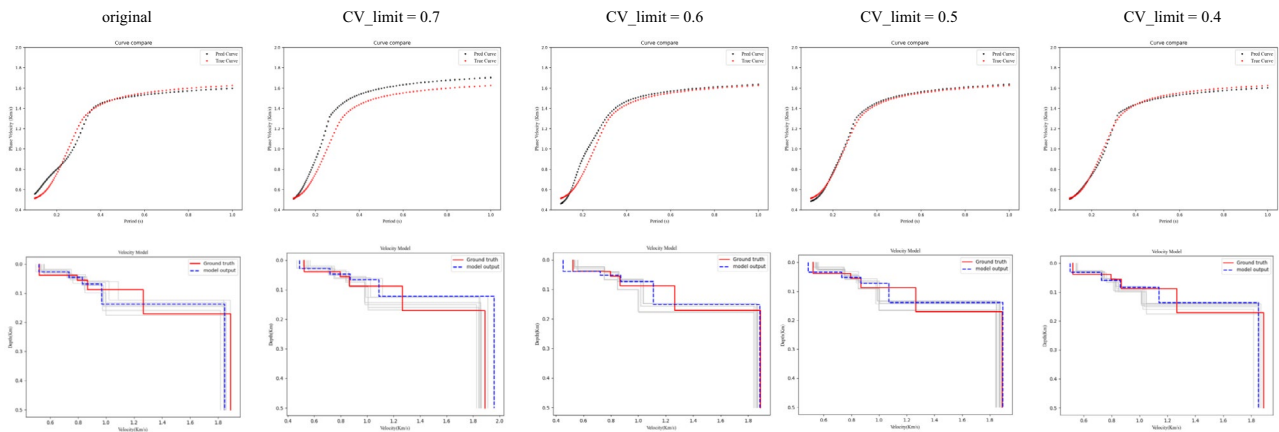
Specifically, (1) The loss decreases faster during optimization. (2) The final inversion result aligns more closely with the true label.

We subsequently applied physical constraints during iteration: the shear-wave velocity of each layer was limited to the range of 0.5 to 2 times its corresponding initial value, and layer thickness (except for the half-space) was constrained within 10–100m. Experimental results (Fig. 8) indicate that imposing reasonable physical bounds during iterative inversion enhances both convergence efficiency and final predictive accuracy.

#### Fine-tuning result

From the inversion results, we observe that, under identical inversion settings, initial models with smaller forward misfits tend to converge more rapidly and are more likely to reach geologically plausible minima. Based on this observation, and to reduce computational cost while improving inversion efficiency during the subsequent fine-tuning stage, we use only a subset of high-quality model predictions as ADsurf initializations. Specifically, candidate predictions are first ranked by their forward misfit, and the six predictions with the lowest misfits are selected as starting models for the ADsurf iterative inversion. It should be noted that the ADsurf inversion procedure exhibits inherent stochasticity; consequently, even initializations with small misfits may occasionally fail to converge or may become trapped in unfavorable local minima (see Fig. 7). Therefore, we recommend preserving diversity among the selected initializations and tailoring the selection criteria to the specific application in order to strike an appropriate balance between computational efficiency and inversion robustness.

ADsurf inversion results are used as pseudo-labels to fine-tune the last two fully connected layers of each model. During fine-tuning, the Adam optimizer is employed with a learning rate of  $10^{-4}$  and an L2 regularization coefficient of  $10^{-3}$  to prevent overfitting. Only 10 training epochs are executed. Multiple fine-tuning experiments are conducted using different coefficient-of-variation thresholds; the variation in predictive performance across these threshold values is plotted in Fig. 9.



**Fig. 10.** Prediction performance of fine-tuned models across CV thresholds.

	Proposed method	Random sample	Worst sample
Mean loss	0.1380	0.1466	0.1620
MRE	1.7705	1.8826	1.9361

**Table 5.** Comparison of results based on different data selection methods.

Comparing the fine tuned model with the original pretrained version(see Fig. 10) reveals that as the coefficient of variation threshold increases, the performance of the fine tuned model improves, and its forward-modeled dispersion curves align more closely with test data. However, the magnitude of improvement shows diminishing returns. Specifically, when the threshold reaches 0.5, further lowering the threshold to include more fine tuning samples no longer yields significant gains in predictive accuracy. This is likely because the newly added samples are highly similar to those already included, offering little additional learning benefit. Conversely, at a threshold of 0.4, the number of pseudo-label samples needed is more than twice that required at 0.5. Considering computational time and efficiency, the model fine-tuned with a CV threshold of 0.5 is selected as the optimal configuration.

When the coefficient-of-variation threshold is set to 0.5, only 45 samples need to undergo inversion processing, allowing the model to achieve good predictive performance with minimal time cost. To validate the effectiveness of the proposed method, we set the CV threshold to 0.5 and then select two groups of equal-sized samples from the test dataset: one randomly sampled and the other consisting of samples with the lowest CV values. Pseudo-labels are generated for both sets and used to fine-tune the models. Finally, we compare the performance of the model fine-tuned on the additionally selected data against that of the model fine-tuned using the proposed method. The comparison results are presented in Table 5.

The final results(Fig. 11) demonstrably indicate that the proposed method yields significantly superior prediction performance compared to random sampling and worst-case sampling. Among these approaches, predictions from worst-case sampling exhibit the largest deviation from ground-truth labels, followed by random sampling. Optimal performance is achieved through fine-tuning with data exhibiting the highest coefficient of variation. These findings substantiate the efficacy of selecting high-uncertainty data based on coefficient of variation thresholds for model refinement.

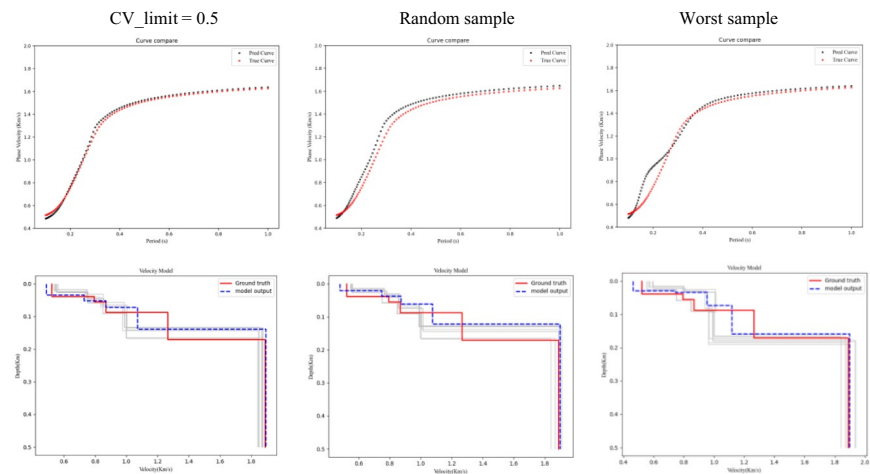
**Field data**

*Iterative inversion results with noisy data*

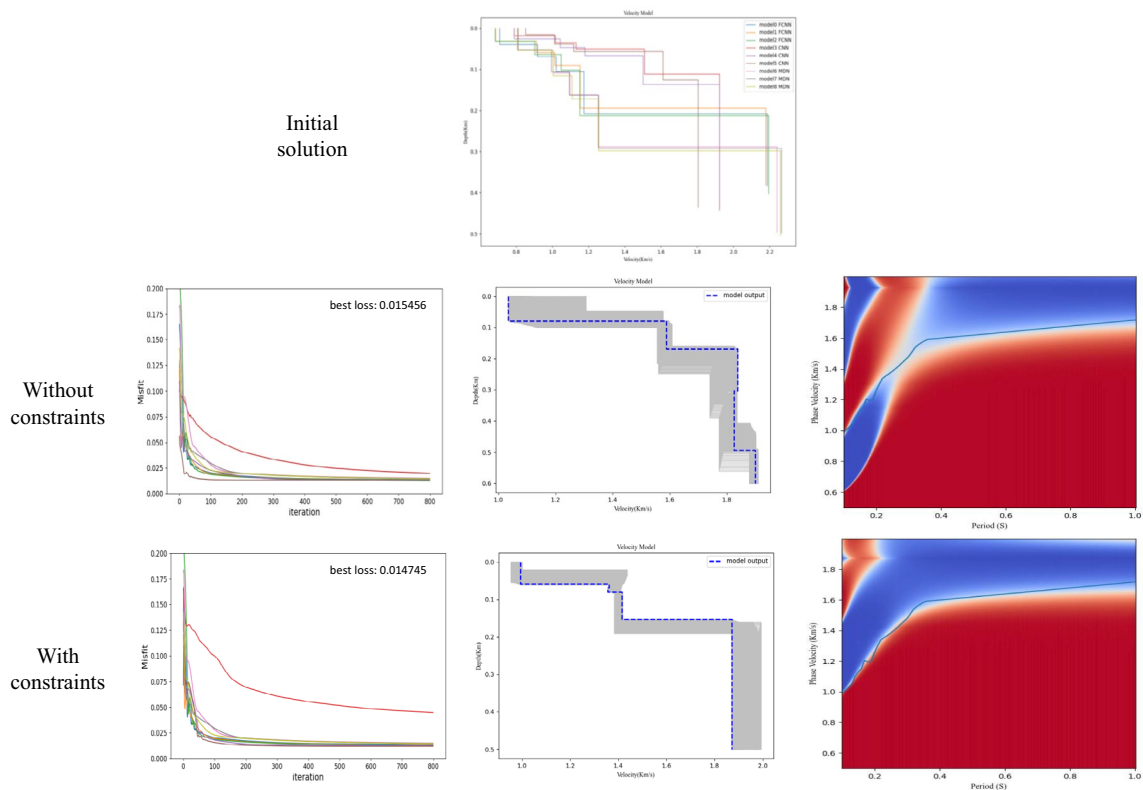
Fig. 12 demonstrates that, under unconstrained conditions, the inversion easily converges to incorrect solutions during iteration; in contrast, when appropriate constraints are applied, the initial model is guided toward more reasonable solutions, enabling rapid loss convergence and yielding inversion results that align with expectations. Despite the presence of disturbance and noise in the data, the final forward-modeled dispersion curve points all lie near the zero-value regions of the determinant, indicating a strong fit. This suggests that ADsurf can produce satisfactory inversion results even when input dispersion curves include some sampling bias or mild noise.

*Fine-tuning result*

When tuning the pretrained models using the same method, and in the absence of detailed geological knowledge about the acquisition area, we evaluate model performance using the mean loss value as the criterion. The pretrained models’ predictive performance on field data is shown in Table 6. The fine-tuning results are shown in Fig. 13 and Fig. 14. From Fig. 13, it can be observed that at CV thresholds of 0.7 and 0.6, the relatively few selected fine-tuning samples contain limited information, which is insufficient for the model to learn useful features—consequently, model performance after fine-tuning shows no significant improvement. In contrast,



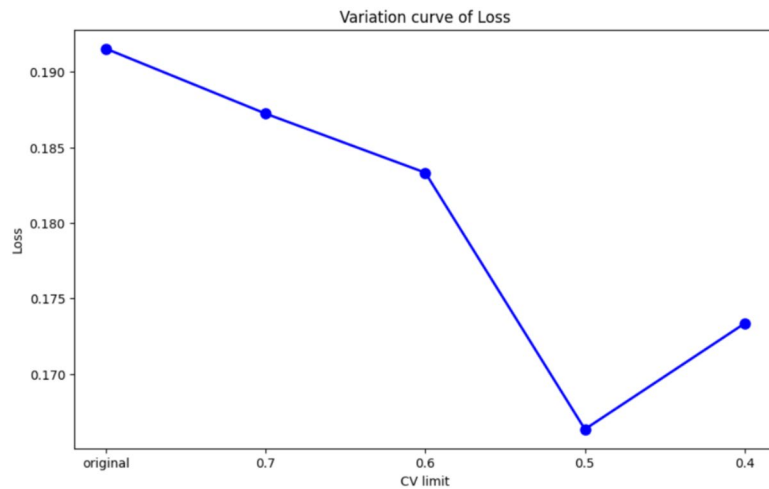
**Fig. 11.** Comparative results of three fine-tuning data selection methods: (a) the proposed method, (b) random sample selection, (c) minimum voting consensus sampling.



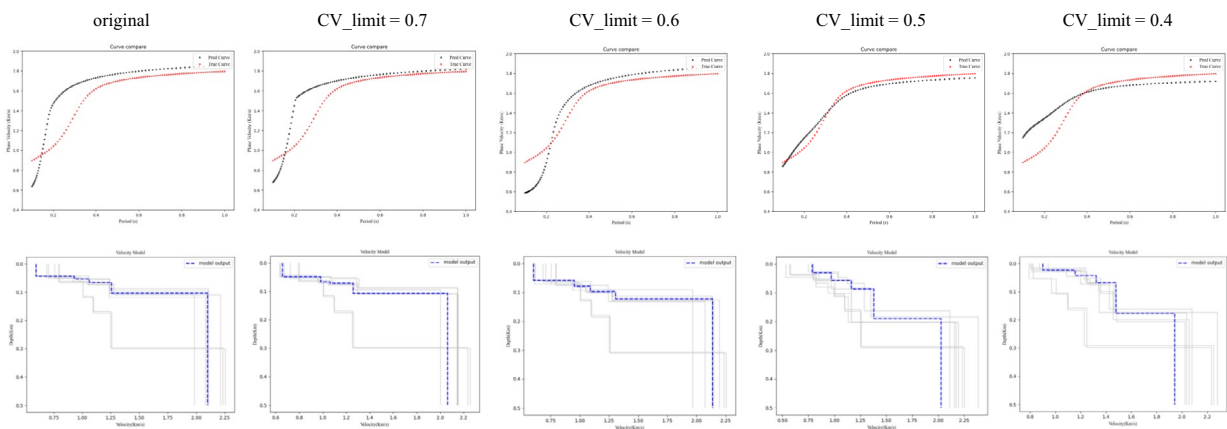
**Fig. 12.** The iterative inversion results using model predictions as initial solutions under both unconstrained and constrained conditions.

Mean loss	Mean relative error	Coefficient of Variation			
		> 0.7	> 0.6	> 0.5	> 0.4
0.1715	~	3	10	36	64

**Table 6.** Pretrained model performance on field data.



**Fig. 13.** Prediction loss curves before and after fine-tuning on the manually picked dispersion-curve data.



**Fig. 14.** Comparison of model prediction performance on dispersion curves extracted from field seismic data after applying the proposed method.

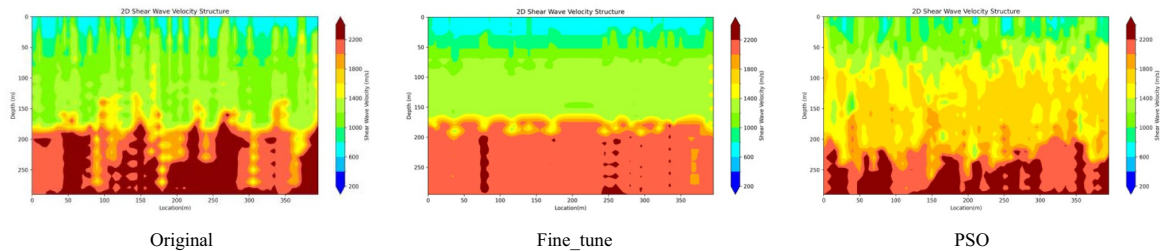
when CV thresholds are set at 0.5 and 0.4, the number of fine-tuning samples increases, and the fine-tuned model performance improves substantially—the forward-modeled dispersion curves from predictions align more closely with actual data. However, at a CV threshold of 0.4, despite using more samples for fine-tuning than at threshold 0.5, the model's performance actually degrades—likely due to an improperly set learning rate or insufficient number of fine-tuning epochs.

We selected the model fine-tuned with a CV threshold of 0.5 as optimal. When its predictions are interpolated into a subsurface profile, the resulting stratification is markedly clearer, revealing distinct layered structures, whereas the profile from the original model's outputs shows no meaningful layering above the half-space. In contrast, the PSO method produces the poorest stratification among the three approaches on real data, offering little useful information for subsurface interpretation. These comparisons indicate that, in the presence of noise or disturbances, our proposed fine-tuning strategy delivers more stable and reliable geological layering information. These comparative profiles are shown in Fig. 15.

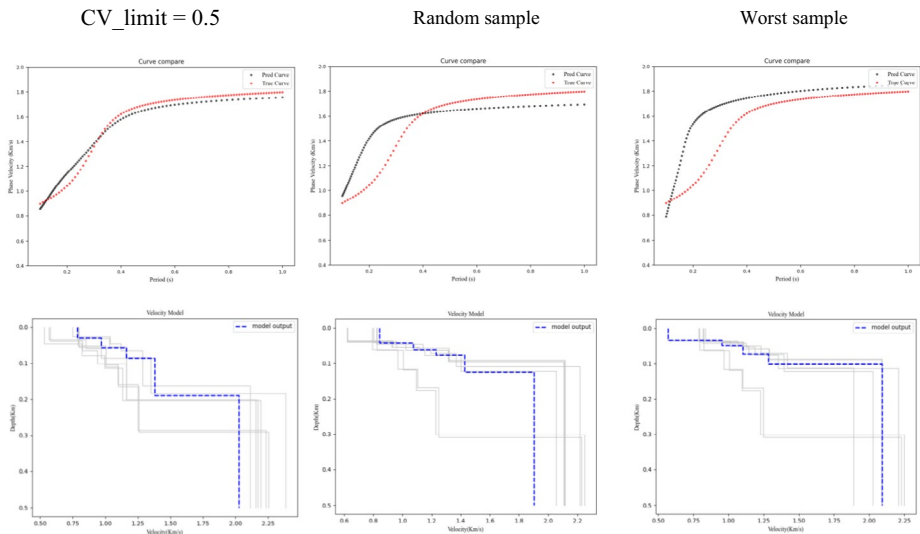
We selected the fine-tuned model obtained with a CV threshold of 0.5 as the reference model, as it yielded the best predictive performance. To further validate the effectiveness of our approach, we extracted two equal-sized sets of samples from the real test dataset: one chosen randomly and the other consisting of samples with the lowest coefficient of variation. Pseudo-labels were generated for both sets and used to fine-tune models using the same workflow. The performance of these models was then compared to that of the reference model. As shown in Fig. 16, the model fine-tuned using our proposed method achieved the highest prediction accuracy, and its forward-modeled dispersion curves matched the field data most closely. The comparison results are presented in Table 7.

## Discussion

The proposed method is based on an uncertainty sampling strategy. It identifies samples with low prediction confidence and generates high-confidence pseudo-labels for these data, which are then used to fine-tune



**Fig. 15.** Pseudo-2D Vs profiles obtained by interpolating inversion results on picked data: original model, fine-tuned model, and PSO optimization.



**Fig. 16.** Comparison of real data results from three methods: the proposed method, random sample selection, and lowest variation sample selection.

	Proposed method	Random sample	Worst sample
Mean loss	0.1663	0.1725	0.1780

**Table 7.** Comparison of results based on different data selection methods.

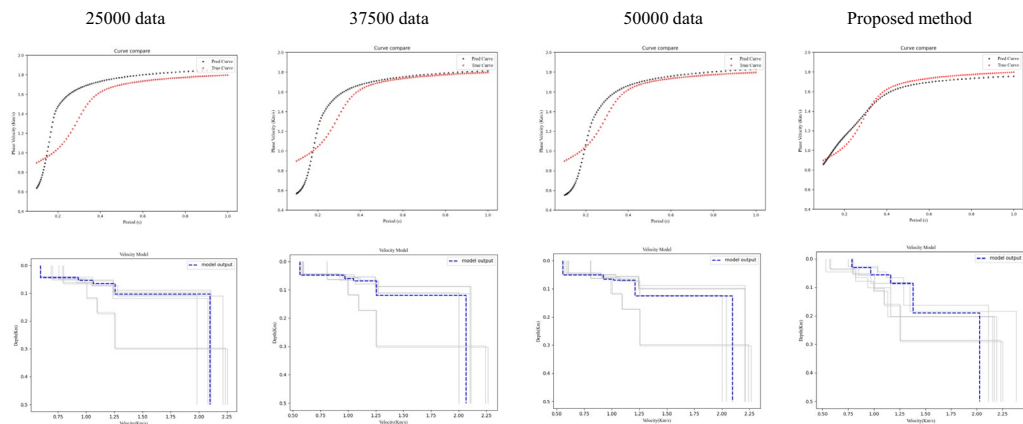
pretrained models. This enables the models to learn more informative features and improve overall prediction accuracy. The effectiveness and feasibility of the proposed approach have been validated through both synthetic and real data experiments.

**Limitations of synthetic data training**

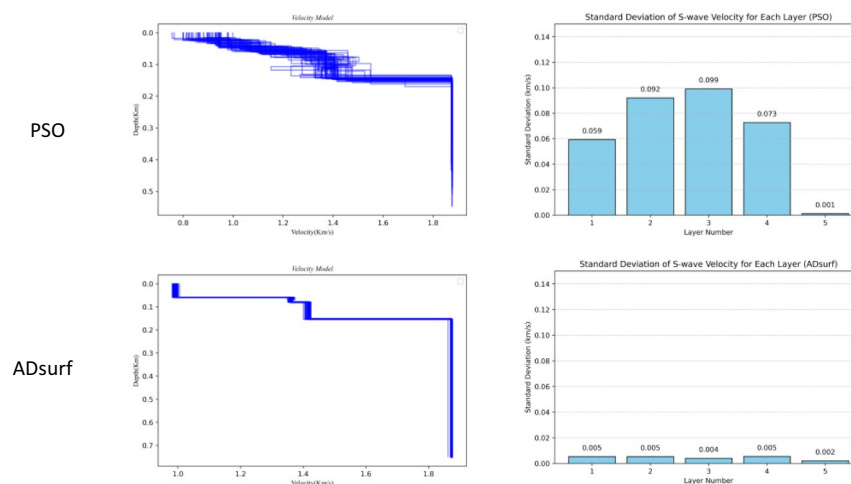
When deep learning models trained on large-scale synthetic datasets perform poorly on target data, a common strategy is to further expand the size of the training set. However, simply increasing the quantity of synthetic data does not significantly improve the model’s predictive accuracy on real-world data (Fig. 17). Possible reasons include:

- 1. An excessive number of synthetic samples may “dilute” the model’s focus on the specific characteristics of the target data. To minimize overall loss, the model may ignore the minority patterns, leading to insufficient learning of key features.
- 2. Regardless of how the synthetic rules are adjusted, synthetic data cannot fully replicate the curve deviations present in real data due to manual picking errors and environmental noise. This results in an inherent gap between synthetic and real samples, which prevents deep models that heavily rely on training data from making accurate predictions on field data.





**Fig. 17.** Comparison of model predictions obtained under different training data sizes and refinement strategies. The result reveals a critical insight: scaling up synthetic data alone is an inefficient strategy for improving field data performance. Even when the synthetic training set is doubled, its performance on field data is markedly inferior to that of a model fine-tuned with only 36 samples using our method. This comparison clearly demonstrates the superiority of our targeted fine-tuning strategy over the conventional approach of merely expanding synthetic data volume.



**Fig. 18.** Comparison of optimization results between the traditional PSO algorithm and the ADsurf method. Both PSO and ADsurf were applied to the same data with 50 independent inversions. The left panel shows the inversion results from multiple runs, while the right panel presents the standard deviation of S-wave velocity predictions for each layer, indicating the variability in the results.

Therefore, this study adopts a targeted retraining approach using a small number of real samples and their corresponding pseudo-labels. This allows the model to better capture the distribution characteristics of the target data and significantly enhance its prediction accuracy on real dispersion curves at a relatively low cost.

#### *Stability and consistency comparison between PSO and ADsurf*

Experimental results also show that velocity profiles obtained through Particle Swarm Optimization (PSO) often exhibit indistinct stratification and discontinuous interfaces. This is primarily due to PSO being a highly stochastic global optimization algorithm, which is prone to local minima. As a result, it may produce vastly different subsurface velocity structures even when inverting highly similar dispersion curves collected from the same region. In contrast, ADsurf, which is based on gradient computation, achieves highly consistent inversion results under identical initializations and optimizer settings. To quantitatively evaluate the stability difference between the two methods, we performed 50 independent inversions on the dispersion curve shown in Fig. 12 using both PSO and ADsurf. The comparison results (Fig. 18) indicate that the inversion outcomes of PSO show high variability and lack reliability, while those of ADsurf demonstrate much greater consistency and robustness.

### CV threshold selection

In this study, we set the CV threshold to 0.5 based on a practical trade-off between inversion cost and the improvement in model accuracy after fine-tuning. For our datasets and computational budget,  $CV = 0.5$  selects an adequate number of informative ‘hard’ samples and yields substantial fine-tuning gains at modest cost. It should be noted that the CV threshold is not universal: its optimal value depends on the application scenario, data distribution, and model architecture. For high-SNR data, the threshold may be raised to reduce inversion workload; for geologically complex or highly heterogeneous datasets, it may be lowered to retain more potentially informative samples. We therefore recommend a progressive, data-driven procedure: first analyze the CV distribution of multi-model prediction results for the target dataset; then start from a larger CV value (selecting only a few highly uncertain samples) and gradually lower the threshold to include more samples. At each step, generate pseudo-labels, fine-tune the models, and evaluate performance against the additional inversion cost; stop once the post-fine-tuning performance meets a predefined target (or when further lowering the threshold would exceed available computational resources). This iterative strategy controls computational expense while ensuring that the selected pseudo-labels materially improve model performance.

### Limitation

In Fig. 12, the determinant computed during the forward modeling process exhibits large regions with values close to zero, resulting in poor localization of the theoretical dispersion curve and consequently leading to inversion errors with ADsurf. This issue may stem from an overly large phase velocity search interval ( $dc$ ), which prevents the forward modeling from accurately locating the dispersion curve. Theoretically, reducing the interval  $dc$  can alleviate this problem, but it would significantly increase the computational cost. Therefore, the phase velocity search interval must be carefully selected based on practical considerations. In current practice, if abnormally low loss values occur alongside poor curve fitting—such as those shown in Fig. 12—manual screening is still required, as there is no reliable automatic method to identify and eliminate such erroneous iterations based on the output alone.

### Future

In the proposed method, dispersion-curve locations are determined from the zero-crossings of the determinant obtained by forward modeling. This approach can concurrently localize both fundamental and higher-order dispersion modes without multiple forward simulations or prior mode classification, offering good scalability. Future work will pursue two complementary directions to enhance the method. First, incorporate higher-mode Rayleigh-wave dispersion curves into the training process to leverage their richer information on deeper velocity structure, thereby improving predictive reliability and accuracy<sup>31,34</sup>. Second, the current workflow relies on manually picked dispersion curves, which is time-consuming and may introduce subjective bias; therefore, we plan to integrate automated dispersion-curve extraction techniques to replace manual picking, such as the method proposed by Hu et al. which implements automated picking with a U-net++ architecture combined with clustering algorithms<sup>33</sup>.

## Conclusion

This paper presents a model optimization method based on the concept of uncertainty sampling. By evaluating prediction uncertainty using the coefficient of variation across multiple models, the method identifies highly uncertain samples and generates high-confidence pseudo-labels for fine-tuning, without requiring any borehole data. Experimental results demonstrate that generating pseudo-labels for only a small portion of the data can significantly improve model performance in the target region. This approach effectively addresses the reduced prediction accuracy often encountered when data-driven deep learning models are applied to new areas, thereby enhancing model generalization and adaptability. Moreover, by employing more reasonable initial models and incorporating prior knowledge as physical constraints during inversion, the method substantially improves the robustness of the ADsurf algorithm under complex geological conditions. Both synthetic and field data experiments confirm that the proposed approach enhances the cross-regional generalization and adaptability of deep-learning-based inversion models, offering an efficient, low-cost, and reliable solution for Rayleigh wave dispersion curve inversion.

## Data availability

Due to confidentiality agreements with China National Petroleum Corporation (CNPC), the seismic data used in this study are not publicly available. For requests to access the dataset, please contact the corresponding author at FXJ\_cdut@outlook.com.

Received: 29 July 2025; Accepted: 26 November 2025

Published online: 05 December 2025

## References

1. Xia, J. et al. Comparing shear-wave velocity profiles inverted from multichannel surface wave with borehole measurements. *Soil Dyn. Earthq. Eng.* **22**, 181–190 (2002).
2. Olafsdottir, E. A., Erlingsson, S. & Bessason, B. Tool for analysis of multichannel analysis of surface waves (masw) field data and evaluation of shear wave velocity profiles of soils. *Can. Geotech. J.* **55**, 217–233 (2018).
3. Park, C. B., Miller, R. D. & Xia, J. Multichannel analysis of surface waves. *Geophysics* **64**, 800–808 (1999).
4. Xia, J. Estimation of near-surface shear-wave velocities and quality factors using multichannel analysis of surface-wave methods. *J. Appl. Geophys.* **103**, 140–151 (2014).

5. Socco, L. V., Foti, S. & Boiero, D. Surface-wave analysis for building near-surface velocity models—established approaches and new perspectives. *Geophysics* **75**, 75A83–75A102 (2010).
6. Cercato, M. Addressing non-uniqueness in linearized multichannel surface wave inversion. *Geophys. Prospect.* **57**, 27–47 (2009).
7. Lei, Y., Shen, H., Li, X., Wang, X. & Li, Q. Inversion of rayleigh wave dispersion curves via adaptive ga and nested dls. *Geophys. J. Int.* **218**, 547–559 (2019).
8. Calderón-Macías, C. & Luke, B. Improved parameterization to invert rayleigh-wave data for shallow profiles containing stiff inclusions. *Geophysics* **72**, U1–U10 (2007).
9. Sun, X., Ji, Z., Yang, Q. & Liu, B. Inversion of rayleigh wave dispersion curves based on an improved sparrow search algorithm. *Geophys. Geochem. Explor.* **46**, 1267–1275 (2022).
10. Meng, Q., Chen, Y., Sha, F. & Liu, T. Inversion of rayleigh wave dispersion curve extracting from ambient noise based on dnn architecture. *Appl. Sci.* **13**, 10194 (2023).
11. Devilee, R., Curtis, A. & Roy-Chowdhury, K. An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for eurasian crustal thickness. *J. Geophys. Res.: Solid Earth* **104**, 28841–28857 (1999).
12. Meier, U., Curtis, A. & Trampert, J. Global crustal thickness from neural network inversion of surface wave data. *Geophys. J. Int.* **169**, 706–722 (2007).
13. Earp, S., Curtis, A., Zhang, X. & Hansteen, F. Probabilistic neural network tomography across grane field (north sea) from surface wave dispersion data. *Geophys. J. Int.* **223**, 1741–1757 (2020).
14. Yang, J., Xu, C. & Zhang, Y. Reconstruction of the s-wave velocity via mixture density networks with a new rayleigh wave dispersion function. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13 (2022).
15. Hefei, A. C. et al. Using deep learning to derive shear wave velocity models from surface wave dispersion data. *Network* **17**, 18 (2020).
16. Chen, X., Xia, J., Pang, J., Zhou, C. & Mi, B. Deep learning inversion of rayleigh-wave dispersion curves with geological constraints for near-surface investigations. *Geophys. J. Int.* **231**, 1–14 (2022).
17. Yang, X.-H., Han, P., Yang, Z. & Chen, X. Two-stage broad learning inversion framework for shear-wave velocity estimation. *Geophysics* **88**, WA219–WA237 (2023).
18. Yang, X.-H., Zu, Q., Zhou, Y., Han, P. & Chen, X. A sample selection method for neural-network-based rayleigh wave inversion. *IEEE Trans. Geosci. Remote Sens.* **62**, 1–17 (2023).
19. Zhu, J., Wang, H., Tsou, B. K. & Ma, M. Active learning with sampling by uncertainty and density for data annotations. *IEEE Trans. Audio Speech Lang. Process.* **18**, 1323–1331 (2009).
20. Yang, Y., Ma, Z., Nie, F., Chang, X. & Hauptmann, A. G. Multi-class active learning by uncertainty sampling with diversity maximization. *Int. J. Comput. Vis.* **113**, 113–127 (2015).
21. Mu, S. & Lin, S. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137* (2025).
22. Qu, L., Araya-Polo, M. & Demanet, L. Uncertainty quantification in seismic inversion through integrated importance sampling and ensemble methods. *arXiv preprint arXiv:2409.06840* (2024).
23. Caruana, R., Niculescu-Mizil, A., Crew, G. & Ksikes, A. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, 18 (2004).
24. Ernst, F. Long-wavelength statics estimation from guided waves. In *69th EAGE Conference and Exhibition incorporating SPE EUROPEC 2007*, cp–27 (European Association of Geoscientists & Engineers, 2007).
25. Ernst, F. Multi-mode inversion for p-wave velocity and thick near-surface layers. In *Near surface 2008-14th EAGE European Meeting of Environmental and Engineering Geophysics*, cp–64 (European Association of Geoscientists & Engineers, 2008).
26. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **30** (2017).
27. Dunkin, J. W. Computation of modal solutions in layered, elastic media at high frequencies. *Bull. Seismol. Soc. Am.* **55**, 335–358 (1965).
28. Herrmann, R. B. Computer programs in seismology: An evolving tool for instruction and research. *Seismol. Res. Lett.* **84**, 1081–1088 (2013).
29. Liu, F., Li, J., Fu, L. & Lu, L. Multimodal surface wave inversion with automatic differentiation. *Geophys. J. Int.* **238**, 290–312 (2024).
30. Keil, S. & Wassermann, J. Surface wave dispersion curve inversion using mixture density networks. *Geophys. J. Int.* **235**, 401–415 (2023).
31. Xia, J., Miller, R. D., Park, C. B. & Tian, G. Inversion of high frequency surface waves with fundamental and higher modes. *J. Appl. Geophys.* **52**, 45–57 (2003).
32. Brocher, T. M. Empirical relations between elastic wavespeeds and density in the earth's crust. *Bull. Seismol. Soc. Am.* **95**, 2081–2092 (2005).
33. Hu, W. et al. Surface-wave dispersion curves extraction method from ambient noise based on u-net++ and density clustering algorithm. *J. Appl. Geophys.* **213**, 105040 (2023).
34. Pan, L., Chen, X., Wang, J., Yang, Z. & Zhang, D. Sensitivity analysis of dispersion curves of rayleigh waves with fundamental and higher modes. *Geophys. J. Int.* **216**, 1276–1303 (2019).

## Author contributions

FZ suggested the original study idea and design the method, XJF designed and completed the experiment, WP provide the data used in the experiments in this paper. FD analysed the results and written the original draft.

## Funding

This research was funded by Sichuan Province General Program Fund (2024NSFSC0514) and the Bureau of Geophysical Prospecting (BGP Inc., CNPC) under grant No. 03-02-2025

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to X.F. or F.Z.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025