JGR Solid Earth

RESEARCH ARTICLE
10.1029/2021JB023598

Special Section:
Machine learning for Solid Earth observation, modeling and understanding

Semi-Supervised Surface Wave Tomography With Wasserstein Cycle-Consistent GAN: Method and Application to Southern California Plate Boundary Region

Ao Cai[1], Hongrui Qiu[1,2], and Fenglin Niu[1,3]

[1]Department of Earth, Environmental and Planetary Sciences, Rice University, Houston, TX, USA, [2]Now at Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA, [3]State Key Laboratory of Petroleum Resources and Prospecting, and Unconventional Natural Gas Institute, China University of Petroleum, Beijing, China

Key Points:

- A machine learning based method is developed for 1-D shear wave velocity (Vs) inversion to include observed dispersion data into the training process
- The Wasserstein Cycle-GAN algorithm is used to improve training stability and spatial continuity of the output 3-D Vs model
- The final Vs model shows reasonable data misfits, sharper images of major faults, and is consistent with the large-scale surface geology

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

H. Qiu,
qiuhonrui@gmail.com;
hongruiq@mit.edu

Citation:

Cai, A., Qiu, H., & Niu, F. (2022). Semi-supervised surface wave tomography with Wasserstein cycle-consistent GAN: Method and application to Southern California plate boundary region. *Journal of Geophysical Research: Solid Earth*, *127*, e2021JB023598. https://doi.org/10.1029/2021JB023598

Received 7 NOV 2021
Accepted 9 FEB 2022

Author Contributions:

Conceptualization: Ao Cai, Hongrui Qiu
Data curation: Hongrui Qiu
Formal analysis: Ao Cai
Funding acquisition: Fenglin Niu
Methodology: Ao Cai
Supervision: Hongrui Qiu, Fenglin Niu

Abstract Machine learning algorithm has been applied to shear wave velocity (Vs) inversion in surface wave tomography, where a set of starting 1-D Vs profiles and their corresponding synthetic dispersion curves are used in network training. Previous studies showed that the performance of such trained network is dependent on the diversity of the training data set, which limits its application to previously poorly understood regions. Here, we present an improved semi-supervised algorithm-based network that takes both model-generated and observed surface wave dispersion data in the training process. The algorithm is termed Wasserstein cycle-consistent generative adversarial networks (Wasserstein Cycle-GAN [Wcycle-GAN]). Different from conventional supervised approaches, the GAN architecture enables the inclusion of unlabeled data (the observed surface wave dispersion) in the training process that can complement the model-generated data set. The cycle-consistency and Wasserstein metric significantly improve the training stability of the proposed algorithm. We benchmark the Wcycle-GAN method using 4,076 pairs of fundamental mode Rayleigh wave phase and group velocity dispersion curves derived in periods from 3 to 16 s in Southern California. The final 3-D Vs model given by the best trained network shows large-scale features consistent with the surface geology. The resulting Vs model has reasonable data misfits and provides sharper images of structures near faults in the top 15 km compared with those from conventional machine learning methods.

Plain Language Summary The speed of surface wave varies with frequency and this dispersion relation has been widely used to infer subsurface structure. Since the inversion of such dispersion relation at each grid cell can be trapped in local minima (i.e., the data misfit is not globally optimized), the accuracy of the inverted velocity model is dependent on the quality of the starting model at that location. In this study, we propose a new machine learning based inversion scheme, termed Wasserstein cycle-consistent Generative Adversarial Networks (Wasserstein Cycle-GAN [Wcycle-GAN]), to overcome the limitation. Different from conventional machine learning methods, where the training data set can only be constructed by model-generated synthetic dispersion relations, the Wcycle-GAN can incorporate both the observed and model-generated dispersion relations in the training process and further alleviates the dependence of the final model on the choice of the starting models. The proposed method is applied to data recorded in Southern California and outperforms conventional machine learning methods. The final Vs model shows sharper images of structures near faults.

1. Introduction

Machine learning (ML), particularly deep learning (Lecun et al., 2015), has attracted great attentions in geophysical fields, both in active- and passive-source seismology, such as automated seismic image segmentation (e.g., Wu et al., 2019), acoustic impedance inversion (e.g., Das et al., 2019), seismic phase picking (e.g., Ross & Ben-Zion, 2014; Ross et al., 2018; W. Zhu & Beroza, 2019), and event detection (e.g., Mousavi et al., 2020). The supervised learning such as convolutional neural networks (CNN) based methods have been widely utilized in geophysical studies. The neural networks approach has been proven to be promising in surface wave studies, for instance, extraction of crustal thickness (Cheng et al., 2019; Devilee et al., 1999; Meier et al., 2007) from surface wave data, and automatic surface wave travel time dispersion picking (e.g., Zhang et al., 2020).

Writing – original draft: Ao Cai, Hongrui Qiu
Writing – review & editing: Ao Cai, Hongrui Qiu, Fenglin Niu

In seismic tomography, ML has shown great promise for efficiently deriving velocity models from observations of seismic waves. Araya-Polo et al. (2020, 2018) used deep neural networks (DNNs) as a tomography operator to directly produce an accurate gridding or layered velocity model from waveform shot gathers. They also investigated deep recurrent architectures for improving model prediction accuracy (Adler et al., 2019). Bianco and Gerstoft (2018) developed an unsupervised learning based traveltime tomography method using dictionary learning and applied it to data recorded in Long Beach, California, to resolve high-resolution Rayleigh wave phase speed maps (Bianco et al., 2019). A more comprehensive review of ML-based seismic tomography can be found in Bergen et al. (2019) and Kong et al. (2019). In addition to tomographic methods, Xiong et al. (2021) demonstrated potential of using *K*-means clustering to evaluate velocity model.

The shear wave velocity (Vs) inversion problem in surface wave tomography, that is, mapping from surface wave velocity dispersion curves to 1-D Vs depth profiles, is highly nonlinear and underdetermined (e.g., Qiu et al., 2019). Conventional methods, such as linearized inversion (e.g., Herrmann, 2013), near-neighbor algorithm (e.g., Wathelet, 2008), and nonlinear Bayesian Markov Chain Monte Carlo method (MCMC; e.g., Roy & Romanowicz, 2017; Shen et al., 2013), are able to provide reliable results in previous studies if an initial model with sufficient accuracy at all grid locations is available. Hu et al. (2020) applied CNN based Vs inversion to Rayleigh wave dispersion data in China and the Southern California (SC) plate boundary regions. The results show the effectiveness of the CNN technique in solving the 1-D Vs inversion. However, they also demonstrate that the network trained using Vs models derived in North America does not work well for data collected in China, suggesting the quality of the training data set can affect accuracy of the output Vs model.

The workflow of the CNN based Vs inversion using surface wave dispersion data developed by Hu et al. (2020) is shown in Figure 1a. First, a labeled data set that consists of a known Vs model and its corresponding theoretical dispersion curves is split into a training set, which provides learnable examples to supervise the training of networks, and a validation set. The neural network stops updating when the training loss reaches pre-assigned threshold, and the validation set is used to avoid overfitting and tune the parameters. This is because the global optimum of deep-learning model is usually hard to achieve, and overfitting may happen in the later training epochs. The trained network is then applied to the observed dispersion data, later referred to as "unlabeled data", to output the best fitting Vs model. Since only labeled data set is used in the training process, quality of the Vs model generated from the CNN depends on whether the study area has structures similar to the input models used in training the network (e.g., Hu et al., 2020). In this study, we develop a deep-learning-based method that attempts to alleviate the dependence on the input Vs model used in the network training process.

This is done by utilizing the structure of generative adversarial networks (GAN; Goodfellow et al., 2014), in which a discriminator is introduced to enable the training process to include both the labeled and unlabeled data sets (i.e., semi-supervised; Figure 1b). Here, we use the cycle-consistent GAN (Cycle-GAN; Yi et al., 2017; J.-Y. Zhu et al., 2017), in which a data generative network that learns to reconstruct the input data from its label is added. Such network structure enforces the model and data generative subnets to be self-consistent with each other and reduces the variance of both the forward and backward generative networks by penalizing the reconstruction misfit. Compared to CNN or GAN, Cycle-GAN has been proven to generate predictions for seismic trace interpolation (e.g., Kaur et al., 2020), impedance inversion (e.g., Wang et al., 2020) and time-lapse monitoring (Zhong et al., 2020) with better accuracy under the same setup. To further improve training stability of the GAN algorithm, we adopt the structure of WGAN-GP, where the Wasserstein distance is used and a gradient penalty (GP; Gulrajani et al., 2017) is added in the adversarial loss function (Arjovsky et al., 2017). Gulrajani et al. (2017) has demonstrated the superior training stability of WGAN-GP over conventional GAN algorithms in image generation, by testing 200 generative network structures with varied activation functions, depth of networks, and number of filters in the convolutional layers, etc. The state-of-the-art hybrid method (hereinafter, Wasserstein Cycle-GAN [Wcycle-GAN]), which combines the structures of Cycle-GAN and WGAN-GP, outperforms conventional ML algorithms in biomedical translation studies (Mcdermott et al., 2018) and seismic impedance inversion (Cai et al., 2020).

The proposed Wcycle-GAN method is applied to surface wave dispersion data derived in Qiu et al. (2019) for the SC plate boundary region, one of the most well-studied areas in the world. The Community Velocity Models CVM-H15.1 (Shaw et al., 2015) is used to generate the labeled data set. We first describe the construction of the training data set, that is, a combination of the labeled data set generated using the CVM-H15.1 and the unlabeled data of the Rayleigh wave velocity dispersion maps from Qiu et al. (2019) in Section 2. The network architecture

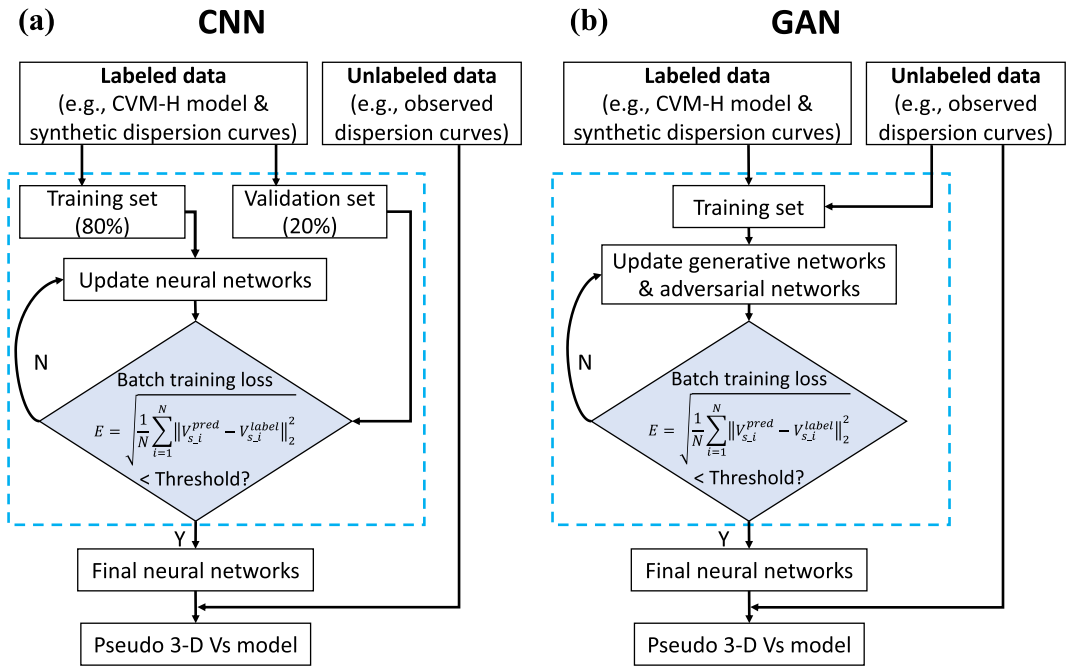

Figure 1. The flowcharts for (a) convolutional neural network (CNN) and (b) generative adversarial network (GAN) algorithms, respectively. The output is a Pseudo 3-D Vs model, that is, an assemble of all 1-D Vs model predictions. The part of chart outlined by the light blue dashed rectangular is further explained in Figure 3.

of the Wcycle-GAN and workflow of the training process are presented in Section 3. We then use the best trained Wcycle-GAN to output the final 3-D Vs model. The resulting 3-D Vs model and the corresponding data misfits are presented in Section 4.1. We further demonstrate the advantage of the Wcycle-GAN based Vs inversion by comparing the results with those obtained from the conventional CNN algorithm (Section 4.2) and linearized inversion (Section 5).

2. Data

2.1. Rayleigh Wave Phase and Group Velocities—Unlabeled Data

We use the isotropic phase and group velocity maps of the fundamental mode Rayleigh wave from Qiu et al. (2019) as the unlabeled data set, which is used in both the training process and generation of the best fitting 3-D Vs model. Qiu et al. (2019) first measured travel times of surface waves constructed from ambient noise cross-correlations using a regional seismic network with 346 stations in SC (triangles in Figure 2) over a period range of 2–20 s. Then, Eikonal tomography is applied to resolve isotropic phase and group velocity maps and corresponding uncertainties with a grid size of 0.05° × 0.05° (grid lines in Figure 2) for periods between 2.5 and 16 s. The derived Rayleigh wave velocity dispersion maps are shown in Figures S1–S4 in Supporting Information S1 and details can be found in Qiu et al. (2019).

In this study, we use velocity dispersions in the period range between 3 and 16 s to construct the unlabeled data, as the velocity maps at 2.5 s are less reliable (i.e., large uncertainties) and only cover a small part of the SC plate boundary region. Dispersion curve and its uncertainty at each grid cell are interpolated and discretized into 17 samples, with an interval of 0.5 s from 3 to 6 and 1 s from 6 to 16 s. Since the uncertainties are estimated from Eikonal tomography by analyzing velocity maps derived for different virtual sources (Section 4 of Qiu et al., 2019), uncertainty values less than 0.05 km/s are set to 0.05 km/s to account for errors from other sources (e.g., dispersion picking). Besides, grid cells with a phase or group velocity dispersion curve that has less than 8 sample points are excluded. In total, 4,076 pairs of Rayleigh wave phase and group velocity dispersion curves are selected to construct the unlabeled data set.

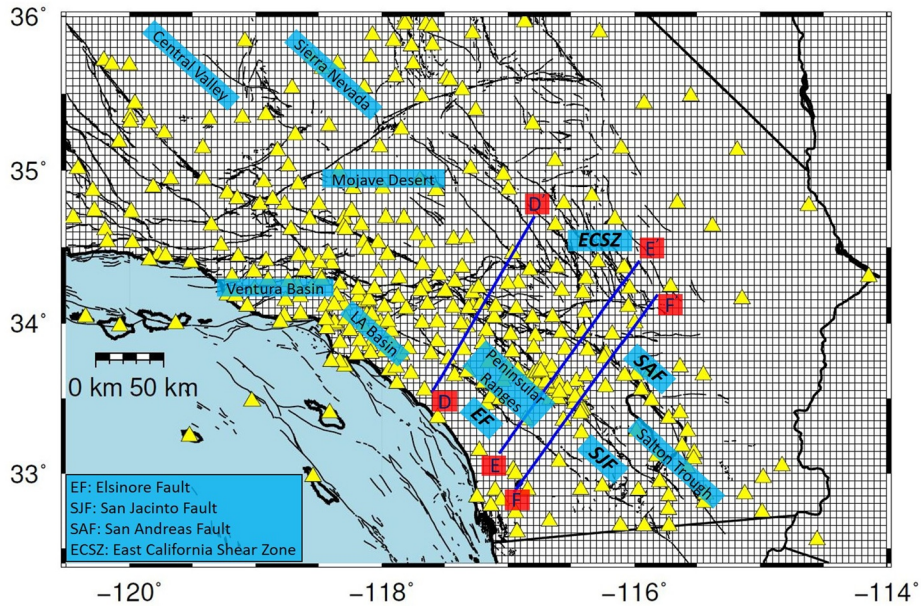

Figure 2. Map of the Southern California plate boundary region. The thick black lines depict surface traces of major faults, coastlines, and state boundaries. The yellow triangles are seismic stations used in Qiu et al. (2019) to derive the Rayleigh wave velocity dispersion maps with a grid size of 0.05° × 0.05° (grid lines). Three cross-sections (i.e., DD' to FF'; blue lines) of the final Vs model are presented in Figure 8. The cross-sections DD' to FF' are of the same locations as those in Qiu et al. (2019). SAF, San Andreas Fault; SJF, San Jacinto Fault; EF, Elsinore Fault; ECSZ, Eastern California Shear Zone.

2.2. CVM-H15.1 and Synthetic Dispersion Curves—Labeled Data

To better evaluate the seismic hazard in SC, two Community Velocity Models (CVM), CVM-H15.1 (Shaw et al., 2015) and CVM-S4.26 (Lee et al., 2014), were derived via full waveform tomography. Here, we prefer using the CVM-H15.1 (later referred to as "CVM-H") to construct the labeled data set for training the network. This is because that the CVM-H includes the effect of topography, fits the observed dispersion data slightly better, and varies less significant with depth, as discussed in Qiu et al. (2019). 16,480 1-D profiles of Vp, Vs, and density are extracted from the CVM-H with a grid spacing of 0.03° × 0.03° for the region covered by the unlabeled data (Figure 2). These 1-D profiles are then discretized into 98 layers with a thickness of 0.5 km from 0 to 49 km (relative to the earth surface) and a half space below 49 km. The study area is confined to a longitude range from 120.2°W to 114.9°W and a latitude range from 32.6°N to 36.0°N. The synthetic velocity dispersion curve is calculated using the Computer Programs in Seismology software package (Herrmann, 2013) for each selected grid cell and labeled with the corresponding 1-D Vs profile.

3. Methodology

3.1. 1-D Shear Wave Velocity (Vs) Inversion

Surface wave is dispersive, that is, travels at a frequency dependent speed, and its dispersion relation is often used to infer subsurface structure. Considering the propagation of surface wave is 2-D, traveltime-based surface wave tomography (e.g., Qiu et al., 2019) usually consists of two steps: the phase and group velocity map at each analyzed period is estimated first, and the Vs structure at each grid cell is then inverted using the dispersion curve extracted from that location. Fang et al. (2016) developed an inversion method that combines the two steps and obtains the final Vs model directly from travel time dispersion measurements, but the kernel used in the inversion is still computed through the two-step process. In this study, we focus on solving the second step, that is, the 1-D Vs inversion problem, using ML algorithm.

The essence of the 1-D Vs inversion is to search the 1-D Vs profile that minimizes the misfit between the predicted and observed surface wave dispersion curves. Traditionally, this problem is solved either through a linearized inversion (e.g., Herrmann, 2013) or through the Markov Chain Monte Carlo approach (MCMC; e.g., Berg

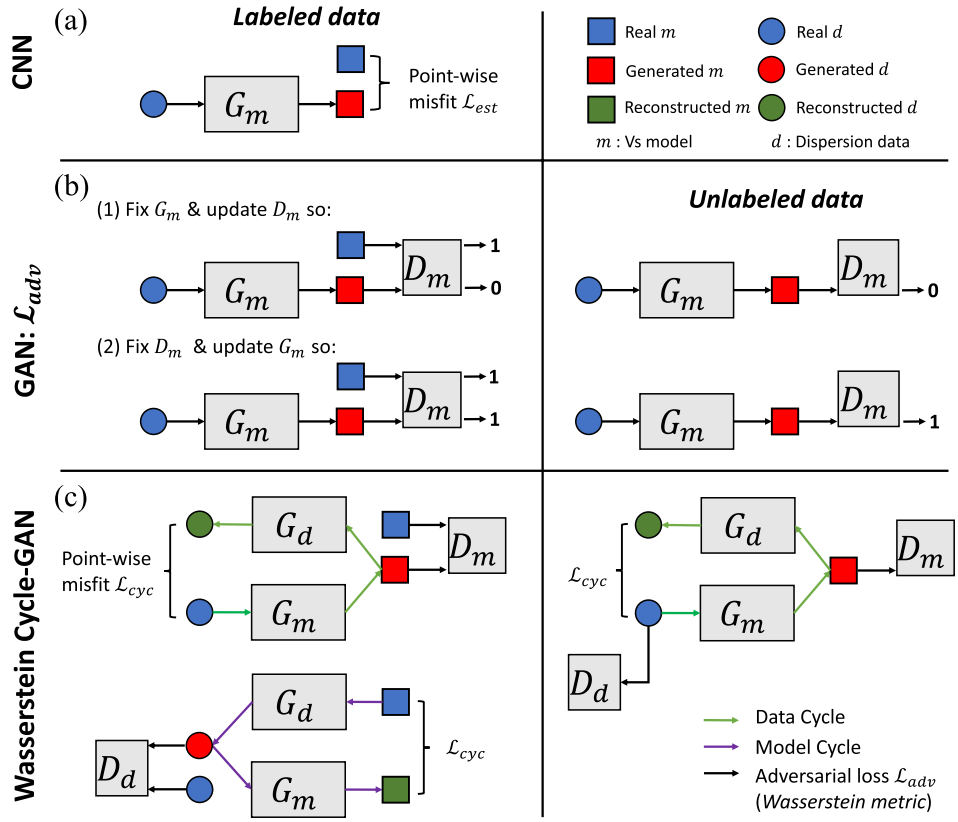

Figure 3. The algorithm comparison between convolutional neural network (CNN), generative adversarial network (GAN), and Wasserstein Cycle-GAN (Wcycle-GAN). The suffix m and d represents shear velocity model and dispersion data, respectively. (a) CNN computes point-wise misfit (estimation loss: $\mathcal{L}_{\text{est}}$) between real samples and generated samples generated by a model generative network (G_m). (b) The GAN introduces an adversarial network (D_m) and computes the difference between distributions of real and generated samples using adversarial loss ($\mathcal{L}_{\text{adv}}$), by updating generator and discriminator separately in a single iteration. (c) The Wcycle-GAN uses Wasserstein metric for adversarial loss in (b). In addition, a data generative subnet (G_d) is incorporated to learn the modeling of velocity model to dispersion data, together with a corresponding data discriminator (D_d). The use of G_d enables an extra constraint, the cycle consistent loss ($\mathcal{L}_{\text{cyc}}$), which is estimated as the misfit between the input real and reconstructed samples. The complete Wcycle-GAN penalty function is a linear combination of three types of the loss function ($\mathcal{L}_{\text{est}}$, $\mathcal{L}_{\text{adv}}$, and $\mathcal{L}_{\text{cyc}}$; Text S1 in Supporting Information S1).

et al., 2018). Since the sensitivity of surface wave velocities to Vs is much larger than those to Vp and density, the 1-D Vp (or Vp/Vs ratio) and density profiles are held fixed during the inversion. Since the inversion can be trapped in local minima, the accuracy of the best fitting Vs profile at each grid cell relies on the starting model (e.g., Qiu et al., 2019).

3.2. CNN Based Vs Inversion

Supervised ML algorithm finds an optimized mapping function from the input data to its labels through training and applies the function to unseen data to generate final predictions. Neural networks-based supervised deep learning approach has been proposed to solve the 1-D Vs inversion problem and proven to be effective in cases characterized by repeated inversion of similar dispersion data with respect to the training data set (e.g., Meier et al., 2007). Different from fully connected neural networks (e.g., Cheng et al., 2019; Devilee et al., 1999), CNN can represent more complicated mapping functions for nonlinear inverse problems (Hu et al., 2020). A typical CNN based Vs inversion consists of three steps: (a) A labeled data set that consists of some synthetic dispersion curves labeled by their corresponding 1-D Vs profiles (e.g., selected from CVM-H in this study) is generated; (b) Then, a neural network generating model prediction (G_m) is trained using the labeled data set (Figure 3a). This is done by iteratively minimizing the point-wise misfit ($\mathcal{L}_{\text{est}}$ in Figure 3a) between the generated model (i.e., G_m

prediction) and the input Vs profile (label); (c) The final pseudo 3-D Vs model is assembled by collecting all the 1-D Vs profiles predicted from the trained network at each grid cell using the observed dispersion data.

Although the performance of the CNN-based inversion still depends on the overall quality (e.g., diversity and accuracy) of the training data set, it does not require the starting model to be sufficiently accurate at every grid cell and thus, to some extent, reduces the chance of the inversion getting trapped in local minima and improve the inversion robustness. The network training usually takes several hours, however, once the neural network is trained, the prediction of 1-D Vs models using large amount of dispersion data is fast (e.g., a few minutes).

3.3. Semi-Supervised Vs Inversion Using Wcycle-GAN

To further alleviate the dependence of the CNN-based Vs inversion result on the choice of input models used in generating the training data set, we add the observed dispersion data into the network training process by using the Wcycle-GAN. The GAN structure (Section 3.3.1) enables the use of unlabeled data in the training process, whereas the cycle-consistency (Section 3.3.2) and Wasserstein metric (Section 3.3.3) further improve the robustness and training stability of the GAN algorithm. Descriptions of sub neural networks architectures and hyperparameter selections are presented in Sections 3.3.4 and 3.4, respectively.

3.3.1. Generative Adversarial Networks (GAN)

GAN incorporates two convolutional networks, a generative network (G_{m}), which is similar to that of the CNN (Figure 3a), and a discriminative network (D_{m}). Different from G_{m}, the discriminator D_{m} is designed to distinguish real Vs profiles from models predicted by G_{m} (i.e., generated model) and allows the unlabeled data to be included in the training process. D_{m} searches for a transformation that maximizes the difference between the labels and the generated Vs models, whereas G_{m} aims at minimizing it. Figure 3b briefly summarizes the training process of GAN with a schematic diagram. First, the trainable parameters are fixed in G_m, and D_m is updated to distinguish the generated Vs models from the labels (i.e., real 1-D Vs profiles) with binary outputs, that is, "0" and "1" representing the generated Vs models and the labels, respectively. Then, the trainable parameters of D_m are fixed, and G_m is updated so that D_m outputs "1" for the resulting generated models. Same procedure is simultaneously applied to unlabeled data (right panel of Figure 3b) except that there are no labels (i.e., 1-D Vs profiles) involved in the first step. The loss function commonly used in GAN is termed adversarial loss ($\mathcal{L}_{\mathrm{adv}}$) and can be calculated in various metrics, such as cross-entropy (Goodfellow et al., 2014), least squares (Mao et al., 2017) and Wasserstein distance (Arjovsky et al., 2017).

3.3.2. Cycle-Consistent GAN (Cycle-GAN)

Conventional Vs inversion methods (Section 3.1) search the final Vs model by minimizing the difference between observed and predicted dispersion data. However, small data misfits may not always be guaranteed using CNN or GAN based Vs inversion methods, particularly when the quality of the training data set is poor. This is because the neural network only learns the projection from dispersion data to its label (1-D Vs profile) using model-generated synthetic dispersion relations. Thus, we implement Cycle-GAN (Figure 3c) that further expands the GAN structure with the concept of "cycle-consistency", that is, the mapping from dispersion data to Vs model and its reversal are self-consistent. In other words, we add a sub-network that reconstructs the input dispersion data using the generated Vs model.

To enforce the neural networks to converge to invertible structures, one method is using invertible neural networks (INNs: Ardizzone et al., 2018; Zhang & Curtis, 2021). The INNs are specifically designed with a serial sequence of reversible blocks that can inherently provide bijective mappings between models and data. However, the INNs are computationally expensive for high dimensional problems and the coupling layer design may affect the expressive ability of the network (Zhang & Curtis, 2021).

Another effective approach is using Cycle-GAN, where an extra data generative (G_d) and a discriminative network (D_d) are involved, to enable the estimation of bidirectional mappings between dispersion data and Vs models. For simplicity, we separate the algorithm into data cycle (green arrows in Figure 3c) and model cycle (purple arrows in Figure 3c). In the data cycle, for the labeled data, after the computation of the adversarial loss, the generated model (i.e., output from G_m) is fed into G_d to reconstruct the original dispersion data. The reconstruction misfit, that is, cycle-consistent loss $\mathcal{L}_{\mathrm{cyc}}$, between the input and reconstructed dispersion data is minimized through the training process. In the model cycle (bottom left of Figure 3c), we obtain the generated data from the real Vs

model and estimate the adversarial loss $\mathcal{L}_{\text{adv}}$ through the data discriminator D_d. The generated data is then fed back into G_m to obtain a reconstructed model, and the cycle-consistent loss of the model reconstruction is also penalized (Figure 3c left column). The unlabeled data go through a similar process with only the data cycle part (Figure 3c right column).

The cycle-consistency loss ($\mathcal{L}_{\text{cyc}}$) for Vs inversion problem can be written as:

$$\mathcal{L}^{d}_{\text{cyc}}\left(\boldsymbol{W}_{G_m}, \boldsymbol{W}_{G_d}\right) = E\left(\boldsymbol{d}^{*}, f_{\boldsymbol{W}_{G_d}}\left(f_{\boldsymbol{W}_{G_m}}\left(\boldsymbol{d}^{*}\right)\right)\right) + E\left(\boldsymbol{d}, f_{\boldsymbol{W}_{G_d}}\left(f_{\boldsymbol{W}_{G_m}}(\boldsymbol{d})\right)\right), \tag{1}$$

For the data cycle, and

$$\mathcal{L}^{m}_{\text{cyc}}\left(\boldsymbol{W}_{G_m}, \boldsymbol{W}_{G_d}\right) = E\left(\boldsymbol{m}, f_{\boldsymbol{W}_{G_m}}\left(f_{\boldsymbol{W}_{G_d}}(\boldsymbol{m})\right)\right), \tag{2}$$

For the model cycle. $\boldsymbol{d}$ and $\boldsymbol{m}$ stand for the synthetic dispersion data and their labels (i.e., Vs models), respectively; $\boldsymbol{d}^{*}$ is the unlabeled data (observed dispersion curves); $\boldsymbol{W}_{*}$ represents the trainable parameters in the networks; $f_{\boldsymbol{W}_{*}}(*)$ is the neural network operator that generates generated samples from a specific input data set. $E(*,*)$ stands for a measurement of the difference between two samples, and, in this proposed method, is computed by mean square error between the input (e.g., $\boldsymbol{d}$) and the reconstructed output (e.g., $f_{\boldsymbol{W}_{G_d}}\left(f_{\boldsymbol{W}_{G_m}}(\boldsymbol{d})\right)$. The overall cycle-consistent loss $\mathcal{L}_{\text{cyc}}$ is the sum of the model and data cycle loss.

$$\mathcal{L}_{\text{cyc}} = \mathcal{L}^{d}_{\text{cyc}} + \mathcal{L}^{m}_{\text{cyc}}. \tag{3}$$

In summary, Cycle-GAN uses neural networks (G_d) to mimic the forward modeling from 1-D Vs profile to synthetic dispersion curve, which provides extra constraints by implementing the reversal mapping and requiring both the model and data to be cycle-consistent through the network reconstruction.

3.3.3. Wasserstein Cycle-GAN (Wcycle-GAN)

While GAN facilitates the semi-supervised learning utilizing unlabeled data, it often suffers from the training instability (e.g., mode collapse and convergence failure; Arjovsky & Bottou, 2017). A popular method to improve the training stability of GAN is to calculate the adversarial loss using the Wasserstein metric (Arjovsky et al., 2017), later referred as WGAN. Considering the real and generated Vs models are drawn from density functions of two probability distributions, the problem of fitting one distribution to another can be portraited as moving one pile of sand to another with an equal mass. Conventional penalty functions such as least squares are the sum of point-wise distances between the two piles of sands. In contrast, the Wasserstein metric, defined as the lowest cost to complete the transportation given a particular cost function, provides weaker topology comparing with the least squares or cross-entropy loss (Arjovsky et al., 2017), and thus improves the convergence of the training process. In addition to WGAN, a GP term of the discriminator is often added to the adversarial loss (Gulrajani et al., 2017) to further improve the training robustness.

Here, we use the Wasserstein metric to compute the adversarial loss of the Cycle-GAN algorithm, later referred as Wcycle-GAN. Following the notations in Section 3.3.2, the adversarial loss ($\mathcal{L}_{\text{adv}}$) in Wcycle-GAN can be written as:

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{W}\left(\boldsymbol{d}, \boldsymbol{d}^{*}, \boldsymbol{m}\right) + \lambda\ \mathcal{L}_{\text{gp}}. \tag{4}$$

Detailed mathematical expressions of Wasserstein loss $\mathcal{L}_{W}$ and GP loss $\mathcal{L}_{\text{gp}}$ can be found in Text S1 in Supporting Information S1. In practice, the weighting factor λ should be large enough to avoid exploding gradient (Gulrajani et al., 2017).

To ensure a good prediction quality in the labeled data set, the estimation loss ($\mathcal{L}_{\text{est}}$) is also penalized in the Wcycle-GAN algorithm, by computing the mean square error between the generated samples and ground truth.

$$\mathcal{L}_{\text{est}}\left(\boldsymbol{W}_{G_m}, \boldsymbol{W}_{G_d}\right) = E\left(\boldsymbol{m}, f_{\boldsymbol{W}_{G_m}}(\boldsymbol{d})\right) + E\left(\boldsymbol{d}, f_{\boldsymbol{W}_{G_d}}(\boldsymbol{m})\right). \tag{5}$$

The complete loss function of the proposed Wcycle-GAN algorithm is a combination of the adversarial loss, cycle-consistent loss, and estimation loss, defined as:

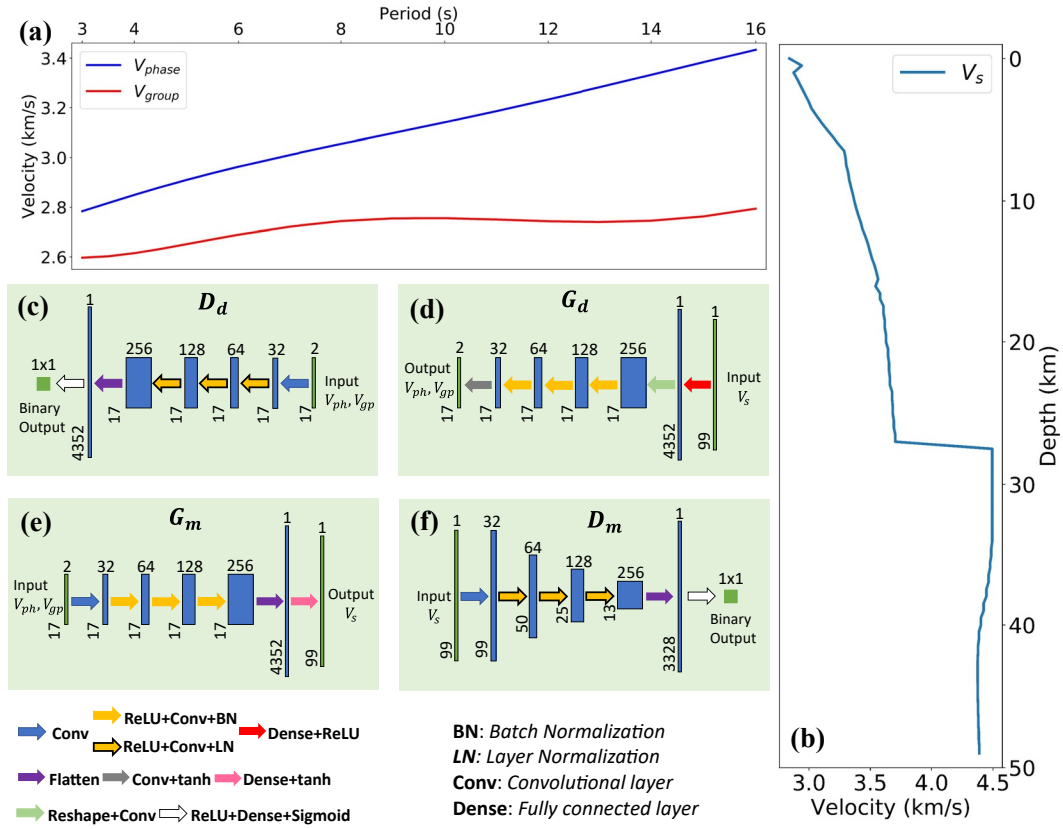

Figure 4. The detailed architectures of Wasserstein Cycle-GAN (Wcycle-GAN). For a labeled Vs—dispersion data pair, it consists of (a) the synthetic Rayleigh wave dispersion data and (b) corresponding Vs model (e.g., CVM-H). The generative and discriminative subnets (c–f) are designed based on the dimension of dispersion data (17 × 2) and Vs model (99 × 1). Each subnet is a 1-D convolutional neural network. The data discriminator (c) $\boldsymbol{D_d}$ discriminates between real data samples and the generated dispersion data, generated by a data generative subnet (d) $\boldsymbol{G_d}$ from input Vs model. The total trainable parameters in the $\boldsymbol{G_d}$ and $\boldsymbol{D_d}$ are 762,402 and 134,945, respectively. On the other side, the model generator (e) $\boldsymbol{G_m}$ learns the inverse mapping from data to model and the model discriminator (f) $\boldsymbol{D_m}$ discerns the generated model from real ones. The total trainable parameters for $\boldsymbol{G_m}$ and $\boldsymbol{D_m}$ are 759,299 and 133,825, accordingly.

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \lambda_1 \mathcal{L}_{\text{cyc}} + \lambda_2 \mathcal{L}_{\text{est}}, \tag{6}$$

where the hyperparameters λ_1 and λ_2 are the weighting factors.

3.3.4. Sub-Neural Network Structures

The proposed Wcycle-GAN consists of four sub neural networks—two generative subnets (G_m and G_d) and two discriminative subnets (D_m and D_d). All the subnets use a 1-D structure as the input data is 1-D two-channel dispersion data. The dimension of input dispersion data is 17 × 2 (Section 2), with the phase and group velocities (Figure 4a) being set as two separate channels, whereas the dimension of output Vs model (Figure 4b) is 99 × 1. Considering the imbalanced dimensions between network input and output, specific structures are designed for data and model generative subnets (Figures 4d and 4e). In the model generator, we double the number of filters at each convolutional layer in the manner of VGG16 network (Simonyan & Zisserman, 2014). The number of filters at each convolutional layer from shallow to deep is 32, 64, 128, and 256 (Figure 4e), respectively. Reversely, in the data generative subnet (G_d), we first upsample the Vs model to a dense feature map of dimension 17 × 256, and sequentially half the number of filters in the following convolutional layers. For the discriminative subnets (Figures 4c and 4f), we double the number of filters in convolutional layers and apply a sigmoid activation function in the fully connected layer to output values between 0 and 1.

In all the subnets, the convolutional layers use a kernel size of 3 × 1 and are zero-padded. The stride equals to 1 except that D_m uses a stride of 2 to reduce the number of trainable parameters. To accelerate the training

process, at each convolutional layer, we apply batch normalization (Ioffe & Szegedy, 2015) after ReLU (Nair & Hinton, 2010) activation and initialize the weight parameters in the convolutional layers using He initialization (He et al., 2015). We note that the He initialization, which is proven to be effective in our study, may not be the optimal initialization for other geoscience tasks. In addition, in the discriminative subnets, the batch normalization is replaced by layer normalization (Ba et al., 2016) as suggested in Gulrajani et al. (2017).

3.4. Evaluations of the Training Process and Output Vs Model

Before feeding dispersion data and Vs model into the neural network, we apply linear transformations (Text S2 in Supporting Information S1) to normalize them into the interval range of $[-1, 1]$, which speed up the convergence of the training process. Neural network outputs in the data and model domains are transformed back to its original amplitudes, when computing misfits or generating final Vs models. During the training process (Figure S5 in Supporting Information S1), we use root-mean-square misfit between shear wave velocities from the network prediction and the true label, that is,

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N_{\mathrm{model}}} \sum_{i=1}^{N_{\mathrm{model}}} \| V_{s\_i}^{\mathrm{pred}} - V_{s\_i}^{\mathrm{label}} \|_2^2}, \tag{7}$$

To estimate the convergence of the network training. N_{model} denotes the number of Vs models in the labeled data set, and $V_{s\_i}^{\mathrm{pred}}$ and $V_{s\_i}^{\mathrm{label}}$ are the predicted Vs models and ground truth, respectively. E_{RMS} is calculated every 25 epochs during the training process, and the training stops when E_{RMS} does not decrease in consecutive 4 estimations (i.e., 100 epochs). This results in a final $E_{\mathrm{RMS}} = \sim$0.06 km/s after 1,700 epochs for the Wcycle-GAN, whereas, for the CNN experiment in Section 4.2, the training ends at a similar misfit level (i.e., 0.06 km/s) after 200 epochs in about an hour. Although a lower root-mean-square misfit could be achieved for CNN at later epochs, but it may result in overfitting. We performed extra K-fold cross-validation with $K = 5$ (see Text S3 in Supporting Information S1) to ensure Wcycle-GAN and CNN are not overfitting at the above training stop points.

Here, CNN achieved a faster convergence in the training process compared to that in Hu et al. (2020). This is because the input dispersion curves are 1-D instead of 2-D in constructing the CNN used in this article, which results in a smaller number of trainable parameters. Wcycle-GAN takes longer time than CNN to converge due to extra efforts on training the adversarial networks, but it still provides sufficient training efficiency as the 1,700 epochs only took ~10 hr using a single NVIDIA GeForce RTX 2080 graphic card.

For the hyperparameters selection of training the Wcycle-GAN, we set GP weight $\lambda = 100$ as suggest by Cai et al. (2020) to ensure numerical stabilities. The weighting factor λ_1 and λ_2 and the training batch size are determined in a trial-and-error manner according to the final E_{RMS} value at the convergent state. In this case, we set $\lambda_1 = 5$ and $\lambda_2 = 3$. The training batch size is 160 for the labeled data and 80 for the unlabeled data. For the extended discussion in Section 5 where we train the Wcycle-GAN but without unlabeled data, the weighting factor λ_2 is changed to 10 to reach the final E_{RMS} level of 0.06 km/s. Adam (Kingma & Ba, 2014) method with a learning rate of 5×10^{-5} and other parameters as default is used for minimizing loss functions.

Using the trained G_m, for both CNN and Wcycle-GAN, it only takes ~30 s to generate the Vs profiles from the 4,076 pairs of real group and phase velocity dispersion curves, demonstrating their prediction efficiencies. To evaluate the final Vs models obtained from different methods, we compute the χ misfit between the predicted data and the observed dispersion data at each grid cell:

$$\chi = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left[\frac{d_i^{\mathrm{pred}} - d_i^{\mathrm{obs}}}{\sigma_i^{\mathrm{obs}}} \right]^2}, \tag{8}$$

where $N = 17 \times 2$ is the dimension of dispersion data, d_i^{pred} and d_i^{obs} are the theoretical and observed dispersion wave speed (i.e., phase and group velocities) at the ith data point, and σ_i^{obs} is the corresponding data uncertainty. A good data fitting is achieved when the normalized χ^2 misfit is close to 1 (Bevington, 1969; Zelt et al., 2003). The predicted data are calculated using the same Vp-Vs-density relation as described in Section 2.2. The χ misfit is a quantitative measurement of the output Vs models quality, but it might be affected by the relation of geophysical

parameters used. Therefore, the comparison of output Vs models with respect to the result of Qiu et al. (2019) and geological information in the study area, is involved for a comprehensive analysis of network performances.

4. Results

The advantages of the proposed Wcycle-GAN method are demonstrated using surface wave dispersion data obtained from the SC plate boundary region. We first present the 3-D Vs model obtained from Wcycle-GAN method and compare it with that of Qiu et al. (2019) and the surface geology (Section 4.1). Then, models derived from different ML algorithms (e.g., CNN) are compared to illustrate the advantages of incorporating unlabeled data into the network training process (Section 4.2).

4.1. Output 3-D Vs Model

We apply the trained generative network in Wcycle-GAN to the observed dispersion data and generate the final 3-D Vs model by assembling all the 1-D Vs predictions. Because of the limited period range (i.e., 3–16 s) of the input Rayleigh wave dispersion curves, the Vs model resolved beyond the 3–20 km depth range are not well constrained (Qiu et al., 2019). Therefore, we only focus on the Vs models at depths of 3–15 km. Depth slices at the depth of 5 and 10 km for the initial model (CVM-H) and differences between the initial and final models are presented in Figure S6 in Supporting Information S1. The largest differences between our final model and the CVM-H are found underneath the basins and near the Salton Trough in the top 3–10 km, consistent with that in Qiu et al. (2019).

Figure 5 shows the depth slices of the Vs model resolved at 5 and 10 km from various methods (Figure S7 in Supporting Information S1 for depth slices at 3 and 15 km). At shallow depths (e.g., in the top 3–7 km; Figure 5c and Figures S7a–S7b in Supporting Information S1), we can clearly see a good agreement between our final model (Figures 5c and 5d) and the surface geology, such as low-velocity anomalies at Southern Central valley, LA Basin, Ventura Basin, and the Salton Trough; areas with high velocity in the Peninsular Ranges (e.g., Berg et al., 2018; Lee et al., 2014; Tape et al., 2010). It is important to note that our model shows the low-velocity zone better within the junction between the San Jacinto Fault (SJF) and San Andreas Fault (SAF) compared to the CVM-H (Figures S6a–S6b in Supporting Information S1).

At greater depths (e.g., below 10 km; Figure 5d and Figures S7c–S7d in Supporting Information S1), a sharp velocity contrast from west to east in the Peninsular Ranges is observed, which is related to the Hemet stepover (Marliyani et al., 2013). Clearer velocity contrasts across major fault systems, such as Elsinore Fault (EF), SJF and SAF are depicted in the map views of the final Vs model (Figure 5d and Figures S7c–S7d in Supporting Information S1), suggesting the derived Vs model yields higher resolutions compared to the CVM-H. These observations agree well with the large-scale features found in the Vs model of Qiu et al. (2019). In addition, the differences between the two models at different depth slices, which are shown in Figures S8–S10 in Supporting Information S1, are rather small. The consistent observation of largest velocity updates beneath basin, coherent large-scale velocity structures, together with small model differences suggest a cross-validation of both the Wcycle-GAN and the Eikonal tomography model.

Unlike the conventional linearized Vs inversion (e.g., Qiu et al., 2019), in which an extra spatial filtering is applied to achieve a smoothed 3-D Vs model, our final Vs model is illustrated without any additional spatial smoothing. Both the horizontal and vertical cross-sections of the output model are still in general continuous laterally (Figures 5c, 5d, and 8), suggesting that the Wcycle-GAN method inherently guarantees a spatial smoothness that is similar to those of the surface wave velocity dispersion maps (Figures S1–S2 in Supporting Information S1). The proposed Wcycle-GAN method shows potential to improve lateral consistency of the neighboring 1-D models, which is often difficult to achieve from conventional 1-D Vs inversion methods. While the map view of Vs model correlates well with the large-scale features observed in surface geology, we also illustrate in Section 5 that our output model illuminates well the structure of major fault systems (e.g., width of low-velocity zone and dipping fault plane) from depth cross-sections (blue lines in Figure 2).

Figure 6 shows histograms of the χ misfit of the dispersion data computed following Equation 8 for Vs models obtained from different methods. In the misfit calculation, the compressional velocity (Vp) model by assuming

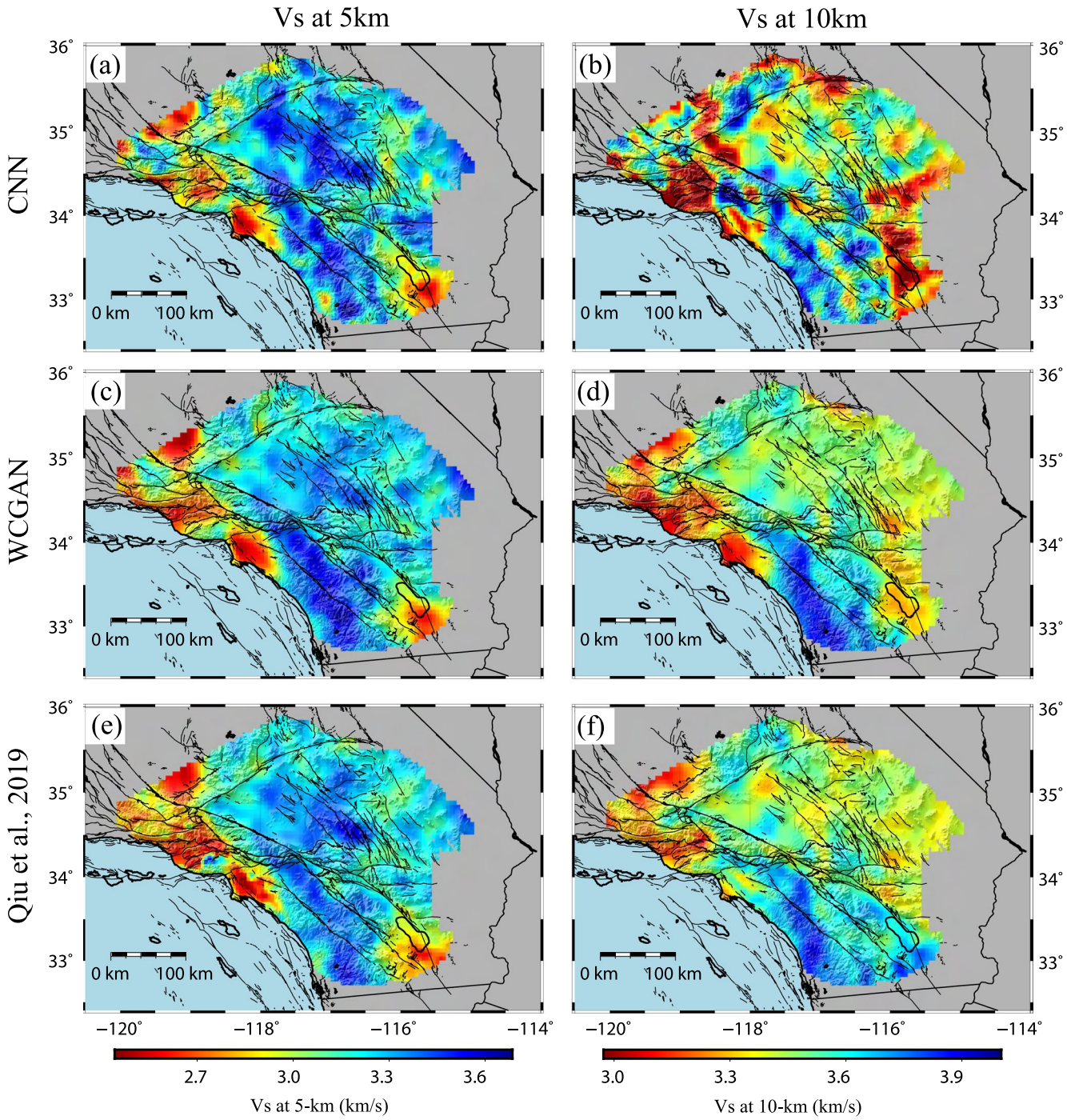

Figure 5. Comparison of depth slices for 3-D Vs models obtained from three different methods. Depth slices at 5 km (left column) and 10 km (right column) for (a–b) CNN-based model, (c–d) the proposed Wcycle-GAN based model, and (e–f) the Eikonal tomography model from Qiu et al. (2019), respectively. Black lines delineate the coastline and light gray lines depict the surface traces of the major faults in Southern California.

the same Vp/Vs ratio as the CVM-H and the density model same as the CVM-H are used. Map views of χ misfits are depicted in Figure S11 in Supporting Information S1. The misfit values are generally lower for the Wcycle-GAN model than those of the Vs model of Qiu et al. (2019) in the Salton Trough region, indicating our final Vs model is likely more accurate (i.e., closer to the global minimum) in the area. The average misfit of the Wcycle-GAN based model (0.949) is close to 1, suggesting the final Vs model is of good fit to and not overall overfitting the input dispersion data. Although the average misfit value of our model is a bit higher than that

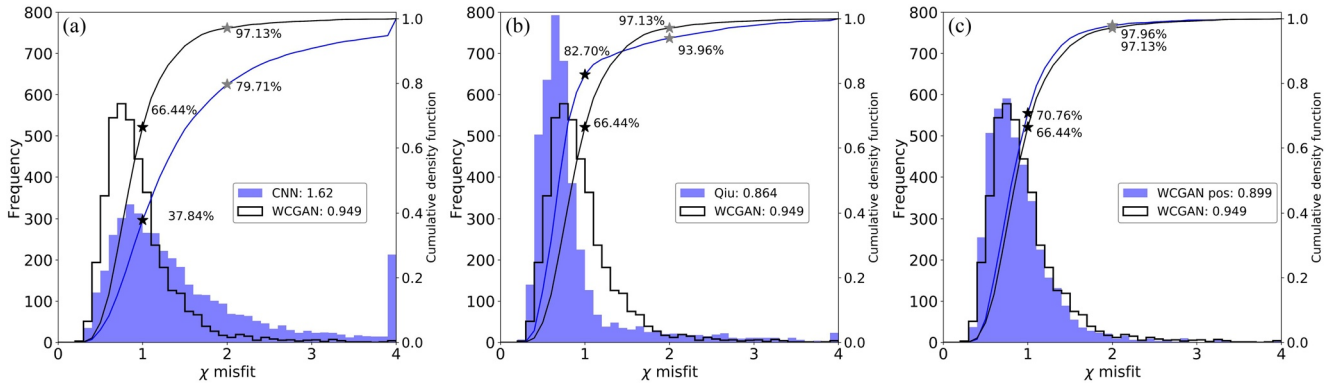

Figure 6. Probability (in blue) and cumulative (blue curve) density distributions for χ misfit. (a) χ misfit values computed for the final Vs model of CNN following Equation 7 overall available grid cells. (b) Same as (a) for the final Vs model of Qiu et al. (2019). (c) Same as (a) for final Vs model derived from Wcycle-GAN with position information incorporated in training. The χ misfit values of the final model output from Wcycle-GAN are set as the background with black outlines and curves representing probability and cumulative density distributions, respectively. The corresponding spatial distribution of the χ misfit values are depicted in Figure S11 in Supporting Information S1. The black and gray stars denote the percentage of grid cells with χ misfit less than 1 and 2, respectively. The χ misfit values larger than 4 are set to 4.

(0.864; Figure 6b) of Qiu et al. (2019), we note that a lower average misfit value below 1 may suggest an overall overfitting of the input data.

4.2. Comparison With the Conventional CNN Algorithm

In this section, we compare the Vs model from the Wcycle-GAN method with that from the conventional CNN algorithm. To exclude the performance differences brought by network structures, the same structure of model generative network (G_m) is used in CNN as that in Wcycle-GAN (Figure 4c). Since G_m is the final product for Vs prediction in both methods, the network performance is then mainly dependent on the algorithm difference between CNN and Wcycle-GAN and usage of unlabeled data during training. The training hyperparameters (e.g., batch size, learning rate) and stopping criteria are illustrated in Section 3.4.

Figures 5a and 5b present depth slices of the Vs model derived from the CNN method at 5 and 10 km, respectively, whereas the data misfit histogram is shown in Figure 6a. Compared to results from the proposed Wcycle-GAN method (Figures 5c, 5d and 6a), the Vs model from CNN is spatially less smooth and continuous, and has much higher average misfit values, suggesting results from the CNN method are less reliable. This is likely due to the limited diversity provided in the labeled data set generated synthetically. In addition, the Wasserstein metric used in the Wcycle-GAN improves the long-wavelength features recovery, resulting in improved training stability and thereby enhanced spatial smoothness of the output 3-D Vs model. Similar property of Wasserstein metric has been observed in near-surface seismic velocity estimation using full-waveform inversion (Yang et al., 2018). Statistically, the Wcycle-GAN based model is more coherent to the physics-driven inverted model (i.e., Qiu et al., 2019) in terms of significantly smaller standard deviation of model differences (Figures S12–S13 in Supporting Information S1) comparing to the model from CNN. From the histogram (Figure S13 in Supporting Information S1), it is also noted that the Wcycle-GAN based model is slightly slower than the physics-driven model at the depth of 10 km, which is due to a lower velocity predicted at the LA basin (Figures 5d and 5f). The better accuracy in fitting the observed dispersion data and spatial continuity of the Vs model from the Wcycle-GAN method demonstrates the effectiveness of the proposed method by incorporating advanced loss function, cycle consistency, and unlabeled data into the training process.

5. Discussions

The proposed Wcycle-GAN method has the potential to compensate for the limited diversity in synthetic dispersion data owing to the addition of real data in training. The improvements come from two main factors. On one hand, the adversarial loss forces the generated Vs models to fall into the same probability distribution as the real Vs models samples, which relieves prediction outliers when applying to real data. On the other hand, the

cycle consistency guarantees bijection mappings between the input dispersion data and corresponding Vs model predictions. The derived Vs model from Wcycle-GAN can reconstruct the input observed dispersion data in an approximate manner of the physical forward modeling learned by data generative networks (G_d). This constraint is typically missing in the conventional deep learning methods.

The Wasserstein adversarial loss provides improved training stability and convergence characteristic comparing with cross-entropy or least squares for Cycle-GAN based Vs inversion. Figure S14 in Supporting Information S1 shows the comparative study of using different metrics for adversarial loss. The results suggest using least squares loss can result in underfitting to the labeled data as the incorrect prediction of the velocity jump at Moho depth. Both cross-entropy and least squares adversarial loss can result in strong artifacts and negative velocity gradient in the Vs predictions using unlabeled data. In comparison, Wasserstein loss results in high model prediction quality using either labeled or unlabeled data. For the weighting factors in the loss functions, we note changes in hyperparameter λ_1 and λ_2 has relatively small effects on the final derived Vs model, but a future study of their effects would be beneficial for the best use of the Wcycle-GAN method.

The proposed method has the potential to be efficiently applied to a new study area through transfer learning. Two situations are considered: (a) when the application is characterized by inverting similar dispersion data with respect to the training data set, we can directly apply the previous trained neural network for prediction, and (b) we can also incorporate the new data into the unlabeled data set and start training from the previous best trained network, which usually converges fast in a small number of iterations (<100 epochs). Then the new trained network is expected to generate improved Vs model predictions. We note that, if the labeled and unlabeled data do clearly show different distributions, additional actions, such as careful selection of labeled data (e.g., a mixture of existing training data set and ones generated from local Vs models), are needed to improve the prediction quality.

Three additional experiments are conducted to further demonstrate the key features of the proposed Wcycle-GAN methods: the importance of integrating unlabeled data into training (Section 5.1), the potential to obtain stable and consistent results with much smaller labeled data set (Section 5.2), and the ability of this method to include additional information (i.e., location of the grid cell) to further improve the final 3-D Vs model (Section 5.3). At last, a rough estimation of model uncertainty is presented in Section 5.4.

5.1. Importance of Incorporating Real Data Into the Training Data Set

Here, we perform an experiment, in which the same Wcycle-GAN structure is used but trained without the unlabeled data. Figure S15 in Supporting Information S1 shows the map view of the output Vs model at different depths. Artificial and abrupt lateral heterogeneities are seen in the model, compared to Figures 5c and 5d. This instability in predictions is consistent with the larger variance of this output model in training, as shown in Figures S12–S13 in Supporting Information S1. Also, training without unlabeled data results in larger data misfits (~2.3 in average) as clearly seen both in histogram (Figure S16a in Supporting Information S1) and map view (Figure S11b in Supporting Information S1). Overall, this comparison strongly suggests that the addition of the unlabeled data into the training process is mainly responsible for the observed improvement in the prediction of reliable Vs model using Wcycle-GAN. The unlabeled data together with the adversarial training mechanism and Cycle-GAN structure promote the prediction quality of the Wcycle-GAN, makes the Wcycle-GAN a promising and powerful ML-based Vs inversion method.

5.2. Downsampling of Labeled Data

We stated that the proposed Wcycle-GAN method could still work well with small labeled data set that is challenging to train the conventional CNN properly. To demonstrate this, we reduce the amount of labeled data by down sampling the CVM-H15.1 with a grid spacing of $0.1° \times 0.1°$ (originally $0.03° \times 0.03°$). This results in a selection of 1890 (originally 16,480) labeled data, which is even less than half of the number (4,076) of observed dispersion curves. Figures 7a and 7b show the depth slices of the Vs model from the Wcycle-GAN method trained with down sampled labeled data set. The resulting Vs model is consistent with that trained using the full labeled data sets. Figure S16b in Supporting Information S1 shows the data misfit of the Vs model from the network trained with reduced labeled data set. Only a small increase in the mean misfit, that is, from 0.949 to 1.10, compared to that of results trained with the full labeled data set is observed. It is important to note that the average misfit value 1.1 is still much smaller than those of the supervised methods (Figure 6a). The result

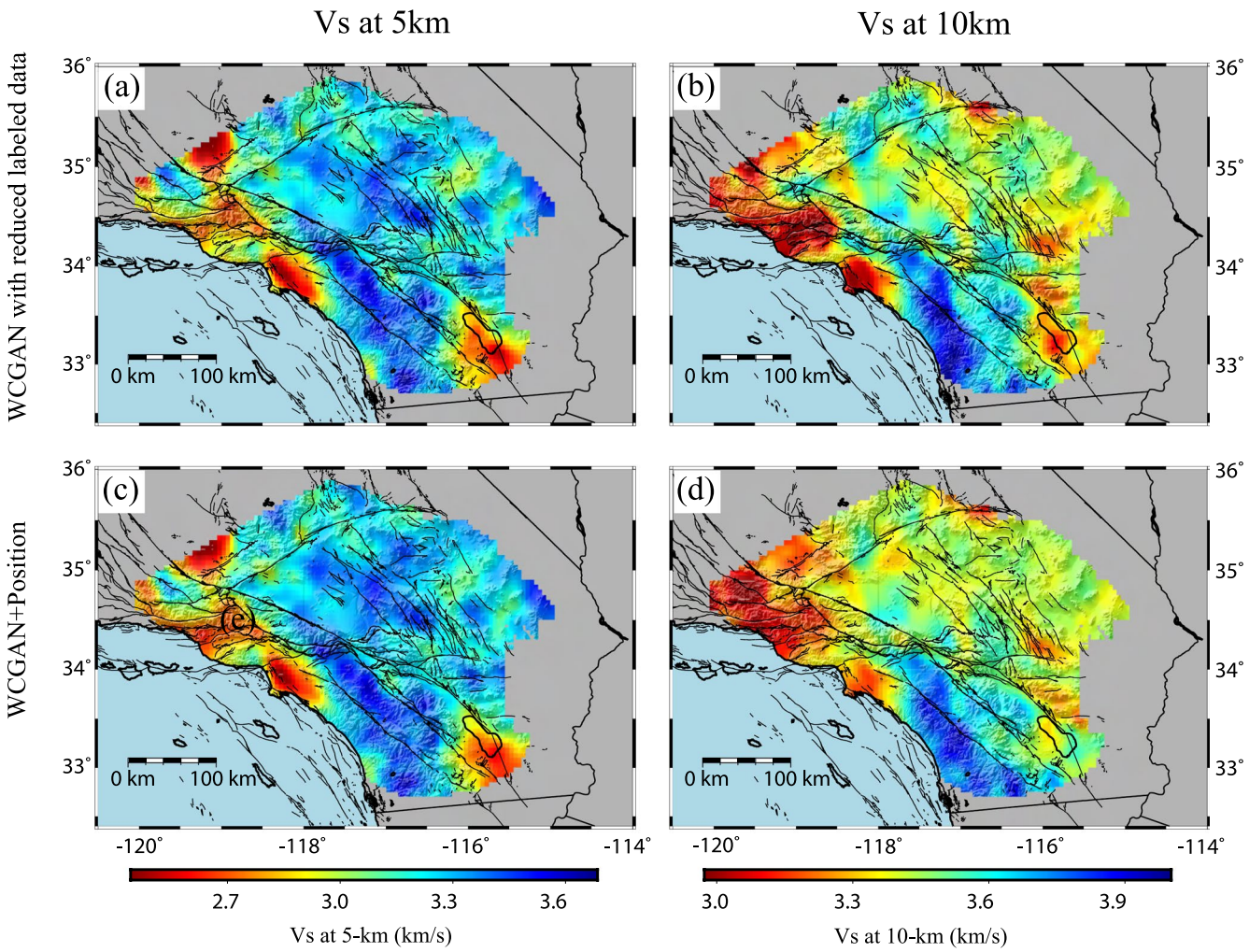

Figure 7. Depth slices of shear velocity model at 5 km (left column) and 10 km (right column) for (a–b) Wcycle-GAN (WCGAN) based model but using down sampled 1,890 labeled data and (c–d) Wcycle-GAN based model with location information added as extra prior information in the network training (WCGAN + Position).

suggests the redundancy in the full labeled data set and further demonstrates the strength of the proposed Wcycle-GAN method in resolving high accuracy Vs model even when the labeled data set is small. It only takes ~4 hr to train the network using the reduced labeled data set.

5.3. Wcycle-GAN With Position Information

An extension to the proposed Wcycle-GAN algorithm is incorporating additional information, such as the location (i.e., longitude and latitude) in the training process, which can further enhance the accuracy in the application of Vs inversion. Map views of the Vs model, derived from the proposed method with the latitude and longitude of both the labeled and unlabeled data incorporated into the training process (hereinafter referred to as "Wcycle-GAN + Position model"), at 5 and 10 km are presented in Figures 7c and 7d, respectively. Details of how to incorporate location information into an ML network training can be found in Text S4 in Supporting Information S1. The Vs models resolved from networks trained with and without the input of location information are nearly identical to each other at a large scale (e.g., tens of kilometers; Figures 5c, 5d, 7c and 7d). Interestingly, incorporating location information further enhances spatial smoothness in the vertical cross-sections (Figures S17–S19 in Supporting Information S1). The data misfits (~0.9 in Figure 6c) are slightly smaller for the Wcycle-GAN + Position model. Therefore, we show the cross-sections of the Wcycle-GAN + Position model in Figure 8 to infer structures of the major fault systems. We note that the incorporation of location information for both the

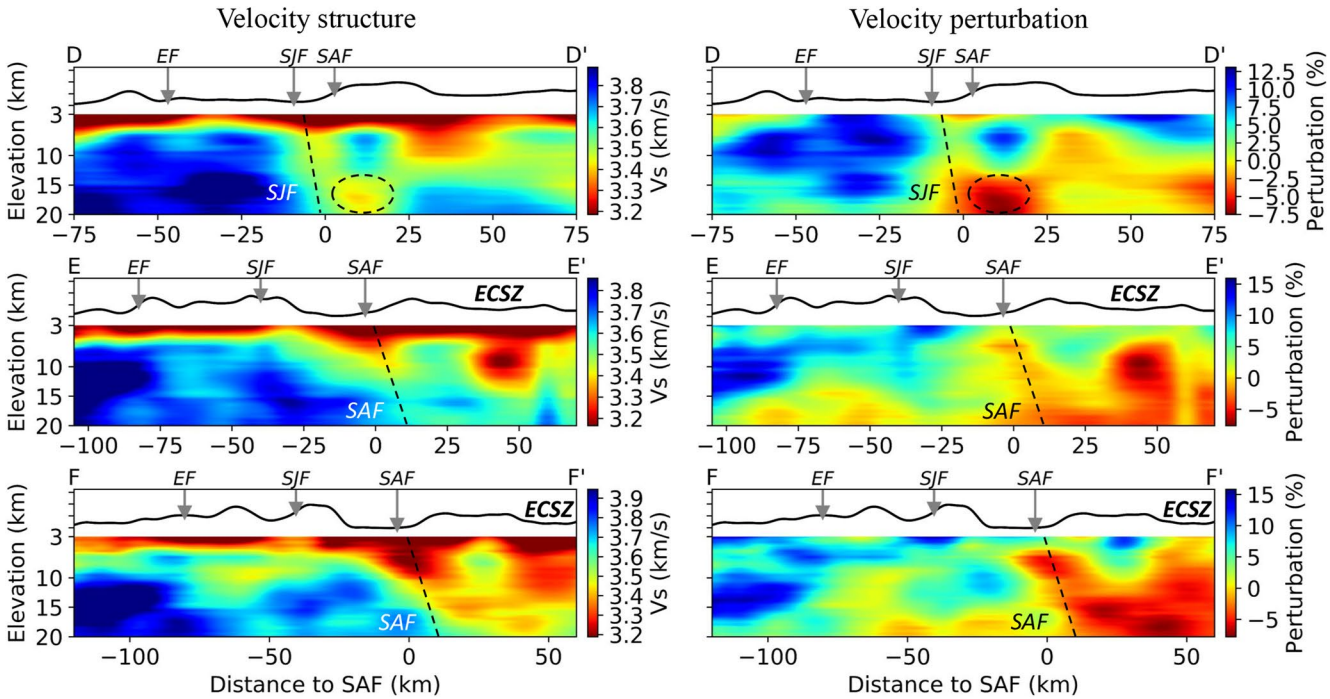

Figure 8. Cross-sections (blue lines in Figure 2) of the Vs model resolved from the Wcycle-GAN network with location information incorporated in training. Left panels show the velocity model, whereas perturbations relative to the 1-D Vs profile averaged along the cross-section are illustrated on the right. The black curve depicts an exaggerated topography variation. The black dashed line in each profile depicts the inferred fault planes for SJF in DD' and SAF in EE' and FF'. The dashed ellipse in DD' outlines a low-velocity anomaly that is likely associated with rock damaged inferred in Ben-Zion and Zaliapin (2019). EF, Elsinore Fault; SJF, San Jacinto Fault; ECSZ, Eastern California Shear Zone.

labeled and unlabeled data could have a greater impact on the result when applying to the Vs inversion at regional or global scales.

We show the cross-sections DD', EE', and FF' (blue lines in Figure 2), the same as those shown in Figure 1 of Qiu et al. (2019), of the final Vs model between 3 and 20 km to infer the structures of EF, SJF, and SAF at depth. In the profile DD', the low-velocity zone indicates both the SJF and SAF are nearly vertical. This is consistent with the fault geometry near San Gorgonio Pass (SGP) from the Community Fault Model in SC (CFMv5; Plesch et al., 2007). Besides, we observe a pronounced low-velocity body (dashed circle, Figure 8) between depths of 15–20 km, which is consistent with the results of Qiu et al. (2019 Figure S17c in Supporting Information S1). This low-velocity anomaly at great depth, with ~5%–7% lower velocities compared to the surrounding media, is likely related to the large damage volume beneath the SGP estimated in Ben-Zion and Zaliapin (2019).

In profile EE', we observe a broad (~5-km-wide) flower-shaped (i.e., width decreases with depth) fault damage zone with ~2%–3% average velocity reduction for the SAF in the top 8–10 km that is clearly dipping toward the northeast. The estimated dipping angle of SAF in profile EE' is ~60°. This dipping angle is consistent with the observation in Qiu et al. (2019), but the flower-shaped fault damage zone is less clear in their results (Figure S18c in Supporting Information S1). Besides, the low-velocity anomaly beneath the Eastern California Shear Zone is slightly deeper than that in Qiu et al. (2019). Similarly, the SAF is highlighted by a flower-shaped low-velocity zone that is dipping toward the northeast with a similar angle (~60°) in the top 10 km. Different from EE', the low-velocity zone is more pronounced (~4%–5%) in FF', likely indicating the rocks inside the fault zone are more damaged in the southwest.

The flower-shaped fault zone structures in EE' and FF' are consistent with the model of Fuis et al. (2017) derived for the southern section of the SAF by jointly inverting gravity and magnetic data. In addition, the observed ~60° dipping angle in both EE' and FF' agrees well with the previous estimation from magnetic data (~65°; Fuis et al., 2012). It is important to note that the model of Qiu et al. (2019 Figures S18c and S19c in Supporting Information S1) is subject to the choice of damping parameter in the inversion and the spatial smoothing

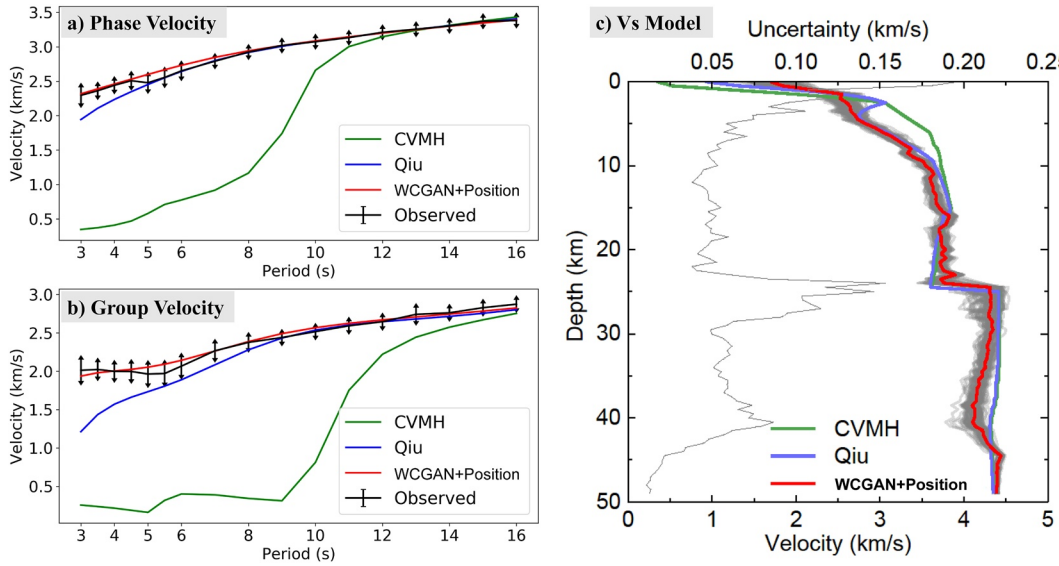

Figure 9. Comparison of the observed and predicted Rayleigh-wave (a) phase and (b) group velocities for models from Wcycle-GAN + Position, Qiu et al. (2019), and CVM-H at an example grid cell (116°W, 33°N) near Salton Trough. The black curve denotes the observed dispersion curve and black arrows at every period point represents the corresponding data uncertainties. The 1-D Vs models are shown in (c), and the dark gray area illustrates the 100 Vs profiles generated by the bootstrap tests described in Section 5. The light gray line in (c) denotes the uncertainties of Vs profile calculated as one standard deviation estimated through the bootstrap process.

afterward. Therefore, through the depth cross-section comparison, we again demonstrate the robustness of our Vs model from the Wcycle-GAN model and confirm with a different method that the flower-shaped damage zone and fault dipping toward northeast observed for the southern section of the SAF in Qiu et al. (2019) are reliable. These features have important implications, such as a better understanding of strong ground motions produced by earthquakes that will occur on the SAF.

5.4. Model Uncertainty Estimation

Model uncertainty estimation associated with DNNs methods such as CNN and Wcycle-GAN is challenging. Mixture density networks (MDNs; Earp et al., 2020; Meier et al., 2007) and INNs (Zhang & Curtis, 2021) can estimate full posterior probability distribution functions (PDFs) of the output Vs model, by defining the probability distribution as a sum of assumed analytic PDFs or by expanding the data vector with a latent variable of chosen probability distribution (e.g., Gaussian distribution) that can be projected to model PDFs, respectively.

As an approximation, we estimate the data-quality based uncertainty of Wcycle-GAN method through bootstrap tests described as follows: (a) for each grid cell, we first resample the observed dispersion data 100 times by adding perturbations (δv) to phase and group dispersion curves according to associate uncertainties at each period. Perturbations are randomly generated following the uniform probability distribution $\delta v \sim U(-\sigma_i^{\text{obs}}, \sigma_i^{\text{obs}})$ at the scale of corresponding data uncertainty σ_i^{obs}; (b) the perturbed data set, consisting of 100 dispersion curves, is then feed into the already trained generative networks to predict 1-D Vs profiles; (c) the model uncertainty is estimated as the standard deviation of the 100 Vs model predictions using this perturbed data set. We note that this is only a rough estimation of how data errors propagated into the predicted Vs model through the networks.

Figure 9 presents an example of such uncertainty estimation near Salton Trough (116°W, 33°N). The Wcycle-GAN + Position model is used in uncertainty estimation since it is the preferred final model as discussed in Section 5.3. Compared to the starting CVM-H model, the predicted dispersion curves given by the Wcycle-GAN derived model yields significantly improved fitting to the observed ones (Figures 9a–9b) that are within data uncertainties. The dispersion curves from bootstrap results at the example grid cell are depicted as dark gray lines (Figure 9c). The estimated model uncertainties (light gray curve in Figure 9c) are generally larger in the top 3 km and near the Moho depth at the depth around 23 km. The large uncertainties at these depth ranges are due to large

errors in short period data and lack of long period (>16 s) data (Qiu et al., 2019). The same analysis is performed at a different grid location (117°W, 34°N) and presented in Figure S20 in Supporting Information S1 for a direct comparison to Figure 12 in Qiu et al. (2019). We also estimate the model uncertainty for the CNN result (Figure S21 in Supporting Information S1). The CNN derived model shows much larger uncertainties than that of the proposed method. It is important to note that, even compared to the result of Hu et al. (2020), where they mainly focused on optimizing the CNN to solve 1-D Vs inversion and thus should provide the best possible performance from CNN, the Wcycle-GAN still outperforms CNN, such as better recovery the velocity jump at the Moho depth at ~25 km (Figure 9) and smaller uncertainty values. Investigation of the optimum Wcycle-GAN structure (e.g., using 1-D or 2-D generative networks) will be a subject of future study.

Another source of model uncertainty comes from possibly different trained neural networks retrieve from different stopping point in training. To investigate this contribution, we extract 11 trained neural networks, that is, one per 10 epochs from 50 epochs before to 50 epochs after the training stop point. Figure S22 in Supporting Information S1 shows 1-D Vs profiles generated from the same input but through these 11 different trained networks. The standard deviations of the output Vs profiles, unlike the data-quality-based model uncertainty, are generally much smaller (below 0.05 km/s) for all depths (Figure 9c). This suggests that uncertainties from network training are negligible compared to contribution from uncertainties of the input data, which advocates Wcycle-GAN as a stable method for Vs inversion problem.

The focus of this article is on the incorporation of real dispersion data into training process to improve Vs prediction quality, but it is possible to further improve Wcycle-GAN in utilizing uncertainty information. For incorporating data uncertainties, we can weight the cycle-consistent penalty function according to uncertainties of the dispersion data (σ_i^{obs}; Figures S1–S2 in Supporting Information S1). For synthetic dispersion data, the weight is set to root-mean-square of the uncertainties of all observed data at a certain period. For estimating the full PDFs of the output Vs model, we may apply the same strategy as Zhang and Curtis (2021) that utilizes a latent variable during the training of Wcycle-GAN.

6. Conclusions

Different from previous studies, we develop the first ML based Vs inversion method that incorporates also unlabeled data in the training process, which has the potential to be applied to data collected from regions that are poorly constrained prior to the inversion. The proposed method shows an improved prediction quality, better training stability, and only requires a small amount of labeled data, compared to CNN-based method. We demonstrate these improvements by using the fundamental mode Rayleigh wave velocity dispersion data derived in the SC plate boundary region. The final Vs model obtained from the proposed method show clearer images of structures near faults in the top 15 km, specifically the low-velocity damage zone centered on the southern section of the San Andreas Fault that is dipping ~60° to the northeast. In addition, integrating longitude and latitude information into the Wcycle-GAN algorithm further improves the prediction quality as well as the spatial continuity of the final Vs model. For future studies, we plan to investigate the potential of this method by reducing the amount of labeled data through leveraging random sampling or sampling strategy based on clustering analysis (Eymold & Jordan, 2019). In addition, how to incorporate data uncertainty and estimate model uncertainty using Wcycle-GAN will also be included in the future development of the proposed method.

Data Availability Statement

The Rayleigh wave velocity dispersion data used in this study are derived in Qiu et al. (2019) and accessible at https://doi.org/10.17632/dt9x54dtrr.1. Codes and benchmark examples are accessible at https://github.com/aocai166/Wasserstein-Cycle-GAN-for-Surface-Wave-Tomography.

Acknowledgments
The labeled data set is extracted from the Southern California Earthquake Center (SCEC) Community Velocity Model of Shaw et al. (2015; CVMH). The Wcycle-GAN is implemented using the deep-learning framework of TensorFlow. The training and prediction processes are conducted using a single NVIDIA GeForce RTX 2080 GPU with a memory of 8 GB. Fruitful discussions with Dr. Jing Hu at University of Science and Technology of China (USTC) are well appreciated. The authors thank the Editor Dr. Michael Bostock, an anonymous Associate Editor, and three anonymous reviewers for their constructive and thoughtful comments and suggestions, which have significantly improved the quality of this article. A.C. is supported by Rice University, and F.N. is partially supported by the National Natural Science Foundation of China (41630209).

References

Adler, A., Araya-Polo, M., & Poggio, T. (2019). Deep recurrent architectures for seismic tomography. *Paper presented at 81st EAGE Conference and Exhibition* (Vol. *2019*, No. 1, pp. 1–5). European Association of Geoscientists & Engineers. https://doi.org/10.3997/2214-4609.201901512

Araya-Polo, M., Adler, A., Farris, S., & Jennings, J. (2020). Fast and accurate seismic tomography via deep learning. In *Deep learning: Algorithms and applications* (pp. 129–156). Cham: Springer. https://doi.org/10.1007/978-3-030-31760-7_5

Araya-Polo, M., Jennings, J., Adler, A., & Dahlke, T. (2018). Deep-learning tomography. *The Leading Edge*, *37*(1), 58–66. https://doi.org/10.1190/tle37010058.1

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., et al. (2018). *Analyzing inverse problems with invertible neural networks*. Retrieved from http://arxiv.org/abs/1808.04730

Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *Paper presented at 5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings*. Retrieved from https://arxiv.org/abs/1701.04862v1

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN*. Retrieved from http://arxiv.org/abs/1701.07875

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). *Layer normalization*. Retrieved from http://arxiv.org/abs/1607.06450

Ben-Zion, Y., & Zaliapin, I. (2019). Spatial variations of rock damage production by earthquakes in Southern California. *Earth and Planetary Science Letters*, *512*, 184–193. https://doi.org/10.1016/J.EPSL.2019.02.006

Berg, E. M., Lin, F. C., Allam, A., Qiu, H., Shen, W., & Ben-Zion, Y. (2018). Tomography of Southern California via Bayesian joint inversion of Rayleigh wave ellipticity and phase velocity from ambient noise cross-correlations. *Journal of Geophysical Research: Solid Earth*, *123*(11), 9933–9949. https://doi.org/10.1029/2018JB016269

Bergen, K. J., Johnson, P. A., de Hoop, M. V., & Beroza, G. C. (2019). Machine learning for data-driven discovery in solid Earth geoscience. *Science*, *363*, eaau0323. https://doi.org/10.1126/science.aau0323

Bevington, P. R. (1969). *Data reduction and error analysis for the physical sciences*. New York: McGraw-Hill. Retrieved from https://ui.adsabs.harvard.edu/abs/1969drea.book.B/abstract

Bianco, M. J., & Gerstoft, P. (2018). Travel time tomography with adaptive dictionaries. *IEEE Transactions on Computational Imaging*, *4*(4), 499–511. https://doi.org/10.1109/tci.2018.2862644

Bianco, M. J., Gerstoft, P., Olsen, K. B., & Lin, F. C. (2019). High-resolution seismic tomography of Long Beach, CA using machine learning. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-019-50381-z

Cai, A., Di, H., Li, Z., Maniar, H., & Abubakar, A. (2020). *Wasserstein cycle-consistent generative adversarial network for improved seismic impedance inversion: Example on 3-D SEAM model. Wasserstein cycle-consistent generative adversarial network for improved seismic impedance inversion: Example on 3-D SEAM model* (pp. 1274–1278). Society of Exploration Geophysicists. https://doi.org/10.1190/segam2020-3425785.1

Cheng, X., Liu, Q., Li, P., & Liu, Y. (2019). Inverting Rayleigh surface wave velocities for crustal thickness in eastern Tibet and the western Yangtze craton based on deep learning neural networks. *Nonlinear Processes in Geophysics*, *26*(2), 61–71. https://doi.org/10.5194/npg-26-61-2019

Das, V., Pollack, A., Wollner, U., & Mukerji, T. (2019). Convolutional neural network for seismic impedance inversion. *Geophysics*, *84*(6), R869–R880. https://doi.org/10.1190/geo2018-0838.1

Devilee, R. J. R., Curtis, A., & Roy-Chowdhury, K. (1999). An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness. *Journal of Geophysical Research: Solid Earth*, *104*(B12), 28841–28857. https://doi.org/10.1029/1999jb900273

Earp, S., Curtis, A., Zhang, X., & Hansteen, F. (2020). Probabilistic neural network tomography across Grane field (North Sea) from surface wave dispersion data. *Geophysical Journal International*, *223*(3), 1741–1757. https://doi.org/10.1093/gji/ggaa328

Eymold, W. K., & Jordan, T. H. (2019). Tectonic regionalization of the Southern California crust from tomographic cluster analysis. *Journal of Geophysical Research: Solid Earth*, *124*(11), 11840–11865. https://doi.org/10.1029/2019JB018423

Fang, H., Zhang, H., Yao, H., Allam, A., Zigone, D., Ben-Zion, Y., et al. (2016). A new algorithm for three-dimensional joint inversion of body wave and surface wave data and its application to the Southern California plate boundary region. *Journal of Geophysical Research: Solid Earth*, *121*(5), 3557–3569. https://doi.org/10.1002/2015JB012702

Fuis, G. S., Bauer, K., Goldman, M. R., Ryberg, T., Langenheim, V. E., Scheirer, D. S., et al. (2017). Subsurface geometry of the San Andreas Fault in Southern California: Results from the Salton Seismic Imaging Project (SSIP) and strong ground motion expectations. *Bulletin of the Seismological Society of America*, *107*(4), 1642–1662. https://doi.org/10.1785/0120160309

Fuis, G. S., Scheirer, D. S., Langenheim, V. E., & Kohler, M. D. (2012). A new perspective on the geometry of the San Andreas Fault in Southern California and its relationship to lithospheric structure. *Bulletin of the Seismological Society of America*, *102*(1), 236–251. https://doi.org/10.1785/0120110041

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). *Generative adversarial networks*. Retrieved from http://arxiv.org/abs/1406.2661

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). *Improved training of Wasserstein GANs*. Retrieved from http://arxiv.org/abs/1704.00028

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*. Retrieved from https://arxiv.org/abs/1502.01852

Herrmann, R. B. (2013). Computer programs in seismology: An evolving tool for instruction and research. *Seismological Research Letters*, *84*(6), 1081–1088. https://doi.org/10.1785/0220110096

Hu, J., Qiu, H., Zhang, H., & Ben-Zion, Y. (2020). Using deep learning to derive shear-wave velocity models from surface-wave dispersion data. *Seismological Research Letters*, *91*(3), 1738–1751. https://doi.org/10.1785/0220190222

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Retrieved from http://arxiv.org/abs/1502.03167

Kaur, H., Pham, N., & Fomel, S. (2020). Seismic data interpolation using CycleGAN. In *SEG International Exposition and Annual Meeting 2019* (pp. 2202–2206). Society of Exploration Geophysicists. https://doi.org/10.1190/segam2019-3207424.1

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. Retrieved from http://arxiv.org/abs/1412.6980

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, *90*(1), 3–14. https://doi.org/10.1785/0220180259

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*. *521*, 436–444. https://doi.org/10.1038/nature14539

Lee, E. J., Chen, P., Jordan, T. H., Maechling, P. B., Denolle, M. A. M., & Beroza, G. C. (2014). Full-3-D tomography for crustal structure in Southern California based on the scattering-integral and the adjoint-wavefield methods. *Journal of Geophysical Research: Solid Earth*, *119*(8), 6421–6451. https://doi.org/10.1002/2014JB011346

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2813–2821. https://doi.org/10.1109/ICCV.2017.304

Marliyani, G. I., Rockwell, T. K., Onderdonk, N. W., & McGill, S. F. (2013). Straightening of the northern San Jacinto Fault, California, as seen in the fault-structure evolution of the San Jacinto Valley stepover. *Bulletin of the Seismological Society of America*, *103*(3), 2047–2061. https://doi.org/10.1785/0120120232

Mcdermott, M. B. A., Yan, T., Naumann, T., Hunt, N., Suresh, H., Szolovits, P., & Ghassemi, M. (2018). Semi-supervised biomedical translation with cycle Wasserstein regression GANs. *Paper presented at 32nd AAAI Conference on Artificial Intelligence*.Retrieved from www.aaai.org

Meier, U., Curtis, A., & Trampert, J. (2007). Global crustal thickness from neural network inversion of surface wave data. *Geophysical Journal International*, *169*(2), 706–722. https://doi.org/10.1111/j.1365-246X.2007.03373.x

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, *11*(1). https://doi.org/10.1038/s41467-020-17591-w

Nair, V., & Hinton, G. E. (2010). *Rectified linear units improve restricted Boltzmann machines*. Retrieved from https://www.cs.toronto.edu/~fritz/absps/reluICML.pdf

Plesch, A., Shaw, J. H., Benson, C., Bryant, W. A., Carena, S., Cooke, M., et al. (2007). Community Fault Model (CFM) for Southern California. *Bulletin of the Seismological Society of America*, *97*(6), 1793–1802. https://doi.org/10.1785/0120050211

Qiu, H., Lin, F. C., & Ben-Zion, Y. (2019). Eikonal tomography of the Southern California plate boundary region. *Journal of Geophysical Research: Solid Earth*, *124*(9), 9755–9779. https://doi.org/10.1029/2019JB017806

Ross, Z. E., & Ben-Zion, Y. (2014). Automatic picking of direct *P*, *S* seismic phases and fault zone head waves. *Geophysical Journal International*, *199*(1), 368–381. https://doi.org/10.1093/gji/ggu267

Ross, Z. E., Meier, M. A., & Hauksson, E. (2018). *P* wave arrival picking and first-motion polarity determination with deep learning. *Journal of Geophysical Research: Solid Earth*, *123*(6), 5120–5129. https://doi.org/10.1029/2017JB015251

Roy, C., & Romanowicz, B. A. (2017). On the implications of a priori constraints in transdimensional Bayesian inversion for continental lithospheric layering. *Journal of Geophysical Research: Solid Earth*, *122*(12), 10118–10131. https://doi.org/10.1002/2017JB014968

Shaw, J. H., Plesch, A., Tape, C., Suess, M. P., Jordan, T. H., Ely, G., et al. (2015). Unified structural representation of the Southern California crust and upper mantle. *Earth and Planetary Science Letters*, *415*, 1–15. https://doi.org/10.1016/j.epsl.2015.01.016

Shen, W., Ritzwoller, M. H., Schulte-Pelkum, V., & Lin, F. C. (2013). Joint inversion of surface wave dispersion and receiver functions: A Bayesianmonte-Carlo approach. *Geophysical Journal International*, *192*(2), 807–836. https://doi.org/10.1093/gji/ggs050

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. Retrieved from http://arxiv.org/abs/1409.1556

Tape, C., Liu, Q., Maggi, A., & Tromp, J. (2010). Seismic tomography of the Southern California crust based on spectral-element and adjoint methods. *Geophysical Journal International*, *180*(1), 433–462. https://doi.org/10.1111/J.1365-246X.2009.04429.X

Wang, Y., Ge, Q., Lu, W., & Yan, X. (2020). Seismic impedance inversion based on cycle-consistent generative adversarial network. In *SEG International Exposition and Annual Meeting 2019* (pp. 2498–2502). Society of Exploration Geophysicists. https://doi.org/10.1190/segam2019-3203757.1

Wathelet, M. (2008). An improved neighborhood algorithm: Parameter conditions and dynamic scaling. *Geophysical Research Letters*, *35*(9). https://doi.org/10.1029/2008GL033256

Wu, X., Liang, L., Shi, Y., & Fomel, S. (2019). FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3-D seismic fault segmentation. *Geophysics*, *84*(3), IM35–IM45. https://doi.org/10.1190/geo2018-0646.1

Xiong, N., Qiu, H., & Niu, F. (2021). Data-driven velocity model evaluation using *K*-means clustering. *Geophysical Research Letters*, *48*(23), e2021GL096040. https://doi.org/10.1029/2021gl096040

Yang, Y., Engquist, B., Sun, J., & Hamfeldt, B. F. (2018). Application of optimal transport and the quadratic Wasserstein metric to full-waveform inversion. *Geophysics*, *83*(1), R43–R62. https://doi.org/10.1190/GEO2016-0663.1

Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. *Proceedings of the IEEE International Conference on Computer Vision, 2017-October*, 2868–2876. https://doi.org/10.1109/ICCV.2017.310

Zelt, C. A., Sain, K., Naumenko, J. V., & Sawyer, D. S. (2003). Assessment of crustal velocity models using seismic refraction and reflection tomography. *Geophysical Journal International*, *153*(3), 609–626. https://doi.org/10.1046/j.1365-246X.2003.01919.x

Zhang, X., & Curtis, A. (2021). Bayesian geophysical inversion using invertible neural networks. *Journal of Geophysical Research: Solid Earth*, *126*(7). https://doi.org/10.1029/2021JB022320

Zhang, X., Jia, Z., Ross, Z. E., & Clayton, R. W. (2020). Extracting dispersion curves from ambient noise correlations using deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, *58*(12), 8932–8939. https://doi.org/10.1109/TGRS.2020.2992043

Zhong, Z., Sun, A. Y., & Wu, X. (2020). Inversion of time-lapse seismic reservoir monitoring data using CycleGAN: A deep learning-based approach for estimating dynamic reservoir property changes. *Journal of Geophysical Research: Solid Earth*, *125*(3). https://doi.org/10.1029/2019JB018408

Zhu, J.-Y., Park, T., Isola, P., Efros, A. A., & Research, B. A. (2017). *Unpaired image-to-image translation using cycle-consistent adversarial networks Monet photos*. Retrieved from https://github.com/junyanz/CycleGAN

Zhu, W., & Beroza, G. C. (2019). PhaseNet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, *216*(1), 261–273. https://doi.org/10.1093/gji/ggy423