DD-Net: Dual decoder network with curriculum learning for full waveform inversion

Xing-Yi Zhang, Fan Min, *Member, IEEE*, Shulin Pan, Qiong Xu, Xue-Yang Min, Guojie Song, Ke Wang

***Abstract*—Deep learning full waveform inversion (DL-FWI) is gaining much research interest due to its high prediction efficiency, effective exploitation of spatial correlation, and lack of the need for an initial estimate. As a data-driven approach, it has several key issues. For example, effective deep networks need to be designed, the training process needs to be controlled, and the generalization ability needs to be enhanced. In this paper, we propose a dual decoder network with curriculum learning (DD-Net) to handle these issues. First, we design a U-Net with two decoders to grasp the velocity value and stratigraphic boundary information of the velocity model. These decoders' feedback will be combined at the encoder to enhance the encoding of edge spatial information. Second, we introduce curriculum learning to network training by organizing data in three difficulty levels. The easy-to-hard training process enhances the data fitting of the network. Third, we apply the network to low resolution seismic observations via a pre-network dimension reducer. This can serve as a general design idea without destroying the original network characteristics. Experiments are undertaken on SEG salt datasets and four synthetic datasets from OpenFWI. The results show that our network is superior to other state-of-the-art data-driven networks. The source code is available at github.com/fansmale/ddnet.**

***Index Terms*—full waveform inversion; neural network; parallel decoder; seismic images.**

I. Introduction

DEEP learning full waveform inversion (DL-FWI) has shown a number of advantages over classical physics-based FWI (NS-FWI, also known as physics-driven FWI). First, although the training stage is very time and resource consuming, end-to-end DL-FWI is quite efficient in the network prediction stage [1]. In contrast, NS-FWI requires enormous computational costs to solve wave equation in high dimensions [2]. Second, DL-FWI does not require any initial estimation model, which plays a fundamental role in NS-FWI [3]. In fact, it often provides NS-FWI with an initial model for further optimization [4]. Third, in traditional FWI, the regularity of seismic exploration sampling is rarely utilized. However, deep learning can be used to explore this potential relationship through network training [5].

Network design is one of the key issues for any deep learning approach. To adapt to the domain knowledge of seismic data, two types of architectures are commonly used in DL-FWI. One type is embedding, which maps seismic waveforms to higher dimensions by feature conversion [6]. Based on the high-dimensional spatial alignment information, the velocity model is reconstructed by a DNN [7] or CNN [8]. In some strategies [9, 10], this spatial alignment information is combined with the position information of shot points and geophones. The other type is the encoder-decoder, which constructs end-to-end mapping through full convolution [11, 12]. Some approaches preserve spatial dimension information in the final encoding [1]. In some methods, a CRF [13] is added at the end of the decoder to ensure spatial association [14].

Training process control has a significant impact on applications related to field data. Transfer learning [15] is popular in training DL-FWI networks. Both FCNVMB [1] and VelocityGAN [16] prepare a downgraded version of the synthetic data for each target dataset for pre-training. Meanwhile, progressive transfer learning [17] has been applied to DL-FWI. For example, the network in [18] can gradually learn the real frequency information through the iteration of the physical module. Similarly, an hierarchical RNN improves the target difficulty by gradually adjusting the scaling factor of the joint loss function [19, 20]. In addition, there are learning strategies that focus on avoiding fixed datasets. For instance, multiCMP CNN [5] leverages randomization techniques for dynamic learning to achieve unlimited data expansion.

The generalization ability determines the validity of data-driven prediction methods. Regarding DL-FWI, maintaining the predictive performance of the network is a key issue when it is applied to new fields. Some networks trained in continuous formations are extended to prediction in fault [14]. Some networks trained in formations with fewer layers are expanded to predictions at more layers [10]. In fact, DL-FWI depends heavily on the number, characteristics and variability of the samples in the dataset [5]. Therefore, Deng et al. [21] integrated the synthetic data of major subsurface structures and used them to conduct more hierarchical generalization tests.

In this paper, we propose a dual decoder network with

Corresponding author: Fan Min. Tel.: +86 135 4068 5200.
Email: minfan@swpu.edu.cn.

Xing-Yi Zhang is with the School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China (e-mail: zxy20004182@163.com).

Fan Min is with the School of Computer Science and Software Engineering; Lab of Machine Learning; Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu 610500, China (e-mail: minfan@swpu.edu.cn).

Shulin Pan is with the School of Earth Science and Technology, Southwest Petroleum University, Chengdu 610500, China (e-mail: shulinpan@swpu.edu.cn).

Qiong Xu is with the School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China (e-mail: xuqiong@swpu.edu.cn).

Xue-Yang Min is with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: mitchellemin@163.com).

Guojie Song is with the School of Science, Southwest Petroleum University, Chengdu 610500, China (e-mail: songgj@swpu.edu.cn).

Ke Wang is with the Sichuan Egretech Technology Co., Ltd, Chengdu 610041, China (e-mail: 346068671@qq.com).

curriculum learning to handle these three issues mentioned above. For the network design, we employ dual decoders based on the U-Net [22] architecture. The first decoder focuses on reconstructing the distribution of the velocity values. The architecture is similar to that of FCNVMB [1]. The second decoder is used as an auxiliary to strengthen the layer boundaries. Edge decoders have proven their effectiveness as branching tasks in certain areas, such as tumor recognition in medicine [23]. Our structure exploits the hard parameter sharing mechanism in multi-task learning [24, 25]. This sharing of parameters and performance of the main/auxiliary tasks ensures a better overview of the original task [19, 20, 26]. Finally, the joint loss function of the mean square error and cross-entropy is set to control the output.

Regarding the training process control, we introduce curriculum learning [27] to enhance the network stability. In fact, curriculum learning is successfully utilized in image and natural language processing [28, 29]. As a somewhat related approach, the progressive transfer learning on seismic low-frequency data [18] has demonstrated the potential of data adaptation. Furthermore, a multi-level hierarchical network for 1D velocity estimation further demonstrates the ability of predefined gradients for data fitting [19]. The follow-up of this research is further expanded to 2D [20]. These results all indicate that the gradient difficulty strategy enables DL-FWI to facilitate data fitting. In curriculum learning, objectively evaluating the difficulty level of network inputs is not an easy task [30]. We infer the difficulty measurer based on single-shot to multi-shot through the study of different shots in FCNVMB [1]. The training scheduler will arrange these data with different difficulty levels in different batches.

To enhance the adaptability of our network design, we design a flexible pre-network dimension reduction structure. When collecting in the field, the number of time sampling points and geophones may be significantly different. This will result in a very large aspect ratio of the common-shot gather image. Although conventional approaches, such as interpolation, can handle the issue [1, 31], in some seismic observations with low resolution [21], this may lead to inevitable information loss. Therefore, we condense the time dimension through a dimension reduction structure composed of multiple convolutions [14]. It can be used as a common design component for acquisition geometries that are sampled at different times.

Experiments are undertaken on SEG salt [32] and OpenFWI datasets [21] with four metrics including MSE, MAE, UIQ and LPIPS [33, 34]. Meanwhile, two DL-FWI networks, FCNVMB [1] and InversionNet [14], are compared with our network. The results show that: 1) Our network outperforms its counterparts in different datasets; 2) The integration of curriculum learning and DL-FWI improves the network learning ability; 3) The dual decoder facilitates the reconstruction of edge details in velocity models; and 4) Our network has good generalization ability for a variety of data.

The rest of this paper is organized as follows. In Section II, two main approaches in DL-FWI are presented. In Section III, the network structure, curriculum learning and loss function of the proposed method are described in detail. In Section IV, the datasets and training details are described. In Section V, the experimental results of our network on two datasets are presented. In Section VI, the experimental results in Section V are discussed and some characteristic analysis of the network are supplemented. Finally, in Section VII, a conclusion is provided.

II. Related work

In a data-driven approach, we will perform inversion entirely based on the learning network. This process is quite challenging because it reduces the reliance on prior knowledge of the wave equation. Complex deep networks will be designed as the main tool to fit the nonlinear mapping relationship of FWI. The DL-FWI approaches adopted thus far can be roughly divided into three categories: 1) end-to-end encoder-decoder network; 2) embedding feature conversion; and 3) optimization of learning strategies and network structures.

The complete end-to-end structure is an intuitive attempt at data-driven solutions. Moseley et al. [35] constructed a 1D inversion task for each trace by imitating the voice architecture WaveNet [36]. While the construction of deep networks from an individual trace perspective is uniquely analyzed in this approach, the mutual influence between traces is ignored. ModifiedFCN [12] is the first attempt to utilize the FCN [11] architecture for FWI. This shows the inversion potential of the FCN for seismic data, but the generalization ability of the network is limited. FCNVMB [1] performs transfer learning by generating synthetic data, thereby improving the generalization ability of the transferred data distribution. However, it still struggles to accurately image background details and stratum edges. In InversionNet [14], a new CNN encoder-decoder network that is applied to more stratum structures, such as curved interfaces and faults, is proposed. While it effectively condenses spatial information to reduce computational costs, it also sacrifices certain specialized neighborhood and global information [9].

By embedding features, the burden of the initial deconstruction of the spatial features on the network can be alleviated. GeoDNN [6] resets the prestack data structure to obtain a semblance cube that is used to train DNN [7]. While it simplifies the data structure, it does not fully exploit the spatial relationships between different shots. SeisInvNet [9] integrates DNN feature generation and CNN decoding, while encoding more global, neighborhood and observation features. However, it lacks a variety of data models that can be used to verify its applicability and universality. Based on SeisInvNet, Liu et al. [10] supplemented the global and neighborhood information of the common-receiver gather during the embedding process. It improves the data generalization of SeisInvNet but inverts poorly for diffraction-generated information such as faults.

Some networks memorize more seismic data details by indirect training control and specific network structures. VelocityGAN [16] uses a DCGAN-like [37] architecture to distinguish the difference between the generated velocity model and the ground truth. It treats the discriminator as a regularizer to enhance the inversion accuracy of the encoder-decoder generator [38]. The multiCMP CNN [5] generates

static validation data while asynchronously using random data for training. This dynamic process greatly expands the breadth of the data distribution, thereby preventing overfitting. Feng et al. [31] proposed a two-step alternate training scheme for high-frequency velocity models. It relieves the parameter coverage of multi-task networks and alleviates the convergence imbalance problem.

III. Methodology

A. Architecture of DD-Net

Fig. 1 shows the architecture of DD-Net. The basic idea is adopted from the popular U-Net [22]. Therefore, DD-Net also exhibits an obvious left-right highly symmetric encoder-decoder structure. However, we use two decoders, making the overall structure Y-shaped.

The encoder assembly on the left side is in charge of the compression process for seismic data. It further compresses the seismic observation records of 29 shots into 1024-dimensional abstract structured information. The two decoders on the right use different ideas to interpret high-dimensional abstract information. The target of the first decoder is the conventional velocity model, which focuses on the accurate fitting of velocity values. It will be the main decoder for prediction. Then, the binary velocity model after contour extraction by Canny [41] is further used as the fitting target for the second decoder. This decoder will serve as an auxiliary decoder for training contour information.

The traditional U-Net still has problems with rough and discontinuous contours when processing image segmentation. However, a multi-decoder provides us with a very interesting way to address these deficiencies [23]. Different decoders turn U-Net into a multi-task learning environment. An inherent shortcoming of U-Net is addressed by specialized decoders of specific tasks. The subtask decoder can also provide parameter context for the main task or subsequent tasks [19, 20, 31]. With this in mind, we apply an additional contour decoder to ensure the smoothness of the segmentation predictions.

There are two basic operations in the network, namely convolution and deconvolution [42], for the encoder and decoders, respectively.

(a) In the encoder, repeated convolutions take full advantage of the continuous information in the seismic waveform. The BN [39] normalizes the subset of data fed into the network after convolution, thereby improving the convergence. The ReLU activation function guarantees layer-to-layer nonlinearity by removing some network nodes [40]. Meanwhile, through the nonlinear superposition of multiple ReLUs, our deep network can approximate the mapping of the wave equation [43].

(b) In the decoder, deconvolution operations can expand highly compressed information into higher-resolution images, while preserving as much information as possible [42]. A skip connection is used to obtain higher-resolution encoder features. In addition, the shallower convolutional layers preserve fine-grained differences between different shots. Therefore, the feature maps of the shallower layers as additional channels [22] can further guide the decoder to describe the boundary information of overlapping detection regions. Table I lists some detailed parameter settings of the neural network.

TABLE I
DD-Net network layer details configuration

Tag in Fig. 1	Kernel size	Stride size	Padding size
3×3 conv, BN, ReLU	3×3	1×1	1×1
2×2 max pooling	2×2	2×2	0×0
2×2 deconv	2×2	2×2	0×0
1×1 conv	1×1	1×1	0×0

A common-shot gather is often associated with extremely large aspect ratios. For some low resolution observations with a small number of geophones, directly using interpolation for downsampling may cause some information to be lost. Therefore, we build a modified version of DD-Net with a dimension reducer for 1000×70 low resolution observations. We named it DD-Net70 based on the number of geophones.

In Fig. 2, DD-Net70 extends a pre-network dimension reducer in front of the network. Meanwhile, a non-square convolution is introduced to compress the time dimension (image height), ensuring that the seismic data is constrained to the same size as the velocity model. The dimension reducer will use multiple convolutions to coordinate to prevent the information loss caused by one convolution compression. This setup is similar to that of InversionNet [14], but we generalize it to the U-Net structure. In fact, this architectural design idea can still be generally applied to other low resolution common-shot gather images.

B. Loss Function

To handle different tasks of different decoders, we propose a new joint loss function for the network architecture. This loss function consists of the mean square error (MSE) and cross-entropy loss function. The first decoder directly outputs a single-channel image. It performs a one-to-one pixel difference loss computation with the ground truth velocity model. The second decoder outputs a two-channel image. These two channels interact to simulate the contour information of the velocity model. In fact, related research has demonstrated the feasibility of cross-entropy loss serving the binary classification subtask in velocity reconstruction [19, 20].

First, the MSE loss function is a common loss function for describing color similarity, *i.e.*,

$$\mathcal{L}_{\text{main}} = \frac{1}{mb} \sum_{i=1}^{b} \left(v_i - \mathcal{N}_1^{(1)}(d_i) \right)^2, \tag{1}$$

where m represents the number of pixels of the output velocity model. d_i and v_i represent the seismic data and their corresponding ground truth velocity models, respectively. b denotes the number of data to be trained simultaneously in each iteration, that is, the batch size. $\mathcal{N}(\cdot)$ represents a network-determined mapping of the seismic data d_i to the predicted velocity model $\hat{v}_i$. Furthermore, $\mathcal{N}_j^{(k)}(\cdot)$ represents the j-th channel of the predicted velocity model in the k-th decoder output. As a consequence, $\left(v_i - \mathcal{N}_1^{(1)}(d_i) \right)$ is the

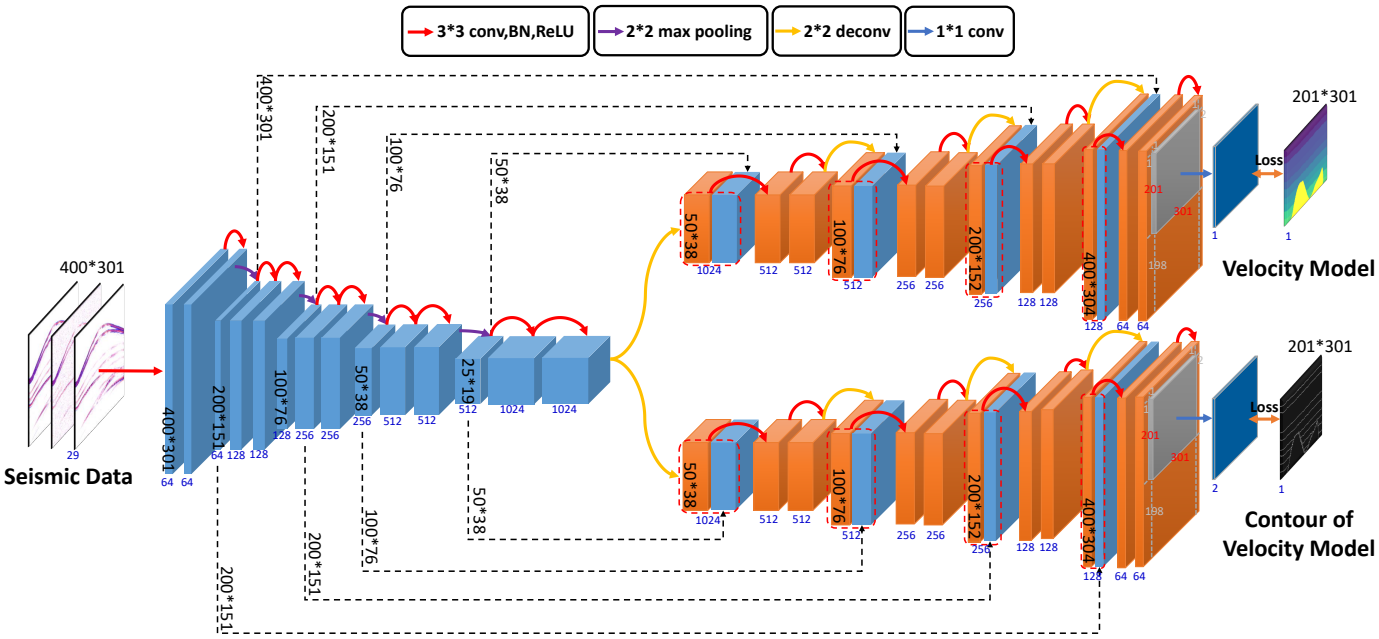


Fig. 1. Deep network architecture of DD-Net for inversion. The width of each cube represents the number of channels in the image. The number of channels is marked in blue below the cube. The black number indicates the image size of the current network layer (*e.g.*, 400×301 input). The arrows above the figure are the simplified representations of the convolution and pooling operations. Among them, the red arrows represent three consecutive fixed operations, namely convolution, BN (batch normalization) [39] and ReLU [40]. The blue cube represents the information of the encoder, and the orange represents the decoder. The red dotted box indicates the concatenation operation of the two-part channel. The dashed black arrows indicate the source of the features input to the decoder, during which the original decoder dimensions are extended.

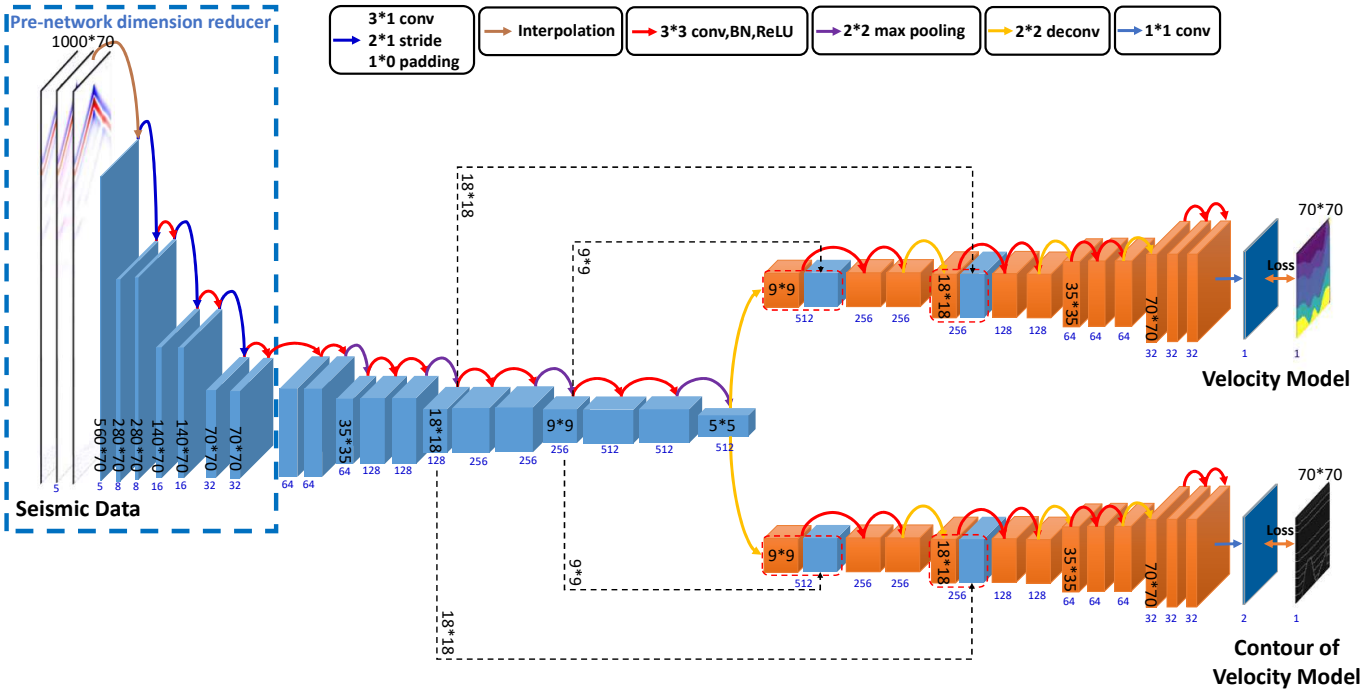


Fig. 2. Deep network architecture of DD-Net70 for inversion. The structure is similar to that of DD-Net (as shown in Fig. 1). The difference is a pre-network dimension reducer on the left. Additionally, non-square convolutions are used to assist the dimension reducer below the figure. Then, the input and output sizes of the network become $1,000 \times 70$ and 70×70, respectively.

residual error between the velocity model predicted by the seismic data d_i and the actual velocity model.

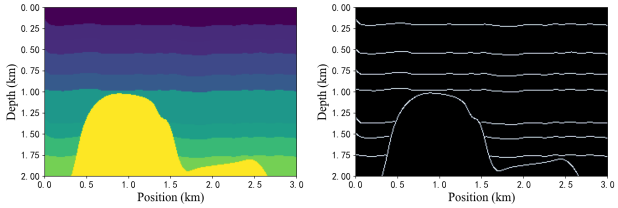
Second, the cross entropy loss function provides a contour fitting idea in contrast to the difference loss, *i.e.*,

$$\mathcal{L}_{\text{contour}} = \frac{1}{mb} \sum_{i=1}^{b} -\log p\big(s_i \mid \mathcal{C}(v_i)\big) - \log p\big(1 - s_i \mid \widetilde{\mathcal{C}}(v_i)\big), \tag{2}$$

where s_i is obtained by performing softmax on the two channels of the second decoder, *i.e.*,

$$s_i = \frac{e^{\mathcal{N}_1^{(2)}(d_i)}}{e^{\mathcal{N}_1^{(2)}(d_i)} + e^{\mathcal{N}_2^{(2)}(d_i)}}. \tag{3}$$

$\mathcal{C}(v_i)$ in Eq. (2) is the contour structure of the velocity model v_i. Figs. 3(a) and 3(b) show the structures of v_i and $\mathcal{C}(v_i)$, respectively. $\widetilde{\mathcal{C}}(v_i)$ is the inverse color of $\mathcal{C}(v_i)$. We



(a) (b)

Fig. 3. An example of a velocity model and its contour. (a) is the simulated velocity model; and (b) is the contour structure of this velocity model obtained by the Canny [41]. The dual thresholds of Canny are set to 10 and 15 respectively.

uniformly denote $\widetilde{\mathcal{C}}(v_i)$ and $\mathcal{C}(v_i)$ as a binary mask operator M. Simultaneously, $1 - s_i$ and s_i are represented as the predicted grayscale image I. We use $I \cdot M$ to extract attention information from I. In terms of conditional probability, we can further interpret $p(\cdot)$ in Eq. (2) as

$$p(I \mid M) = \frac{\|I \cdot M\|_1}{m}. \tag{4}$$

The more obvious the contour is, the larger $p(I \mid M)$ is. To ensure that the loss is smaller at this time, the cross-entropy loss introduces negative logarithms. Thus Eq. (4) is often replaced as

$$-\log p(I \mid M) = \frac{\|-\log(I) \cdot M\|_1}{m}. \tag{5}$$

For samples with large marginal biases, the cross-entropy loss is more penalized. This ensures that the fitting loss can be focused on the contour instead of computing residuals globally. Meanwhile, the pixel proportion of the contour in the image is small. This may lead to an unbalanced distribution of pixels in the contour image. Compared with MSE, the cross-entropy loss is used to considered the differences of different classes, guaranteeing a fairer consideration of the contour and background information. The design of this similar contour loss function has been practiced in the field of image segmentation [23].

Finally, we combine the two loss functions through hyper-parameters α_1 and α_2, *i.e.*,

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{main}} + \alpha_2 \mathcal{L}_{\text{contour}}. \tag{6}$$

The proportion of these two parameters can be adjusted according to different situations. To predict accurate velocities, some networks may be trained and predicted with unnormalized velocity models [1]. At this time, the order of magnitude difference in the losses of the two decoders can be very large. For example, when the SEG salt model [32] is not normalized, $\frac{\alpha_2}{\alpha_1}$ is set to 10^6, but for the normalized OpenFWI [21] model, this value is often set to 10 or 10^2.

C. Curriculum Learning

Numerous training ideas for networks attempt to mimic human thought processes. Curriculum learning [27] is a simulation of human progressive cognition.

1) Curriculum Learning of FWI: In traditional machine learning, the difficulty attribute of the dataset presented to the network is often random. Thus, its complexity and the current learning state of the network are ignored. However, curriculum learning has changed this deficiency with the idea of staged adaptation, and has achieved considerable results in many fields [28, 29]. It tries to gradually increase the difficulty of training datasets to ensure an efficient fit. Specifically, in the beginning, we start with a relatively simple dataset to train the network. This helps the network converge quickly and prevents the local optima from being stuck. Then, as the network reaches a certain performance level, we gradually introduce more complex datasets.

If only a single-shot point is used, then the subsurface information obtained by the network is the local area covered by the wave. Therefore, we use the seismic wave reflection records fed back by multiple shot points to jointly predict the entire subsurface area. However, in some deep learning strategies [1], single-shot data can also be used to achieve certain predictive effects. This means that the network can construct a memory map from the local waveform and frequency to the entire velocity model through learning. Although this mapping is not graceful from a physical perspective, it can be used as a pilot for network learning. It guides the network to leave a basic style impression from the time-domain waveform to velocity models. This process is in line with the idea of curriculum learning.

2) Predefined Curriculum Learning for Seismic Data: The usage of curriculum learning techniques in the network must specify the details of both components [30].

- **Difficulty Measurer**: The difficulty measurer tells us which data are difficult. In other words, it can decide the priority of different data.
- **Training Scheduler**: The training scheduler tells us when and what difficulty data need to be trained. Meanwhile, it also tells the network the training parameters of the selected data, including the number of epochs, and learning rate.

Currently, curriculum learning can be subdivided into predefined and automatic curriculum learning according to whether the measurer and scheduler are customized or data-driven. Given the lack of a prior theory describing the difficulty of seismic data, we use predefined curriculum learning. Fig. 4 shows the three stages we adopted in the specific setting of the curriculum.

(a) Single-shot data with distorted information: The seismic observation records consist of q common shot gathers. We fetch the $\lceil \frac{q}{2} \rceil$-th record among them. First, noise and amplitude amplification are performed on this record to obtain two distorted records. Second, we copy $\lceil \frac{q}{3} \rceil$ copies of each distorted record and $q - 2\lceil \frac{q}{3} \rceil$ copies of the original record. Finally, we concatenate the three to obtain a combination of q observation records.

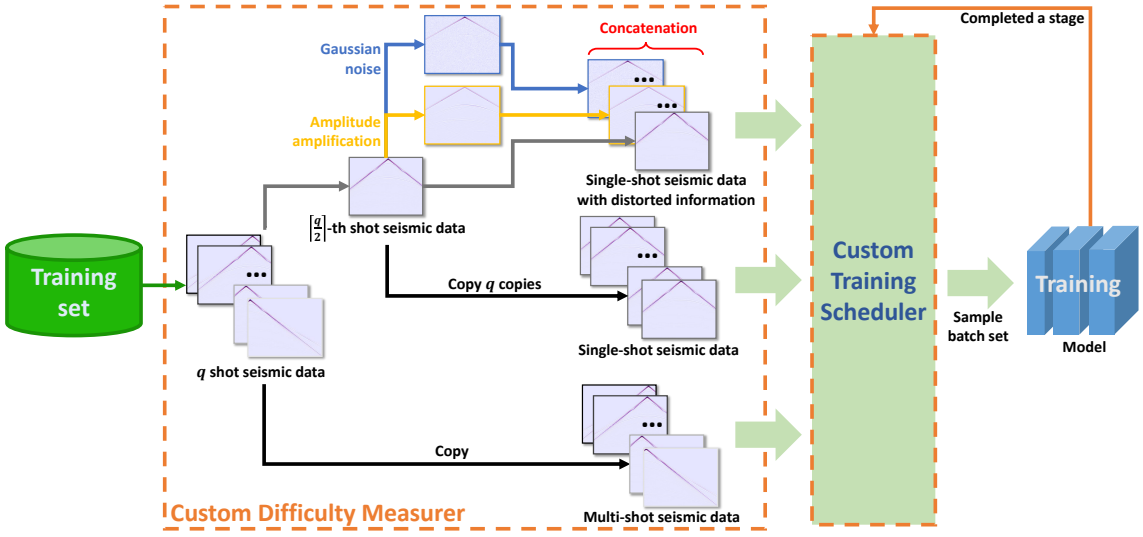


Fig. 4. A predefined curriculum process based on FWI. The orange area in the middle demonstrates how we process the seismic data into three-stage data. The arrows of different colors in the orange box represent different operations. The blue and yellow arrows represent adding noise and amplifying the amplitude, respectively. q denotes the number of seismic sources during simulation detection, which is also equivalent to the number of input channels of the network. The right side shows how we arrange the three-stage data into the network. The circles formed by the orange and green arrows represent iterations of a stage in the training process. This iteration will be performed three times. One iteration will feed multiple batches for training.

(b) Single-shot data: We fetch the $\lceil \frac{q}{2} \rceil$-th record among the seismic observation records. Then, we obtain q copies of this record. These copies as a whole are used as the data of this stage through concatenation.

(c) Multi-shot data: The seismic observation records with q shot points are directly used as the data of this stage. Because the input port of the network is determined, curriculum data at each stage always carry q channels.

There are two considerations for placing the single-shot data before the multi-shot data. On the one hand, single-shot data are fed into the network after multiple copies. The fewer the channels of the input layer that differ from each other, the fewer training samples that the network needs to capture the differences between inputs. Therefore, compared to the multi-shot data, the single-shot data have a faster fitting ability. On the other hand, although the stratigraphic region detected by single-shot data is limited, it can be used as a style guide for multi-shot data training. An initial cognitive network will be first established from the waveform to velocity model through incomplete information and short-term training. Additionally, the motivation for setting distorted information before the single-shot data is similar to data augmentation. The robustness of the network is ensured by pre-training a mixed noisy version of the original data.

Our custom training scheduler sets different training epochs for different stages. First, the scheduler will feed data to the network in the order set. Meanwhile, the data fed to the network will be trained independently. Second, when the number of training epochs reaches the predefined upper bound, a new round of scheduling is performed. There are significant differences in the distribution of data within different datasets. Therefore, there is also an obvious adaptability gap of the network to the three-stage data. For this reason, the different scheduling information may be used in the experimental settings of different datasets in Section IV.

IV. Experimental Setting

First we introduce the two types of datasets that we used in this study.

1) SEG salt and its simulated data will be used to train DD-Net. The purpose is to explore the transfer learning ability of our network.
2) OpenFWI datasets will be used to train DD-Net70. The purpose is to explore the adaptability of our network to different subsurface structures.

Then, we will discuss the training and forward setup. Finally, we will highlight two of the four evaluation metrics that we employed.

A. SEG Salt Datasets

SEG salt datasets are mainly composed of two parts: the salt data and its simulated synthetic data.

1) Salt Data: The salt data are an open-source 3D dataset from the SEG Research Committee [32]. In existing studies, the 2D data of 140 cross-sections are extracted from the 3D model, and related experiments are conducted [1, 16]. These cross-sectional data can be downloaded from GitHub [1].

The salt data describe an area of approximately 2 km $\times$ 3 km underground, with a pixel size of 201×301. The wave propagation velocity varies from $1,500$ m/s to $4,482$ m/s. We call this dataset "SEGSalt".

2) Synthetic Data: Considering the lack of SEGSalt, we need to pre-train our network with simulated data in advance. In our experiments, we adopt the synthetic dataset used by FCNVMB [1]. This simulated dataset has $1,700$ synthetic velocity models of the same size as the SEGSalt data. The number of strata in these velocity models ranges from 5 to 12

[1]https://github.com/YangFangShu/FCNVMB-Deep-learning-based-seismic-velocity-model-building

layers. There is an irregular salt dome in each velocity model. Meanwhile, the formation velocity fluctuation of the simulated data also is controlled at $2,000$ m/s to $4,500$ m/s. We call this dataset "SEGSimulation".

B. OpenFWI Datasets

OpenFWI [21] is an open source dataset containing numerous synthetic seismic data [2]. It includes interfaces, faults, CO_2 reservoirs, 3D underground structures and other stratum data types. In our experiments, we focus on the identification of the subsurface interfaces and faults. Therefore, we mainly use four OpenFWI datasets: FlatVelA, FlatFaultA, CurveVelA and CurveFaultA. Each of these datasets describes a 0.7 km $\times$ 0.7 km subsurface region with a pixel size of 70×70. The wave propagation velocity ranges roughly between $1,500$ m/s and $4,500$ m/s.

In addition, OpenFWI also provides an analysis of the velocity model complexity. Three metrics that are used to measure the complexity of the velocity model are introduced in [21], *i.e.*, spatial information [44], gradient sparsity index [45] and Shannon entropy [46]. Table II shows the numerical results of the complexity of the four datasets that we selected under these metrics. We found that the fault and curved data are harder than the non-fault and flat data.

TABLE II
Comparison of the evaluation results of the three metrics spatial information, gradient sparsity index and Shannon entropy on the datasets FlatVelA, FlatFaultA, CurveVelA and CurveFaultA. The weight ratio of the weighted average is $10 : 10 : 1$. The complexity of the dataset increases from top to bottom.

Datasets	Spatial of information	Gradient sparsity index	Shannon entropy	Weighted average
FlatVelA	0.07	0.12	2.30	1.40
CurvelVelA	0.10	0.24	2.38	1.93
FlatFaultA	0.10	0.26	2.92	2.17
CurvelFaultA	0.11	0.32	3.50	2.60

Finally, it is also practical to compare these datasets. The FlatVelA and CurveVelA datasets reflect the flat and curved underground strata that have clear interfaces. The FlatFaultA and CurveFaultA datasets reflect the subsurface fault structure caused by the displacement of the rock layer. Fault formations can trap fluid hydrocarbons and are therefore very good reservoirs [47]. Meanwhile, providing an accurate fault description will help to perform reservoir characterization and well placement [48].

C. Training and Forward Settings

Before training, it is necessary to use numerical approaches to carry out forward modeling on the velocity model to obtain common-shot gathers. The OpenFWI datasets are simulated using the finite-difference [49] with absorbing boundary conditions [50]. Meanwhile, the optimization scheme of the second-order time direction and the fourth-order space direction is adopted. However, for the SEG salt datasets, its spatial direction adopts a sixth-order optimization scheme. Table III provides more parameter differences between the two.

Table IV shows some important parameters involved in the training process. Before training the SEG salt datasets, we first train SEGSimulation to obtain the pre-trained DD-Net network. Then, we only need to perform a small amount of training on the SEGSalt to obtain the transferred DD-Net network. Therefore, we do not set up curriculum learning for SEGSalt. However, the OpenFWI datasets do not require transfer learning because of their large amount of data.

Fig. 5 shows some forward simulation results for all datasets. The receiver numbers and time steps can be thought of as the width and height of the common-shot gather. The size of the SEG simulated seismic observation data is 2000×301. It will be downsampled to 400×301 before being fed into the network.

To verify the performance of our network on the corresponding datasets, we compare FCNVMB [1] with DD-Net and InversionNet [14] with DD-Net70. They all have convolution parameter settings similar to their respective comparison algorithms. Among them, the dimension reduction strategy of DD-Net70 also draws on InversionNet. However, from the viewpoint of the architecture, both DD-Net and DD-Net70 adopt dual decoders that are different from the comparison algorithms. Moreover, the skip connection strategy of FCNVMB is different from that of DD-Net. InversionNet does not even use skip connections. Meanwhile, DD-Net70 does not excessively condense the feature map to 1×1 at the end of the encoder, similar to InversionNet. This strategy may result in a loss of details in the velocity model [9].

In the training details, the parameter settings of FCNVMB follow the original paper. Its training epoch is 100, the learning rate is 0.001, and the batch size is 10. The parameter settings of InversionNet are mostly the same as the one provided in [21], *i.e.*, epoch = 120 and learning rate = 0.0001. The only difference is that the batch size is adjusted from 256 to 128. In this way, the performance is improved, as will be depicted in respective tables and figures.

D. Evaluation Metrics

We mainly use two metrics to measure the similarity of images, *i.e.*, UIQ (a universal image quality index) [33] and LPIPS (learned perceptual image patch similarity) [34]. However, at the same time, we will still implement two basic evaluation metrics based on the pixel difference: MSE (mean square error) and MAE (mean absolute error). They will be used as the basis for determining whether the prediction of the underground velocity values is accurate.

1) UIQ: The UIQ calculates the correlation between two images by numerically analyzing the global average pixel, variance and covariance. Compared with traditional error summation methods, the image structure information described by the UIQ is more in line with human perception. It is given by

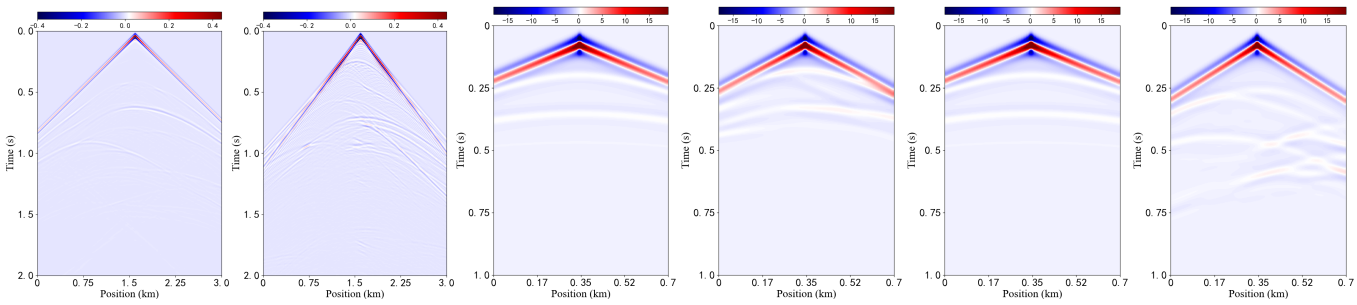
$$\mathrm{UIQ} = \frac{\sigma_{\mathbf{xy}}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}} \cdot \frac{2\bar{\mathbf{x}}\bar{\mathbf{y}}}{(\bar{\mathbf{x}})^2 + (\bar{\mathbf{y}})^2} \cdot \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2}, \tag{7}$$

where $\frac{\sigma_{\mathbf{xy}}}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}$ measures the degree of linear correlation between the target and predicted velocity model. Its value

[2]https://openfwi-lanl.github.io/

TABLE III
Seismic forward modeling configuration.

Datasets	Grid spacing	Source frequency	Source spacing	Source numbers	Receiver spacing	Receiver numbers	Time Spacing	Time Steps
OpenFWI datasets	10 m	15 Hz	140 m	5	10 m	70	0.001 s	1,001
SEG salt datasets	10 m	25 Hz	103.45 m	29	10 m	301	0.001 s	2,000



(a) SEGSimulation (b) SEGSalt (c) FlatVelA (d) FlatFaultA (e) CurveVelA (f) CurveFaultA

Fig. 5. Seismic data after forward modeling of different velocity models. The selected seismic images are all from intermediate sources. The size of (a) and (b) is 2000×301. The source of these two seismic records is located at a horizontal distance of 1.6 km. The size of (c) to (f) is 1000×70. The source of these four seismic records is located at a horizontal distance of 0.35 km.

TABLE IV
Training configuration of DD-Net and DD-Net70. The fifth column indicates the training epoch proportion of the three tasks in Subsection III-C.

Datasets	Number of training set	Number of testing set	Batch size	Training epoch scheduling proportion
SEGSimulation	1,600	100	10	40 : 30 : 30
SEGSalt	130	10	10	0 : 0 : 50
FlatVelA	24,000	6,000	64	10 : 10 : 100
FlatFaultA	48,000	6,000	64	10 : 10 : 80
CurvelVelA	24,000	6,000	64	10 : 10 : 90
CurvelFaultA	48,000	6,000	64	10 : 10 : 90

varies between $[-1, 1]$. $\frac{2\bar{\mathbf{x}}\bar{\mathbf{y}}}{(\bar{\mathbf{x}})^2+(\bar{\mathbf{y}})^2}$ measures how close the predicted image is to the target image in terms of average pixel brightness. $\frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}}{\sigma_{\mathbf{x}}^2+\sigma_{\mathbf{y}}^2}$ measures the similarity of contrast between images. The value range of the last two components is $[0, 1]$. The interaction of the three determines that the higher the UIQ is, the higher the similarity. The SSIM (Structural Similarity) [51] metric is obtained based on the improvement of the UIQ. Specifically, the SSIM multiplexes these three components and adds a positive constant to each component to improve the evaluation credibility in areas with small pixel values. Additionally, variable positive constants are added as the exponent of each component, *i.e.*,

$$\mathrm{SSIM} = [l(\mathbf{x}, \mathbf{y})]^{\alpha}[c(\mathbf{x}, \mathbf{y})]^{\beta}[s(\mathbf{x}, \mathbf{y})]^{\gamma}, \tag{8}$$

where the explanations of $l(\mathbf{x}, \mathbf{y})$, $c(\mathbf{x}, \mathbf{y})$ and $s(\mathbf{x}, \mathbf{y})$ are

$$\begin{cases} l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{y}}+C_1}{\mu_{\mathbf{x}}^2+\mu_{\mathbf{y}}^2+C_1}; \\ c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}+C_2}{\sigma_{\mathbf{x}}^2+\sigma_{\mathbf{y}}^2+C_2}; \\ s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{\mathbf{xy}}+C_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{y}}+C_3}. \end{cases} \tag{9}$$

2) LPIPS: However, the mathematically defined measures are satisfactory in many scenarios. These methods also have defects in some smooth images. Therefore, a perceptual loss method based on network features is applied to image distance calculation, *i.e.*, LPIPS [34]. The LPIPS is used to employ learned networks to act as surrogates for human perception. It determines the structural distance of each corresponding patch in the two images by convolution.

Specifically, the two images are fed into the learned network with the same structure. Then, the distance between the feature maps of each i-th layer in these two networks is given by

$$d_i(\mathbf{x}, \mathbf{y}) = \sum_i \frac{1}{H_i W_i} \sum_{h,w} \left\| \mathbf{z}_i \odot \left(\hat{\mathbf{x}}_{hw}^i - \hat{\mathbf{y}}_{hw}^i \right) \right\|_2^2. \tag{10}$$

where $\hat{\mathbf{x}}^i, \hat{\mathbf{y}}^i \in \mathbb{R}^{H_i \times W_i \times C_i}$ represent the normalized feature maps of the two images obtained at the i-th layer. $\mathbf{z}^i \in \mathbb{R}^{C_i}$ represents a vector for adjusting the weights of different layers. Finally, the LPIPS will be obtained by the average value of d_i in each layer.

V. Experimental Results

All experiments are performed on a workstation equipped with an NVIDIA RTX 3060 GPU, a 13th Gen Intel (R) Core (TM) i5-13600K, and a Windows operating system implementing PyTorch. The training runtimes of FCNVMB and DD-Net are close, approximately 140 s per epoch. The training runtime for the dataset used for transfer learning is shorter than that for the original dataset, almost only 40 s per epoch. The training runtimes of InversionNet and DD-Net70 are approximately 125 s and 80 s per epoch for FlatVelA and CurveVelA, respectively. The training runtimes of FlatFaultA and CurveFaultA is twice that of CurveVelA and FlatVelA. Regardless of the network, the average running time of a single sample in a testing batch hardly exceeds 0.3 s.

A. Transfer Learning Performance on the SEG Salt Dataset

Fig. 6 shows the inversion results of DD-Net and FCNVMB on SEGSimulation. Table V demonstrates the advantages of DD-Net from the perspective of evaluation measures. The salt body shown in the first row of the figure is at the bottom of the probed depth. The second row is the display of the salt body at the middle depth. Compared with FCNVMB, the edge of the salt body of DD-Net is inverted more clearly. For example, there is an obvious redundant crack on the right side of the salt body in Fig. 6(f) but not in Fig. 6(e).

At the same time, not only the salt body itself but also the performance of DD-Net on details of inversion results. For example, for the boundaries, shown at the depth of 250 m and 750 m in Fig. 6(a), are clearly resolved by DD-Net. And it is clearer than the results of FCNVMB. This clarity is precise for every tiny dent and protrusion. Although there is still a gap between the deep details of DD-Net and the ground truth, compared with FCNVMB, the correct stratigraphic interface has basically taken shape. The boundary inversion results of DD-Net and FCNVMB at a depth of $1,500$ m in the stratigraphic region of Fig. 6(d) validate this assertion.

TABLE V
Comparison results of the evaluation metrics of DD-Net and FCNVMB in the SEGSimulation test dataset. The first part represents the average results of all test data. The next two parts represent the results in Fig. 6.

Network	MSE ($\times 10^4$) ↓	MAE ($\times 10^2$) ↓	UIQ ↑	LPIPS ↓
FCNVMB	6.467534	1.362893	0.995529	0.041795
DD-Net	**4.047805**	**0.974000**	**0.997324**	**0.032533**
Fig. 6(c)	5.442814	1.179624	0.996433	0.037635
Fig. 6(b)	**2.432738**	**0.706980**	**0.998546**	**0.029603**
Fig. 6(f)	6.955711	1.463810	0.995190	0.054956
Fig. 6(e)	**5.393489**	**0.994390**	**0.996325**	**0.047344**

Fig. 7 shows the inversion of DD-Net and FCNVMB on SEGSalt. The first and second rows show the two scenarios without salt bodies and with salt bodies, respectively, in SEGSalt. Fig. 7(c) shows that the FCNVMB can only invert the velocity but hardly resolve the formation interfaces. However, Fig. 7(b) reveals that DD-Net can do both.

According to Figs. 7(e) and 7(f), DD-Net outperforms FCNVMB in getting as close as possible to the correct salt body shape. Meanwhile, there should be a low-velocity zone below the salt body in Fig. 7(d). However, this area in FCNVMB is not as large as that in DD-Net. Compared with the formation without a salt body, the salt body in Fig. 7(e) interferes with the inversion of the stratigraphic interfaces of DD-Net. However, compared to FCNVMB, which cannot invert it, a large improvement is already achieved.

Table VI confirms that DD-Net has a stronger transfer learning ability than FCNVMB in SEGSalt from the perspective of the evaluation measures. In addition, comparing Tables V and VI, the gap between the two networks is largest in SEGSalt. This may indicate that the transfer training ability of DD-Net is superior to the fitting performance.

TABLE VI
Comparison results of the evaluation metrics of DD-Net and FCNVMB in the SEGSalt test dataset. The first part represents the average results of all test data. The next two parts represent the results in Fig. 7.

Network	MSE ($\times 10^4$) ↓	MAE ($\times 10^2$) ↓	UIQ ↑	LPIPS ↓
FCNVMB	20.340554	1.570017	0.982991	0.114347
DD-Net	**9.726930**	**1.081671**	**0.991866**	**0.071388**
Fig. 7(c)	1.035627	0.537435	0.998728	0.062763
Fig. 7(b)	**0.723318**	**0.525795**	**0.999135**	**0.023725**
Fig. 7(f)	38.570818	2.407324	0.968474	0.144820
Fig. 7(e)	**12.917457**	**1.304126**	**0.988983**	**0.096494**

B. Performance on OpenFWI Dataset

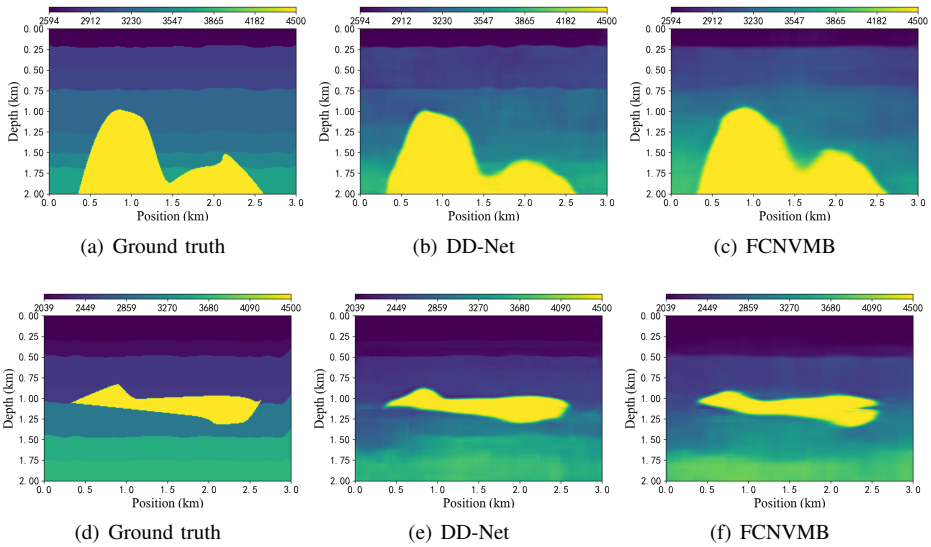
1) Performance on FlatVelA: The performance of InversionNet and DD-Net70 on FlatVelA is generally excellent. But in some extreme cases with small velocity changes, DD-Net70 may be better. Fig. 8 shows a case where deep velocity changes are small (the interface pointed by the red arrow). Although InversionNet can vaguely outline the boundaries, it is not as clear as DD-Net70. In Table VII, the metrics of DD-Net70 are significantly lower than those of InversionNet. Table VII also proves that these differences in details lead to the advantage of DD-Net70 in metrics.

TABLE VII
Comparison results of the evaluation metrics of DD-Net70 and InversionNet in the FlatVelA test dataset. The first part represents the average results of all test data. The second part represents the results in Fig. 8.

Network	MSE ↓	MAE ↓	UIQ ↑	LPIPS ↓
InversionNet	0.000041	0.004873	0.878029	0.003116
DD-Net70	**0.000039**	**0.003778**	**0.885140**	**0.000668**
Fig. 8(b)	0.000222	0.011488	0.942938	0.001438
Fig. 8(c)	**0.000115**	**0.006742**	**0.947739**	**0.001278**

2) Performance on FlatFaultA: The difficulty of FlatFaultA is to accurately identify the fault direction and discover the hidden layers around the fault. First, in Fig. 9(a), we use a yellow straight line to indicate the direction of the fault. DD-Net70 gives the expected direction, but InversionNet gives the wrong fault direction in the middle layer (velocity is about $3,100$ to $3,400$ km/s). This inaccuracy also exists in the inversion results in Fig. 9(d). In Fig. 9(d), we use three red lines to mark the fault directions of the three strata. Under correct inversion result, these three straight lines should be collinear. Compared to DD-Net70, InversionNet gives wrong angle and position. Second, there is a potential formation with low velocity marked by the orange box in Fig. 9(d). DD-Net70 gives a relatively obvious velocity difference with the surroundings in this area. Finally, Table VIII demonstrates the advantages of DD-Net70 from the perspective of the metric.

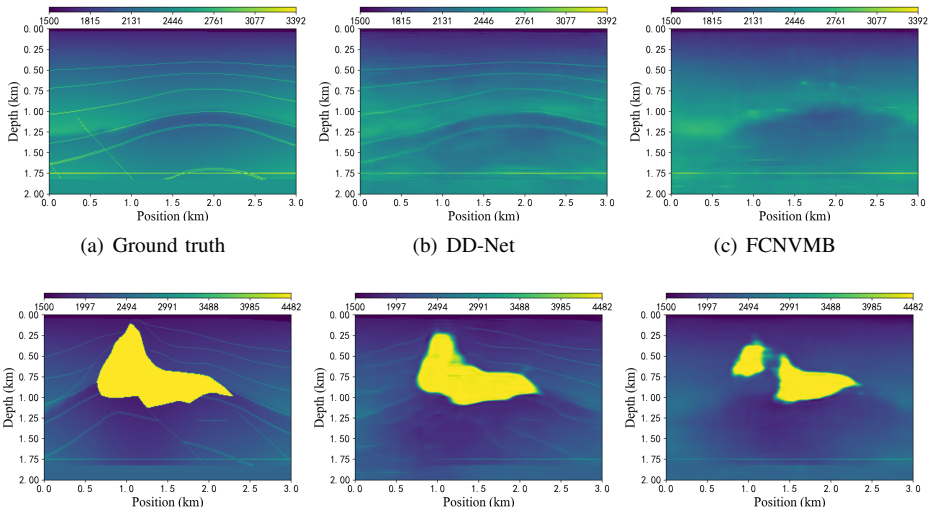
3) Performance on CurveVelA: The difficulty of CurveVelA focuses on the inversion of the number of strata and the strike of the interface. First, the strata marked by the red arrow in Fig. 10(a) are correctly inverted by DD-Net70 in Fig. 10(c). However, it is mistakenly recognized as two layers on InversionNet in Fig. 10(b). In addition, DD-Net70 also has better deep formation Inversion capabilities. The red



(a) Ground truth (b) DD-Net (c) FCNVMB

(d) Ground truth (e) DD-Net (f) FCNVMB

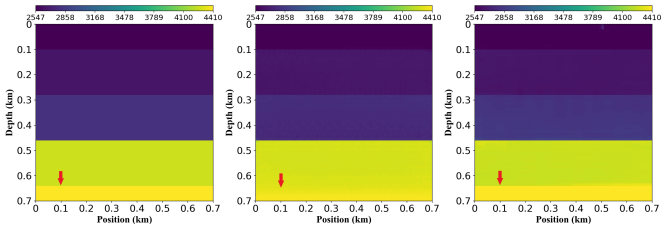
Fig. 6. Comparison example of the inversion results of DD-Net and FCNVMB after pre-training in the SEGSimulation test dataset.



(a) Ground truth (b) DD-Net (c) FCNVMB

(d) Ground truth (e) DD-Net (f) FCNVMB

Fig. 7. Comparison example of the inversion results of DD-Net and FCNVMB after re-training in the SEGSalt test dataset.



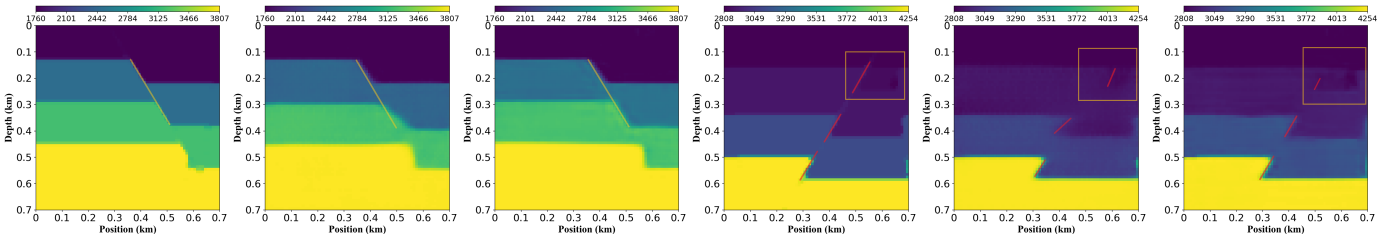
(a) Ground truth (b) InversionNet (c) DD-Net70

Fig. 8. Comparison example of the inversion results of DD-Net70 and InversionNet in the FlatVelA.

TABLE VIII
Comparison results of the evaluation metrics of DD-Net70 and InversionNet in the FlatFaultA test dataset.

Network	MSE ↓	MAE ↓	UIQ ↑	LPIPS ↓
InversionNet	0.000730	0.012618	0.861179	0.022774
DD-Net70	**0.000604**	**0.011101**	**0.861213**	**0.018049**
Fig. 9(b)	0.001416	0.016131	0.856938	0.036595
Fig. 9(c)	**0.000464**	**0.008104**	**0.862049**	**0.012467**
Fig. 9(e)	0.000593	0.013821	0.845548	0.024047
Fig. 9(f)	**0.000321**	**0.009363**	**0.865325**	**0.009961**

arrow in Fig. 10(d) indicates a deep formation with high velocity. Inversion of InversionNet almost confuse this area with outlying strata in Fig. 10(e). But DD-Net70 can give us a more obvious velocity difference in Fig. 10(f). Second, the strike of the interface specially marked by the red curve in Fig. 10(d) should be smooth. But there are obviously incorrect uplift in InversionNet, which is far less smooth than DD-Net70. Finally, Table IX confirms the excellence of DD-Net70



(a) Ground truth (b) InversionNet (c) DD-Net70 (d) Ground truth (e) InversionNet (f) DD-Net70

Fig. 9. Comparison example of the inversion results of DD-Net70 and InversionNet in FlatFaultA.

from a metric perspective.

TABLE IX
Comparison results of the evaluation metrics of DD-Net70 and InversionNet in the CurveVelA test dataset. The first part represents the average results of all test data. The next two parts represent the results in Fig. 10.

Network	MSE ↓	MAE ↓	UIQ ↑	LPIPS ↓
InversionNet	0.006901	0.045993	0.833045	0.106516
DD-Net70	**0.005329**	**0.041106**	**0.854070**	**0.085504**
Fig. 10(b)	0.003667	0.035748	0.860321	0.081421
Fig. 10(c)	**0.001649**	**0.023952**	**0.895917**	**0.031608**
Fig. 10(e)	0.012038	0.079429	0.822573	**0.121826**
Fig. 10(f)	**0.008300**	**0.057128**	**0.886421**	0.129628

4) Performance on CurveFaultA: CurveFaultA needs to realize fault prediction while taking into account the inversion of interface strike. First, in Fig. 11(b), InversionNet perfectly inverts the interface strike. However, the wrong depression appears in the marked red box. This may lead to misjudgment of faults. Second, in Fig. 11(e), the inversion of InversionNet for both faults (marked with orange lines in Fig. 11(d)) is accurate. However, strata (velocity greater than $3,600$ km/s) marked with red boxes that should not be continuous are incorrectly inverted. Finally, Table X proves that most metrics of DD-Net70 are better than InversionNet.

TABLE X
Comparison results of the evaluation metrics of DD-Net70 and InversionNet in the CurveFaultA. The first part represents the average results of all test data. The next two parts represent the results in Fig. 11.

Network	MSE ↓	MAE ↓	UIQ ↑	LPIPS ↓
InversionNet	0.002335	0.023032	0.824779	**0.024136**
DD-Net70	**0.001887**	**0.019628**	**0.832722**	0.027090
Fig. 11(b)	0.002377	0.029584	0.844224	**0.017032**
Fig. 11(c)	**0.001175**	**0.015468**	**0.851098**	0.017796
Fig. 11(e)	0.004576	0.034423	0.860697	**0.024353**
Fig. 11(f)	**0.002413**	**0.027564**	**0.873155**	0.044018

VI. Discussion and Analysis

After obtaining the results for all datasets, we further analyze some key features of the network. Specifically, we will answer five questions.

A. Does the network outperform its counterparts in different datasets?

On the one hand, from the perspective of numerical comparison, DD-Net and DD-Net70 can guarantee the optimal level of the four evaluation metrics. On the other hand, from the perspective of visual comparison, DD-Net and DD-Net70 can resolve the stratigraphic information that many traditional algorithms lack.

First, as shown in Fig. 6, we can see that DD-Net can more accurately invert the salt body. The salt body edge inverted by DD-Net is always smoother and has fewer artifacts. Meanwhile, this ability can be preserved through transfer learning. In Fig. 7(f), the salt body is incorrectly inverted as two separate parts. However, DD-Net almost inverts the shape of the entire salt body.

Second, both of our networks have an excellent ability to extract stratigraphic edges. For example, in Figs. 6 and 8, the boundaries inverted by our network are always sharper rather than smoother. Meanwhile, our network also maintains good inversion adaptability when the velocity difference on both sides of the edge is small. For example, the hidden formation marked in Fig. 9(d) has very small velocity differences from its surroundings. However, our network can still invert these details. Fig. 7 shows that DD-Net can invert stratigraphic interfaces that are difficult to determine with FCNVMB.

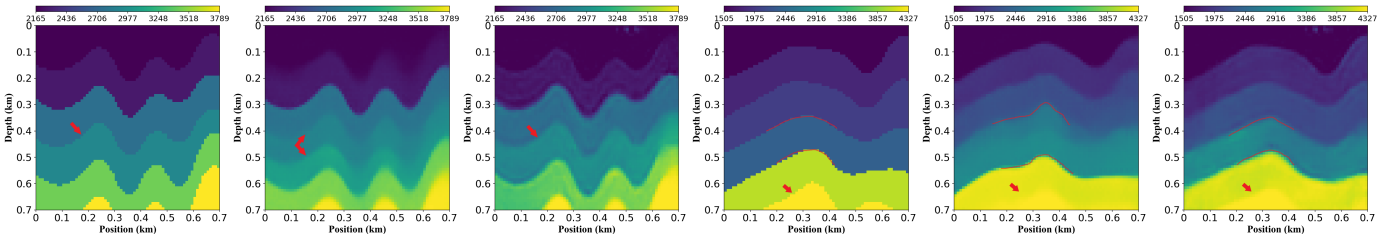
These datasets include a wide variety of underground structures, and results indicate that our network performs well on different datasets. Therefore, we can consider that under the existing experimental framework, our network works on different datasets.

B. Is it worthwhile to introduce the idea of curriculum learning in network training?

To facilitate the description of the experiments, we abbreviate the three curriculum tasks.

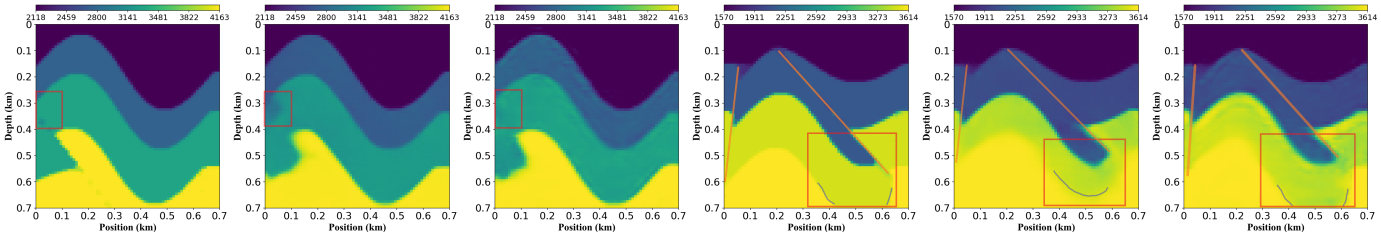
1) For the multi-shot seismic data, we abbreviate them as the M-strategy.
2) For the single-shot seismic data, we abbreviate them as the S-strategy.
3) For the single-shot seismic data with distorted information, we abbreviate them as the D-strategy.

In our network, the order of "DSM" is adopted for the customized training schedule mentioned in Subsection III-C. Defining the M-strategy as the last task is mandatory. Any



(a) Ground truth (b) InversionNet (c) DD-Net70 (d) Ground truth (e) InversionNet (f) DD-Net70

Fig. 10. Comparison example of the inversion results of DD-Net70 and InversionNet in the CurveVelA.



(a) Ground truth (b) InversionNet (c) DD-Net70 (d) Ground truth (e) InversionNet (f) DD-Net70

Fig. 11. Comparison example of the inversion results of DD-Net70 and InversionNet in the CurveFaultA test dataset.

inversion of single-shot seismic data lacks physical explanation and rationality. Based on this principle, we design three ablation experiments. They are the DD-Net networks in which the M-strategy, SDM-strategy and DM-strategy curriculum learning strategies are implemented. They answer three questions.

(a) Is curriculum learning better than conventional methods? The M-strategy represents the conventional method of training using all information at once. It will be used to test how curriculum learning differs from traditional courses. (b) Should the seismic wave information with distorted information be learned first? The only difference between the SDM strategy and our scheme is the order of "S" and "D". Distorted information can be a double-edged sword for the network. On the one hand, its addition enhances the robustness of the network. On the other hand, it may also learn local erroneous information during the training process, misleading the network. (c) Is the S-strategy task in the middle redundant? The DM strategy discusses the situation without an intermediate S strategy. This strategy can help explore whether transition tasks can help the network learn preparatory knowledge.

We use networks with different strategies as the initial network for SEGSalt transfer training to observe the fitting ability of different strategies in new similar scenarios. We evenly select various distributed data from SEGSalt as the validation set. Fig. 12 shows the validation set loss curve during training. First, by comparing the purple and red curves, it is found that the pre-trained networks have significantly better fitting speeds than randomly initialized networks. Second, by comparing the red and lower curves, it is found that the networks trained by curriculum learning have a better fitting advantage than conventional pre-trained networks. Third, by comparing the blue and upper curves, it is found that the DSM strategies may have better fitting abilities than other collocation strategies.

Meanwhile, in Table XI, the DSM strategy also has the best performance on SEGSimulation. However, upon examining the results presented in Table XI, the advantages of SEGSalt become more apparent. In other words, the true strength of this strategy lies in its ability to generate a pre-trained network that exhibits exceptional fitting skills.

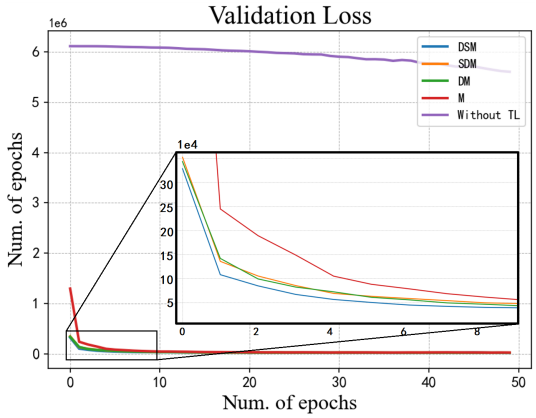


Fig. 12. Validation loss curve for SEGSalt transfer learning with various strategies. The network used for transfer is learned through three stages: Single-shot data with distorted information (D), Single-shot data (S), and Multi-shot data (M). "DSM, SDM, DM, M" respectively indicate the stages and scheduling sequence adopted. In addition, "Without TL" represents the results of training on a random network. It reflects the convergence speed of the network when learning without transfer

C. Does the second decoder contribute to velocity model reconstruction?

To study this issue, we employ an ablation strategy to examine the role of the second decoder. For comparison purposes, we construct the SD-Net (single-decoder network)

TABLE XI
Effects of the different curriculum learning schemes on the evaluation metrics of DD-Net: pre-training and re-training phases.

Dataset	Schemes	MSE($\times 10^4$)↓	MAE($\times 10^2$)↓	UIQ↑	LPIPS↓
SEG Simulation	DSM	**4.047805**	**0.973999**	**0.991866**	0.032533
	SDM	4.520725	1.147198	0.996907	0.038256
	DM	4.187494	0.992131	0.997232	**0.030750**
	M	5.863929	1.228347	0.996081	0.034688
SEGSalt	DSM	**0.972693**	**1.081671**	**0.991866**	**0.071388**
	SDM	1.678360	1.452829	0.985868	0.090469
	DM	1.479179	1.401037	0.987586	0.082824
	M	1.537818	1.390901	0.987064	0.082600

network by retaining only the first decoder from the dual decoder setup.

Fig. 13 illustrates the main role of the second decoder to make the identification of shallow interfaces clearer. SD-Net and SD-Net70 do not provide the expected clear and continuous interfaces strike in the direction pointed by the arrow in Fig. 13. However, DD-Net and DD-Net70 are better, especially the performance of DD-Net in the shallow layer of SEG salt data.

Because the proportion of contour pixels in the overall pixel is relatively small, it may be difficult for global metrics to distinguish differences objectively. Therefore, we adopt local MSE and MAE metrics to evaluate the fitting effects of the second decoder on the contour part. Specifically, we identify the contour area through the Canny operator. Only pixels near the contour area in the velocity model are used for MSE and MAE evaluation.

Table XII reflects the evaluation of local MSE and MAE metrics on the OpenFWI and SEG salt datasets. In the OpenFWI datasets, data containing faults are relatively better in Local MSE, while data with flat interfaces are better in Local MAE. In SEG salt datasets, the advantage of DD-Net in SEGSalt is more obvious.

TABLE XII
Comparison of local MSE and MAE between DD-Net70 and SD-Net70 networks.

Dataset	Network	Local MSE↓	Local MAE↓
FlatVelA	SD-Net70	0.000256	0.003836
	DD-Net70	**0.000212**	**0.002976**
CurveVelA	SD-Net70	0.004430	0.025307
	DD-Net70	**0.003884**	**0.024465**
FlatFaultA	SD-Net70	0.000782	0.006861
	DD-Net70	**0.000467**	**0.005181**
CurveFaultA	SD-Net70	0.002023	0.013765
	DD-Net70	**0.001530**	**0.011564**
Dataset	Network	MSE($\times 10^4$)↓	MAE($\times 10^2$)↓
SEGSimulation	SD-Net	1.9284	0.3767
	DD-Net	**1.8873**	**0.3182**
SEGSalt	SD-Net	7.7420	0.7084
	DD-Net	**6.1111**	**0.6111**

Therefore, the incorporation of a second encoder effectively enhances the reliability of contour information for reconstructing the velocity model. This indicates that DD-Net and DD-Net70 are more adept at accurately capturing and representing the contours present in the data.

D. Does our network have the ability to generalize to different environments?

We conduct tests on four OpenFWI datasets using four trained DD-Net70 networks. To evaluate the generalization performance of the networks, we employ the LPIPS [34] metric. Fig. 14 shows the heatmap generated after testing the networks on the datasets. In the heatmap, darker boxes indicate better performance. Each network consistently produces the best predictions for its respective dataset. Therefore the boxes along the diagonal line in each row will always be the darkest.

In addition, the heatmap also reveals some intriguing observations. First, network trained by CurveVelA is the least sensitive to all types of data. The average difference between the maximum and minimum LPIPS is only 0.0791. Second, the most sensitive is FlatVelA, with an average maximum LPIPS difference of 0.4708. Finally, by comparing the three heatmaps in Fig. 14, DD-Net70 outperforms the other two networks on all values except the values in the "CFA" column. This is why Fig 14(a) is overall darker than Figs. 14(b) and 14(c). Remarkably, the DD-Net70 network trained by CurveFaultA has worse LPIPS than InversionNet when tested on CurveFaultA. But this network performs better than InversionNet on other datasets. Therefore, compared to InversionNet, the generalization ability of DD-Net70 trained through this dataset is even stronger than its fitting ability.

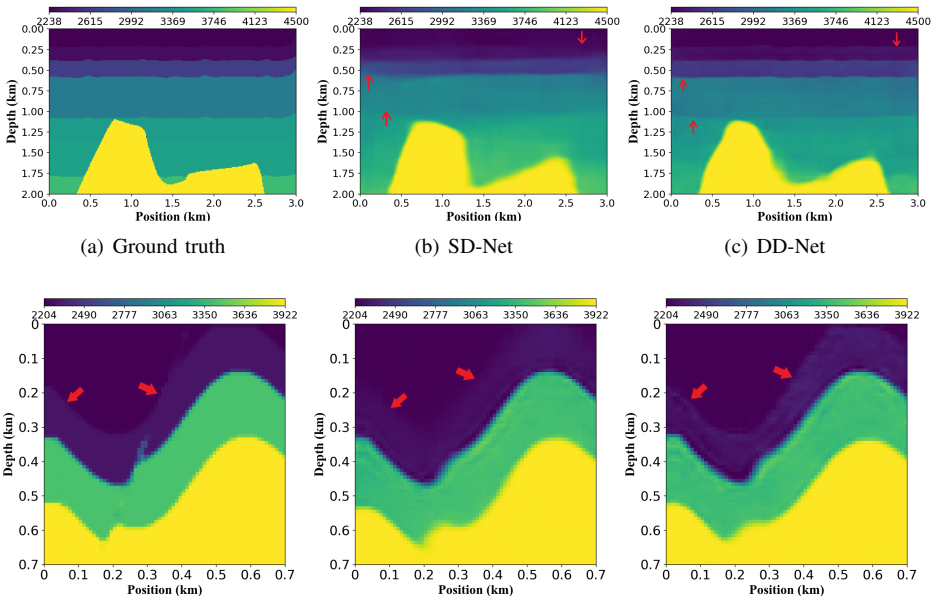
These findings confirm that DD-Net70 exhibits a higher level of adaptability and generalization during changes in the data distribution. From Figs. 14(a), 14(b) and 14(c), this advantage is the result of the joint action of the network structure and curriculum learning.

In addition, we also try to use DD-Net trained by SEG salt data to directly test the layered data in OpenFWI. First, we interpolate two randomly selected layered data to a scale consistent with the SEG salt data (201×301). Second, we put it under the same acquisition geometry conditions as the SEG salt data for forward simulation. Fig. 15 shows the inversion results of these two layered data on the DD-Net trained by the SEG salt data. The shallow interface inversion of the velocity model is in line with expectations, but the deep effect is not good.

After speculation, the network did not see formation structures without salt bodies during training. Therefore, without transfer learning, the network will always tend to invert a salt structure in predictions. This may also explain why there are high-velocity regions similar to salt bodies at the bottom of inversion results. Obviously, these high-velocity regions greatly interfere with the inversion results of the bottom. However it is undeniable that accurate shallow inversion result still proves that it has certain generalization ability.

E. What are the limitations of DD-Net and DD-Net70?

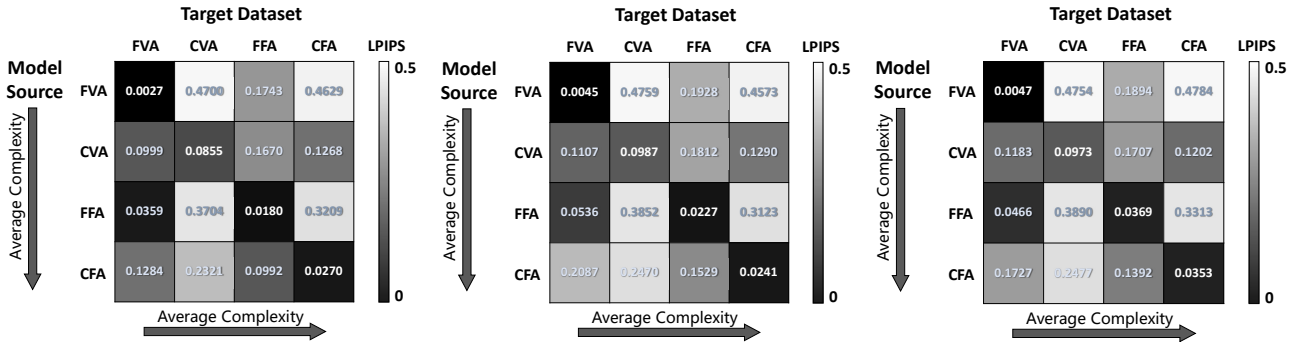
DD-Net and DD-Net70 still have some deficiencies that need to be improved. We divide the deficiencies into two parts



(a) Ground truth (b) SD-Net (c) DD-Net

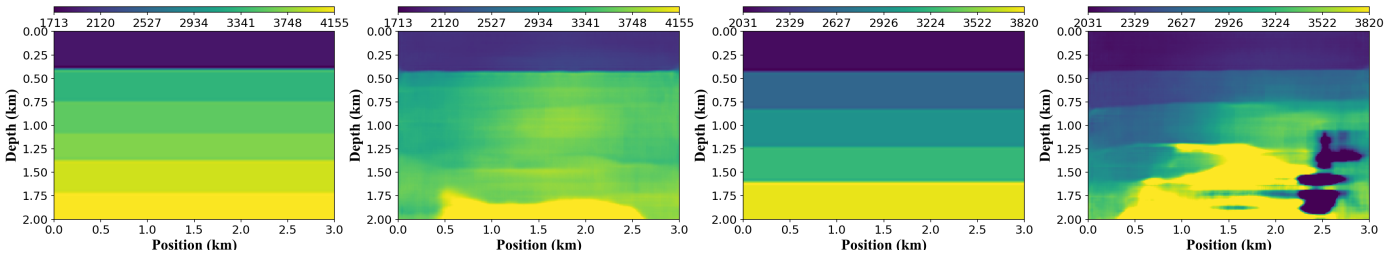
(d) Ground truth (e) SD-Net70 (f) DD-Net70

Fig. 13. Comparison example of inversion results caused by whether to use the second decoder.



(a) Heatmap of DD-Net70 (b) Heatmap of InversionNet (c) Heatmap of DD-Net70 without curriculum learning

Fig. 14. Heatmap of the generalization performance for the OpenFWI datasets. In the heatmap, "FVA" represents "FlatVelA", and the same naming convention applies to the other datasets. The horizontal axis of each box represents the source dataset used to train the network, while the vertical axis represents the dataset on which the network is tested. The dataset order follows the sorting outlined in Table II.



(a) Ground Truth (b) Prediction (c) Ground Truth (d) Prediction

Fig. 15. Out-of-sample examples trained on the SEG salt data and tested on the layered data.

based on whether they pertain to the edges or the interior of the formation.

First, there are abnormal shadow artifacts in the stratigraphic interface inverted by DD-Net70. It is not difficult to find from Figs. 10 and 11 that this phenomenon is more obvious on curved interfaces. This relief-like shading helps us spot interfaces with smaller velocity changes. But it can also lead to inaccurate velocity inversion near the interface. Second, scattered velocity instability spots can be observed in deeper formations, and these spots can have adverse effects on the

accurate inversion of wave propagation velocity within the formation. For example, the velocity in the deep layer of Fig. 10(f) and the shallow layer of Fig. 11(c) is not stable. In addition, this instability is more prevalent when there are fewer training samples. For example, in Figs. 6 and 7, the bottom of the velocity model inverted by DD-Net is all velocity unstable.

VII. Conclusion

In this study, a novel approach employing a dual-decoder architecture based on curriculum learning is presented. The dual decoder allows the network to consider edge structures when reconstructing velocity distribution, enhancing the accuracy of the inversion. The curriculum learning strategy guides the network to learn seismic data in a progressive manner, preventing overfitting and promoting generalization. The pre-network dimension reducer enables the network architecture to adapt to some low resolution seismic observation data. Our networks demonstrate superior performance on both the SEG and OpenFWI datasets, and ablation analysis further confirms the effectiveness of our proposed techniques. Moving forward, we plan to explore the following areas of improvement:

1) Feature embedding for seismic input data: We aim to investigate the integration of embedding features to capture information that may be challenging for the network to exploit. Previous studies have shown the potential of feature embedding [9, 10], but further exploration is needed to combine it with complex network architectures.
2) Enriching curriculum learning strategies: We intend to incorporate more prior knowledge [52] to design more comprehensive curriculum tasks. Additionally, we aim to develop automated difficulty measurers and training schedulers driven by this prior knowledge.
3) Advancing the integration of physics prior knowledge and network architecture: Physics-guided DL-FWI has the potential to correct irrational or unrealistic inversion results in the network [3]. While this field has garnered significant research interest [52, 53], it remains challenging due to its complexity. Hence we plan to further integrate our network architecture into this challenging and promising field.

Acknowledgments

This work is supported in part by Nanchong Municipal Government-Universities Scientific Cooperation Project (Nos. 23XNSYSX0062, 23XNSYSX0013), the Central Government Funds of Guiding Local Scientific and Technological Development (No. 2021ZYD0003), the National Natural Science Foundation of China (Nos. 62136002, 41674141) and the National Social Science Foundation of China (No. 22FZXB092).

References

[1] F. Yang and J. Ma, "Deep-learning inversion: a next generation seismic velocity-model building method," *Geophysics*, vol. 84, no. 4, pp. R583–R599, 2019.
[2] W. Zhu, K. Xu, E. Darve, B. Biondi, and G. C. Beroza, "Integrating deep neural networks with full-waveform inversion: Reparameterization, regularization, and uncertainty quantification," *Geophysics*, vol. 87, no. 1, pp. R93–R109, 2022.
[3] Y. Lin, J. Theiler, and B. Wohlberg, "Physics-guided data-driven seismic inversion: Recent progress and future opportunities in full-waveform inversion," *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 115–133, 2023.
[4] D. Datta and M. K. Sen, "Estimating a starting model for full-waveform inversion using a global optimization method," *Geophysics*, vol. 81, no. 4, pp. R211–R223, 2016.
[5] V. Kazei, O. Ovcharenko, P. Plotnitskii, D. Peter, X. Zhang, and T. Alkhalifah, "Mapping full seismic waveforms to vertical velocity profiles by deep learning," *Geophysics*, vol. 86, no. 5, pp. R711–R721, 2021.
[6] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, "Deep-learning tomography," *The Leading Edge*, vol. 37, no. 1, pp. 58–66, 2018.
[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[9] S. Li, B. Liu, Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, "Deep-learning inversion of seismic data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 2135–2149, 2020.
[10] B. Liu, S. Yang, Y. Ren, X. Xu, P. Jiang, and Y. Chen, "Deep-learning seismic full-waveform inversion for realistic structural models," *Geophysics*, vol. 86, no. 1, pp. R31–R44, 2021.
[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.
[12] W. Wang, F. Yang, and J. Ma, "Velocity model building with a modified fully convolutional network," in *SEG International Exposition and Annual Meeting*, 2018.
[13] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *ICML*, pp. 282–289, 2001.
[14] Y. Wu and Y. Lin, "InversionNet: An efficient and accurate data-driven full waveform inversion," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 419–433, 2019.
[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
[16] Z. Zhang and Y. Lin, "Data-driven seismic waveform inversion: A study on the robustness and generalization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 6900–6913, 2020.
[17] Z. Yu, D. Shen, Z. Jin, J. Huang, D. Cai, and X.-S. Hua, "Progressive transfer learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 1340–1348, 2022.
[18] W. Hu, Y. Jin, X. Wu, and J. Chen, "Progressive transfer learning for low-frequency data prediction in full-waveform inversion," *Geophysics*, vol. 86, no. 4, pp.

R369–R382, 2021.
[19] G. Fabien-Ouellet and R. Sarkar, "Seismic velocity estimation: A deep recurrent neural-network approach," *Geophysics*, vol. 85, no. 1, pp. U21–U29, 2020.
[20] J. Simon, G. Fabien-Ouellet, E. Gloaguen, and I. Khurjekar, "Hierarchical transfer learning for deep learning velocity model building," *Geophysics*, vol. 88, no. 1, pp. R79–R93, 2023.
[21] C. Deng, S. Feng, H. Wang, X. Zhang, P. Jin, Y. Feng, Q. Zeng, Y. Chen, and Y. Lin, "OpenFWI: Large-scale multi-structural benchmark datasets for full waveform inversion," in *NIPS*, vol. 35, 2022, pp. 6007–6020.
[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 234–241.
[23] B. Murugesan, K. Sarveswaran, S. M. Shankaranarayana, K. Ram, J. Joseph, and M. Sivaprakasam, "Psi-Net: Shape and boundary aware joint multi-task deep network for medical image segmentation," in *EMBC*, 2019, pp. 7223–7226.
[24] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.
[25] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine Learning*, vol. 28, pp. 7–39, 1997.
[26] Y. Li, J. Song, W. Lu, P. Monkam, and Y. Ao, "Multitask learning for super-resolution of seismic velocity model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 8022–8033, 2021.
[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41–48.
[28] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, "Curriculumnet: Weakly supervised learning from large-scale web images," in *ECCV*, 2018, pp. 135–150.
[29] Y. Tay, S. Wang, A. T. Luu, J. Fu, M. C. Phan, X. Yuan, J. Rao, S. C. Hui, and A. Zhang, "Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives," in *ACL*, 2019, pp. 4922–4931.
[30] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.
[31] S. Feng, Y. Lin, and B. Wohlberg, "Multiscale data-driven seismic full-waveform inversion with field data study," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.
[32] F. Aminzadeh, N. Burkhard, L. Nicoletis, F. Rocca, and K. Wyatt, "Seg/eaeg 3-d modeling project: 2nd update," *The Leading Edge*, vol. 13, no. 9, pp. 949–952, 1994.
[33] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.
[34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.
[35] B. Moseley, T. Nissen-Meyer, and A. Markham, "Fast approximate simulation of seismic waves with deep learning," *Solid Earth*, vol. 11, pp. 1527–1549, 2018.
[36] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *ISCA*, 2016, pp. 125–125.
[37] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016, pp. 1–16.
[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.
[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.
[40] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013, p. 3.
[41] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.
[42] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *CVPR*, 2010, pp. 2528–2535.
[43] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.
[44] H. Yu and S. Winkler, "Image complexity and spatial information," in *QoMEX*, 2013, pp. 12–17.
[45] L. Li, H. Cai, Y. Zhang, W. Lin, A. C. Kot, and X. Sun, "Sparse representation-based image quality index with adaptive sub-dictionaries," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3775–3786, 2016.
[46] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.
[47] M. Araya-Polo, T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, "Automated fault detection without seismic processing," *The Leading Edge*, vol. 36, no. 3, pp. 208–214, 2017.
[48] X. Wu, L. Liang, Y. Shi, and S. Fomel, "FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation," *Geophysics*, vol. 84, no. 3, pp. IM35–IM45, 2019.
[49] P. Moczo, J. O. Robertsson, and L. Eisner, "The finite-difference time-domain method for modeling of seismic wave propagation," *Advances in Geophysics*, vol. 48, pp. 421–516, 2007.
[50] B. Engquist and A. Majda, "Absorbing boundary conditions for numerical simulation of waves," *Proceedings of the National Academy of Sciences*, vol. 74, no. 5, pp. 1765–1766, 1977.
[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[52] X. Wu, J. Ma, and J. Z. Xu Si, "Sensing prior constraints in deep neural networks for solving exploration geophysical problems," *Proceedings of the National Academy of Sciences*, vol. 120, no. 23, p. e2219573120, 2023.
[53] M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions," *Journal of Geophysical Research: Solid Earth*, vol. 127, no. 5, p. e2021JB023120, 2022.

Xing-Yi Zhang was born in Deyang, Sichuan, China in 2000. He is a graduate student with Southwest Petroleum University, Chengdu, China. He has published a paper in Information Science. His research interest is full waveform inversion.

Fan Min was born in Chongqing, China in 1973. He received the M.S. and Ph.D. degrees from the School of Computer Science and Engineering, University of Electronics Science and Technology of China in 2000 and 2003, respectively. He visited the University of Vermont from 2008 to 2009. He is currently a professor with Southwest Petroleum University, China. He has published more than 200 refereed papers in various journals and conferences, including IEEE Transactions on Geoscience and Remote Sensing, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Geophysical Journal International, Pattern Recognition, Knowledge-Based Systems, Computers and Geosciences, and Information Sciences. His research has been supported by National Natural Science Foundation of China. His current research interests include machine learning and its applications such as seismic inversion, recommender systems, and microbiology.

Shulin Pan was born in Dezhou, Shandong, China in 1979. He received the M.S. and Ph.D. degrees in Geophysical Prospecting and Information Technology from Chengdu University of Technology in 2005 and 2008. Since 2011, he is a Professor with Southwest Petroleum University, sichuan, china. His research interest includes automatic first-arrival time picking, Seismic data Interpolation and AVAZ (amplitude versus incident and azimuthal angle) inversion. His research has been supported by Science and Technology Cooperation Project of the CNPC-SWPU Innovation Alliance (2020CX020000), Open Projects Fund of the State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation (grant no. PLN201733), and National Natural Science Foundation of China (grant nos. 41204101, 41704134).

Qiong Xu was born in Jinzhong, Shanxi, China in 1991. She received B.S. and M.S. degrees from Xidian University in 2014 and 2017, respectively. She is currently pursuing a Ph.D. at Southwest Petroleum University. She research interest is seismic inversion.

Xue-Yang Min is a Ph.D. candidate with School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include multi-label learning and forward modeling.

Guojie Song received the M.S. and Ph.D. degrees from the Department of mathematical science, Tsinghua University, Beijing, China, in 2008 and 2011, respectively. He is currently a professor with Southwest Petroleum University, Chengdu. He has published more than 50 refereed papers in various journals and conferences, including the Geophysics, BSSA, et. al. His current research interests include forward modeling and inversion of seismic wave, deep learning.

Ke Wang was born in Neijiang, Sichuan, China in 1985. He received B.S. and M.S. degrees from College of Optoelectronic Engineering, Chongqing University in 2008 and 2011, respectively. He is currently the CEO of Sichuan Egretech Technology Co., Ltd. His research interest is seismic inversion.