Wave-dynamics simulation using deep neural networks

Weiqiang Zhu
Stanford University
zhuwq@stanford.edu

Yixiao Sheng
Stanford University
yixiao2@stanford.edu

Yi Sun
Stanford University
ysun4@stanford.edu

Abstract

Numerical simulation is the dominant method to analyze wave-dynamics. Few neural network methods have been tried on wave-dynamics simulation before. In this project, we have trained a neural network to test if it could learn wave-dynamics only by watching wave propagation sequences without explicit physical equations. We firstly generate a dataset of seismic waves using homogeneous media. The results show that the neural network can successfully predict the propagation of wavefront and keep good sharpness. Then we generate another dataset using complex media with random velocity anomalies, which contains much more complex wave-dynamics phenomena like reflection and diffraction. The prediction results prove that the neural network is also able to predict the complex signals based on the velocity model. Our results demonstrate that the neural network method may be used as an alternative method for wave-dynamics simulation.

1. Introduction

When we see an apple falling from a tree, or a ball throwing at us, we can predict where the apple or the ball would go in just seconds. We do not form physics equations and solve it in our mind to get the answer. Instead, our prediction is based on our instinct and experience. But can a machine learn the basic physical instincts just like humans? Can neural networks make long-term predictions based on observations without explicitly solving the underlying physical equations?

Recently with the fast development of deep neural network methods, many researchers in physics begin to experiment with using neural networks to simulate physical processes. It is a challenging idea to train a neural network to mimic humans' ability to observe and learn physical laws. By now, numerical simulation is still the most accurate and effective way to understand the complex physical phenomenon. As traditional numerical simulation usually requires extensive computing resources, neural networks may provide a less accurate, but much faster and more efficient solution. It is also very crucial for model-based decision-making and planning of robots in complex physical environments.

In this project, we are trying to apply deep neural networks on wave-dynamics simulation to test if the neural network can learn and predict the dynamics of wave propagations. We have used a multi-scale convolutional neural network to map the input several frames of wavefields to the output frame. The neural network is trained by 'watching' tons of seismic waves propagating through different media. No physical equations or numerical approximation methods are fed into the neural network. The training data and test data are generating using a numerical simulation code for seismic waves. The results show that the neural network can predict the following sequence of wave propagation given only the first several frames. It also learns the diffraction and reflections given complex media with velocity anomalies inside.

2. Related Work

Watters [13] introduced Visual Interaction Network, a general-purpose model for learning the dynamics of a physical system from raw visual observations. The model consists of a perceptual front-end based on convolutional neural networks and a dynamics predictor based on interaction networks, which can produce a predicted physical trajectory of arbitrary length.

Carleo and Troyer [1] applied machine learning on the simulation of quantum many-body systems by using an artificial neural network to represent the wave function of a quantum many-body system and making the neural network learn what the ground state (or dynamics) of the system is.

Enrhardt *et al*. [3] investigated the effectiveness of neural networks for end-to-end long-term prediction of mechanical phenomena. The network is also able to generate a distribution of outcomes to capture the inherent uncertainty in the data.

DeVires *et al*. [2] trained a deep neural network to learn a computationally efficient representation of viscoelastic solutions. The machine learning approach accelerates viscoelastic calculations by more than 50,000%, which enables the modeling of geometrically complex faults over thousands of earthquake cycles.

Lerer *et al*. [9] explored the ability of deep feed-forward models to learn the physical behavior of the world by playing wooden blocks toy. They trained large convolutional network models, which are able to accurately predict the outcome of randomized initial stability, as well as estimating the block trajectories.

Tomposon *et al*. [12] trained a convolutional network from a training set of simulations to realize real-time and highly realistic simulations of fluid and smoke.

Guo *et al*. [5] demonstrated that convolutional neural networks could estimate the velocity field two orders of magnitude faster than a GPU-accelerated CFD solver and four orders of magnitude faster than a CPU-based CFD solver at the cost of a low error rate, which provides immediate feedback for real-time aerodynamics related design.

Ladick *et al*. [7] proposed a novel machine learning based approach to approximate the behavior of particles observed in the large training set of simulations obtained using a traditional solver. The GPU implementation led to a speed-up of one to three orders of magnitude compared to the state-of-the-art position-based fluid solver based on Navier-Stokes equations.

As far as we know, using deep neural networks to do simulation is still a relatively burgeoning area. Our work may be the first one to use deep neural networks for wave-dynamics simulation.

3. Methods

3.1. Network Architecture

The structure of our neural network is depicted in Figure 1. We used the method in video prediction [10][1] to build a deep neural network [8] to learn an end-to-end mapping that predicts the wavefields at following continuous time-steps from only a few initial time-steps. In this project, we will predict the next time-step wavefield based previous four time-steps. The neural network architecture mainly consists of two components:

1. Multi-scale encoding and decoding convolutional layers (Figure 1a). It essentially is four mini ConvNets with input and output sizes of $n \times n, \frac{n}{2} \times \frac{n}{2}, \frac{n}{2^2} \times \frac{n}{2^2}$ or $\frac{n}{2^3} \times \frac{n}{2^3}$, where n is the dimension of the input wavefields. The parameters of these ConvNets are shown in Table 1. For example, if the dimensions of wavefields are (32×32), the input and output sizes of the four ConvNets are $32 \times 32, 16 \times 16, 8 \times 8$ and 4×4. The four ConvNets are connected successively from small scale to large scale. To be more specific, the ConvNet at scale (8×8) takes both the down-sampled raw wavefields (32×32 down-sampled to 8×8) and the output of the previous ConvNet (4×4), which are up-sampled to (8×8) through interpolation, as the inputs and predicts the next time-step (8×8) wavefield. The main advantage of this multi-scale structure is that it skips the pooling/unpooling pairs and could preserve the high-frequency information[10]. As waves usually have many characterizing frequencies, it is imperative for wave-dynamics simulation to have accurate frequency information and to keep sharpness of wavefront.

2. Generative adversarial networks (GAN)[4] (Figure 1b). GAN is known to be effective in producing realistic images. In addition to the multi-scale generative model above, a discriminative model is trained simultaneously. For each scale, a discriminative network is trained to take both the true down-sampled input wavefield and the predicted wavefield as input to predict True/False. The parameters of the discriminate network are shown in Table 1. These two models are competing against each other, in a way that the generative model tries to 'foul' the discriminative model by generating more realistic wavefields, while the discriminative model tries to be smarter to tell if the generative model is 'cheating'.

For wave-dynamics simulation, we need to consider the background velocity model too, which will control waves' propagation speed, direction and path. It also produces reflection, refraction and diffraction. For both generative model and discriminative model, we have included the velocity model as three indecent channels, so that we can test if the neural network is able to learn the complex wave-dynamics phenomena based on physical models.

During training, each sample consists wavefields of five continuous time steps: four input wavefields and one actual output wavefield. During testing, we use a recessive way to generate long time steps predictions. The first predicted wavefield is based on the four input wavefields, then we drop wavefield of the first time-step and append the newly predicted wavefield as the last input wavefield. The updated sample is used to generate next wavefield. This process is repeated to generate a long continuous sequence of wavefields.

3.2. Loss Functions

The loss function used to evaluate how realistic the predicted wavefield is compared with the true wavefield. The loss function for our generative network is:

$$\mathcal{L}^G_{total}(Y,\hat{Y}) = \lambda_{L2}\mathcal{L}_2(Y,\hat{Y}) + \lambda_{GDL}\mathcal{L}_{GDL}(Y,\hat{Y}) + \lambda_G\mathcal{L}^G(Y,\hat{Y})$$

$$\mathcal{L}_2(Y,\hat{Y}) = \frac{1}{N}||Y - \hat{Y}||_2^2$$

$$\mathcal{L}_{GDL}(Y,\hat{Y}) = \frac{1}{N}||\nabla Y - \nabla\hat{Y}||_2^2$$

[1] https://github.com/dyelax/Adversarial_Video_Generation

	Generative Network	Discriminative Network
Scale 1	$3 \times 3 \times 20 \times 128$	$3 \times 3 \times 5 \times 64$
	$3 \times 3 \times 128 \times 256$	
	$3 \times 3 \times 256 \times 128$	FC: $(512, 256, 1)$
	$3 \times 3 \times 128 \times 5$	
Scale 2	$5 \times 5 \times 25 \times 128$	$3 \times 3 \times 5 \times 64$
	$3 \times 3 \times 128 \times 256$	$3 \times 3 \times 64 \times 128$
	$3 \times 3 \times 256 \times 128$	$3 \times 3 \times 128 \times 128$
	$3 \times 3 \times 128 \times 5$	FC: $(1024, 512, 1)$
Scale 3	$5 \times 5 \times 25 \times 128$	$5 \times 5 \times 5 \times 128$
	$3 \times 3 \times 128 \times 256$	
	$3 \times 3 \times 256 \times 512$	$5 \times 5 \times 128 \times 256$
	$3 \times 3 \times 512 \times 256$	
	$3 \times 3 \times 256 \times 128$	$5 \times 5 \times 256 \times 256$
	$3 \times 3 \times 128 \times 5$	FC: $(1024, 512, 1)$
Scale 4	$7 \times 7 \times 25 \times 128$	$7 \times 7 \times 5 \times 128$
	$5 \times 5 \times 128 \times 256$	$7 \times 7 \times 128 \times 256$
	$5 \times 5 \times 256 \times 512$	$5 \times 5 \times 256 \times 512$
	$5 \times 5 \times 512 \times 256$	$5 \times 5 \times 512 \times 128$
	$5 \times 5 \times 256 \times 128$	
	$7 \times 7 \times 128 \times 5$	FC: $(1024, 512, 1)$

Table 1: Multi-scale Network Parameters

$$\mathcal{L}^G(Y, \hat{Y}) = -\frac{1}{N} \sum_i \log(\mathbf{D}(\hat{Y})_i)$$

where Y is the true wavefield, $\hat{Y}$ is the predicted wavefield, (λ_{L2}, λ_{GDL}, λ_G) are weights for different loss functions. N is the number of data points in a wavefield. $\mathbf{D}$ is the generative network.

The loss function of the generative network is:

$$\mathcal{L}^D(Y, \hat{Y}) = -\frac{1}{N} \sum_i (\log(\mathbf{D}(Y_i)) + \log(1 - \mathbf{D}(\hat{Y}_i)))$$

The Euclidean loss $\mathcal{L}_2$ is the most straightforward way to measure the difference between two wavefields. Minimizing $\mathcal{L}_2$ is to make sure the amplitude of predicted wavefield is close to ground truth.

The spatial gradient loss $\mathcal{L}_{GDL}$ is a measurement of the sharpness of wavefront. By minimizing the gradient difference loss, we can preserve the sharpness of the generated wave field and prevent blurry wavefronts.

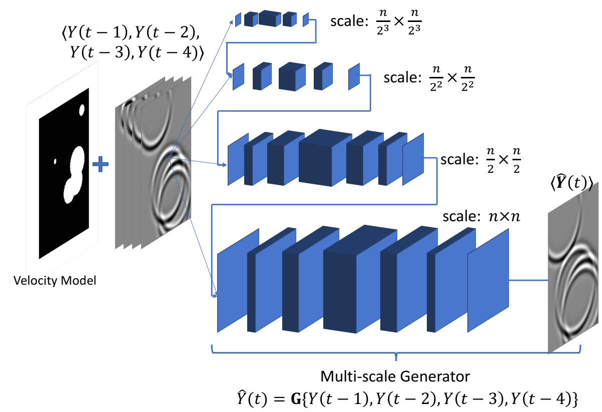
By combining these losses, we hope to improve the sharpness of wavefront and keep frequency information, which is vital for wave-dynamics simulation.

4. Dataset and Features

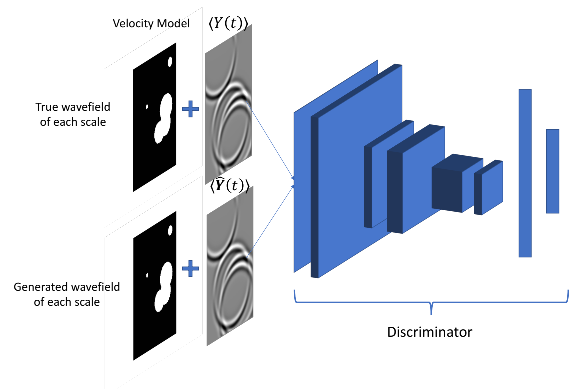
We use a 2-D seismic wave numerical simulator [6][2] to randomly generate wave fields both for training and testing datasets. The parameters we change during simulations are: source number, source location, source frequency, P-wave velocity, S-wave velocity and material density. By combining different choices of these parameters, our simulations can capture different seismic wave phenomenon: reflection, diffraction, P-to-S or S-to-P conversion, heterogeneous propagation velocity *et al.* [11]. The simulation domain ($32km \times 42km$) is discretized into 160×210 grids. The displacements in both x and y directions are recorded at continuous time steps ($dt = 0.075s$), serving as the input data for our neural network.

We have tested two datasets: one of homogeneous media and the other of complex media with velocity anomalies. For the dataset of homogeneous media, we randomly pick the source numbers (1∼20 in our case) as well as their locations but fix all the other parameters. For the dataset of complex media with velocity anomalies, we only use

[2] http://geodynamics.org/cig/software/seismic_cpml/



(a) Generative Network to predict next-step wavefield



(b) Discriminate Network

Figure 1: Network architecture for wave-dynamics simulation

one random source and randomly decide the numbers, locations and radius of the circular anomalies in the media. The anomalies are added to material density. They can also be added to the velocity of **P** wave or velocity of **S** wave. Because these three media parameters have similar effects, we think if the neural network is able to learn reflection or diffraction from density anomalies, it can also lean wave-dynamics based on the velocity of **P** and **S** waves.

For both datasets, we repeatedly generated 72 test samples and 720 training samples. Each sample consists of 85 time-steps. Figure 2 shows one example of the synthetic wavefields in complex media. The white circles are density anomalies with $5000km/m^3$ while the background density is $3000km/m^3$. We can observe very complicated signals from reflection and diffraction in this example. Then the numerically generated samples of 85 time-steps are randomly clipped into small samples of only five time-steps: four as inputs and one as true output. The input data include five channels: $(U_x, U_y, V_p, V_s, \rho)$ which are x component displacement, y component displacement, velocity of **P** wave, velocity of **S** wave, material density.

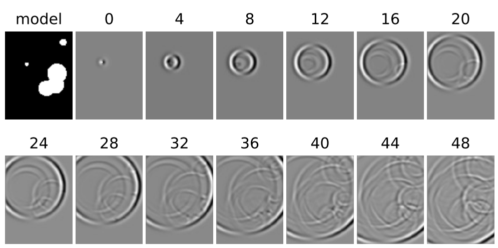


Figure 2: Training example of synthetic wavefields in complex media produced by the numerical simulator

5. Results

We have done three experiments using the two numerical generated datasets: 1.homogeneous media without GAN; 2.complex media without GAN; 3.complex media with GAN. Two evaluation criteria, Peak Signal-to-Noise Ratio (PSNR) and Sharpness Difference, are used to evaluate the quality of our generated wavefield.

$$\text{PSNR}(Y, \hat{Y}) = 10 \log_{10} \frac{1}{\frac{1}{N}||Y - \hat{Y}||_2^2}$$

$$\text{Sharpness}(Y, \hat{Y}) = 10 \log_{10} \frac{1}{\frac{1}{N}|\nabla Y - \nabla \hat{Y}|}$$

5.1. Homogeneous media

In this experiment, all channels of V_p, V_s, ρ of training data are zeros. GAN is not used ($\lambda_G = 0$). The two loss weights (λ_{L2} and λ_{GDL}) are set to one:

$$\mathcal{L}^G_{total}(Y, \hat{Y}) = \mathcal{L}_2(Y, \hat{Y}) + \mathcal{L}_{GDL}(Y, \hat{Y})$$

Figure 3 shows one test example of predicted wavefield of U_x. Images in the first row are the true wavefields from a numerical simulation. Images in the second row are the wavefield generated by our neural network. Step $0 \sim 3$ are the four input wavefields, the rest steps are generated based on previous four continues steps. The newly generated step will be used for the next predictions recursively.

The image quality is quite good for homogeneous media. The generated wavefronts are clear and sharp.There are few noticeable differences between the true wavefields and generated wavefields at early time-steps. At very late time-steps, the wavefronts start to suffer from blurry effect.

As we used absorbing boundary condition in the numerical generator, the wavefront will be absorbed on the boundary. There are no reflections from the boundary in the training dataset. The neural network also learns how to deal with the boundary and generate no reflections from the boundary. The wavefront disappears after it hit the boundary.

5.2. Complex media

In this experiment, we train our model using the dataset with complex media. There are anomalies in the density channel (ρ) to cause reflections and diffractions. This example could further test if the neural network can learn complex wave-dynamics. The two loss weights (λ_{L2} and λ_{GDL}) are still set to one.

Firstly, We train a model without GAN. The testing results are shown in Figure 4 and Figure 5. Same as the example in homogeneous media, the major incident wavefront is clear and sharp through all time steps. Moreover, the reflections and diffractions after the incident wavefront are also generated by the neural network. Comparing the generated wavefields and the true ones in Figure 4, we can see the complex pattern of reflected or diffracted waves are correctly predicted. But the amplitudes of the weak signals are predicted to be too large at late time-steps.

The change of loss of generative network is shown in Figure 6. With the decrease of training loss, the two evaluation criteria (PSNR and sharpness difference) (Figure 7b and Figure 7a) gradually improve. Figure 8 quantitatively measures the deterioration of generating quality with prediction steps. PSNR and sharpness difference decrease to about half of its origin value after 20 predictions.

In some test examples, We also observe some undesired reflection signals from the boundary for complex media. Figure 5 is one test example showing very clear abnormal reflection signals. The true reflection signals from density anomalies are correctly predicted, but two wavefronts start to grow after step 8 from the left boundary.

Secondly, we try to incorporate GAN into the model by increasing λ_G and training the discriminative network simultaneously. The training of GAN model is not successful. We have tested $\lambda_G = 0.2, 0.4, 0.6, 0.8, 1.0$ and different learning rates for the discriminative model. The generative loss (Figure 9a) and discriminate loss (Figure 9b) both decrease at the early iterations. But the generative loss will explode if we continue training for longer time (Figure 9c). We also don't observe any increase in PSNR and Sharpness difference on the test set.

6. Discussion

The predicted results prove that the neural network can successfully learn from physical simulation and generate realistic wavefields of following time-steps in both homogeneous and complex media. The predictions in homogeneous media show that the neural network not only accurately learns the propagation of the seismic wave but also learns the absorbing on the boundary. The test in complex media demonstrates that the neural network is also able to predict reflections and diffractions based on velocity model. This makes the neural network fully cable to predict wave-dynamics simulations with different velocity models.

However, we also observe some problems that limit the accuracy and robustness of results:

1. Prediction quality decreases as time step increases. This is a problem of recursive prediction, as the errors will accumulate through time. How to ensure long time-step stability is still challenging. RNN or LSTM method may help to consider long time-step dependence and improve predictions further.

2. Abnormal reflections from boundaries. In some examples with complex media, we find some reflections on the boundary which don't exist in the actual simulation data. One possible cause could be the model misunderstand the boundary as obstacles and reflects. Further analysis is needed to find the source of these abnormal reflections and remove them from prediction.

3. GAN gives little improvement. The improvement in GAN is not as large as we thought. The training of GAN is much harder for the multi-scale network structure. It is tricky to balance the learning of different scales. Different ratios of learning rate between generator and discriminator and different ratios between three losses are tested. Further fine-tuning the hyper-parameters is needed.

One of our origin goals is to use neural network to provide a fast but less accurate solution for seismic simulation so that computational intensive inversion methods could become practical. But the prediction speed of our current network is not distinctly faster than the numerical simulation code used. There is also a trade-off between complexity and speed of neural network architecture too. However, the neural network method learns from raw observations. It does not depend on the complexity of physical equations behind the phenomenon and can be applied to other different physical processes as well. If given a more complex problem, like fluid simulation[12], which is much slower using traditional numerical simulation, the advantage of neural network method may become more significant.

7. Conclusion

We have successfully trained a neural network to learn and predict wave-dynamics. The network uses a multi-scale structure to predict the next time-step wavefield based on previous four time-steps. Beside loss of wave amplitude difference, sharpness difference is added into the total loss function of the generative network to keep wavefront clear and sharp. We have also added GAN to test if it could improve the prediction quality.

The experiment results turn out to be promising. In homogeneous media, the network can predict the propagation of wavefronts inside the media and the absorbing on the boundary with high accuracy. In complex media, the network is further proved to be able to predict complex reflections and diffractions based on velocity model. But the training using GAN still needs future improvement. The training of GAN is not stable at current state.

Our work has proven the neural network could become an alternative method for wave-dynamics simulations beside numerical simulation. The neural network method learns from raw data and does not rely on explicit physical equations, so the same network structures may be used learn and predict other physical phenomena too.

Acknowledgements

This work has not been submitted to any conference or journal. It has not ever been used as a project for other classes. The numerical simulation code for 2D seismic wave simulation is modified based on open-source code "seismic cpml" (`http://geodynamics.org/cig/software/seismic_cpml/`). The neural network code for wave-dynamics prediction is modified based on code "adversarial video generation" (`https://github.com/dyelax/Adversarial_Video_Generation`). We would like to thank these two projects for sharing their codes.

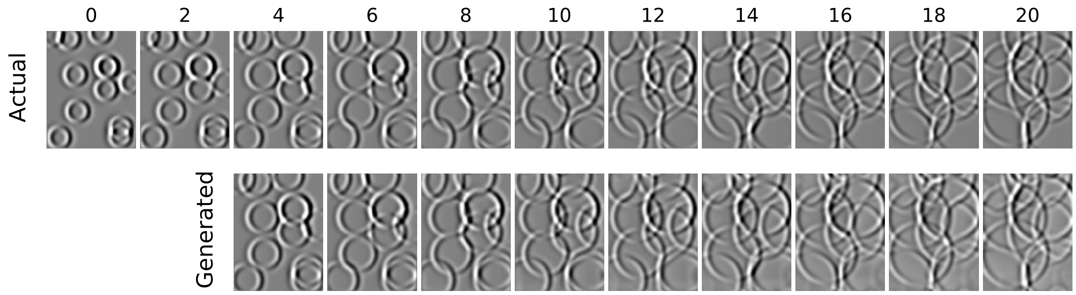


Figure 3: Test example of generated wave propagation in homogeneous media

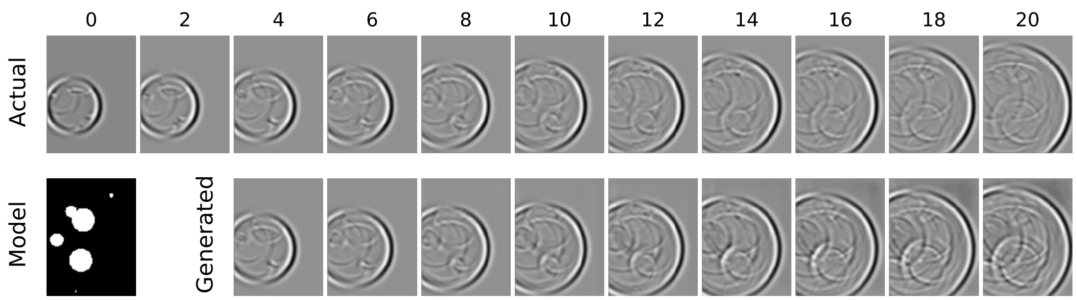


Figure 4: Test example of generated wave propagation in complex media

References

[1] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[2] P. R. DeVries, T. B. Thompson, and B. J. Meade. Enabling large-scale viscoelastic calculations via neural network acceleration. pages 2662–2669, 2017.

[3] S. Ehrhardt, A. Monszpart, N. J. Mitra, and A. Vedaldi. Learning A Physical Long-term Predictor. 2017.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.

[5] X. Guo, W. LI, and F. Iorio. Convolutional Neural Networks for Steady Flow Approximation. *Kdd2016*, 2016.

[6] D. Komatitsch and R. Martin. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):SM155–SM167, 2007.

[7] L. u. Ladick, E. Zurich, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross. Data-driven Fluid Simulations using Regression Forests. *ACM Trans. Graph. Article*, 34(9):2–5, 2015.

[8] Y. Lecun and Y. Bengio. Convolutional Networks for Images, Speech, and Time-Series. *The handbook of brain theory and neural networks*, (May 2013):255–258, 1995.

[9] A. Lerer, S. Gross, and R. Fergus. Learning Physical Intuition of Block Towers by Example. *Arxiv*, 2016.

[10] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *Iclr*, (2015):1–14, 2015.

[11] P. M. Shearer. *Introduction to seismology*. Cambridge University Press, 2009.

[12] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating Eulerian Fluid Simulation With Convolutional Networks. 2016.

[13] N. Watters, A. Tacchetti, T. Weber, R. Pascanu, P. Battaglia, and D. Zoran. Visual Interaction Networks. pages 1–14, 2017.

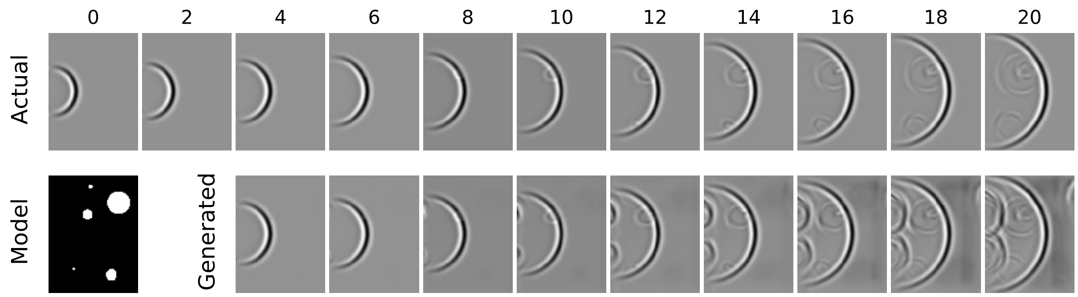


Figure 5: Test example with undesired reflections from boundary

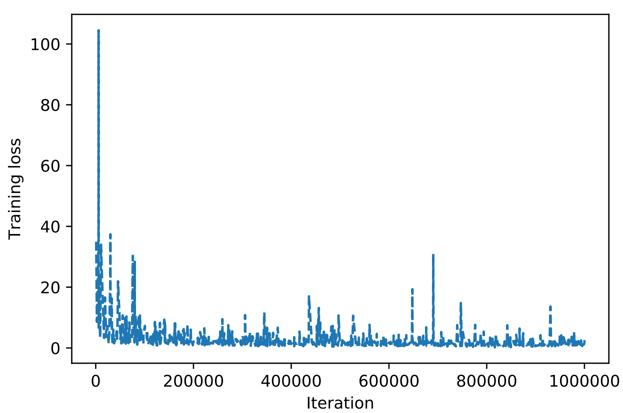
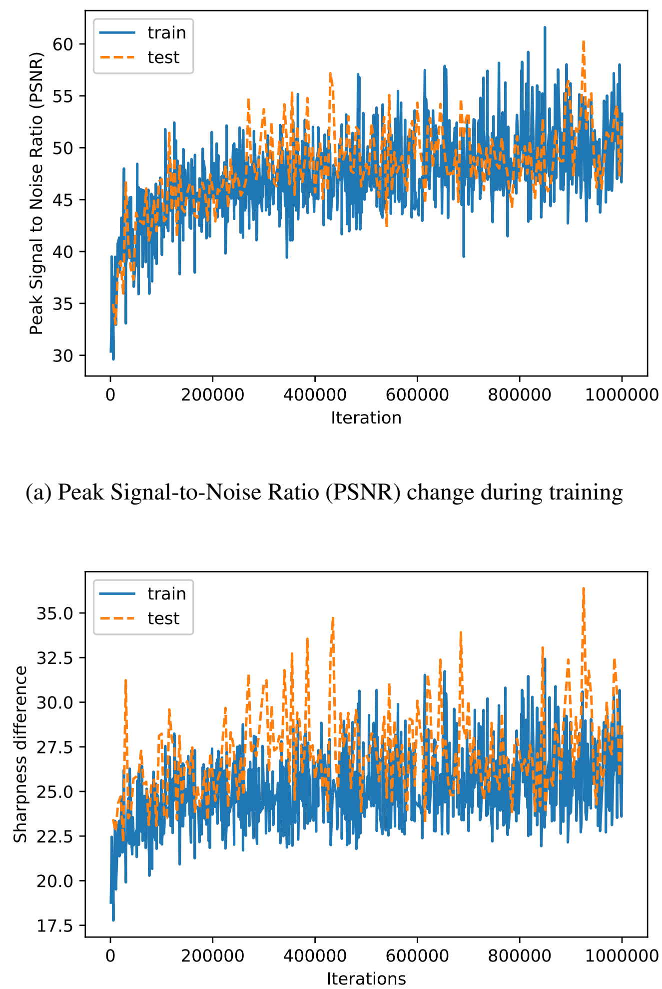
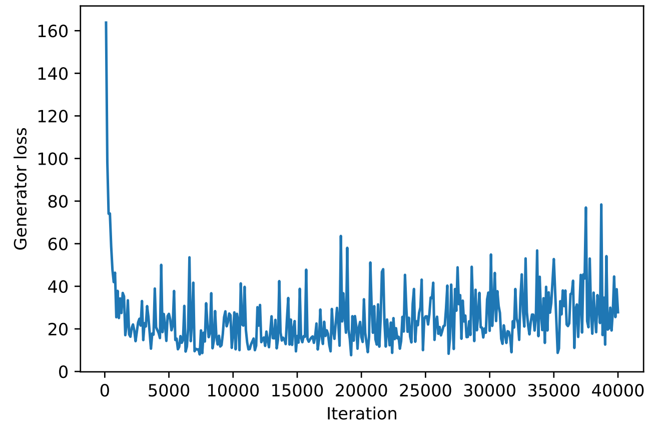


Figure 6: Training loss change of generative network without GAN



(a) Peak Signal-to-Noise Ratio (PSNR) change during training

(b) Sharpness difference change during training



(a) GAN loss of generative model in early time

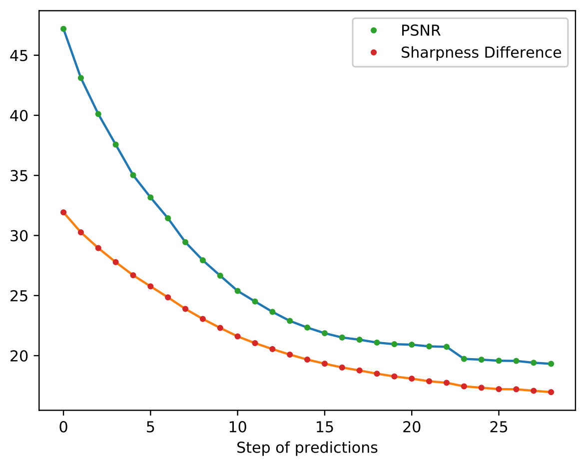
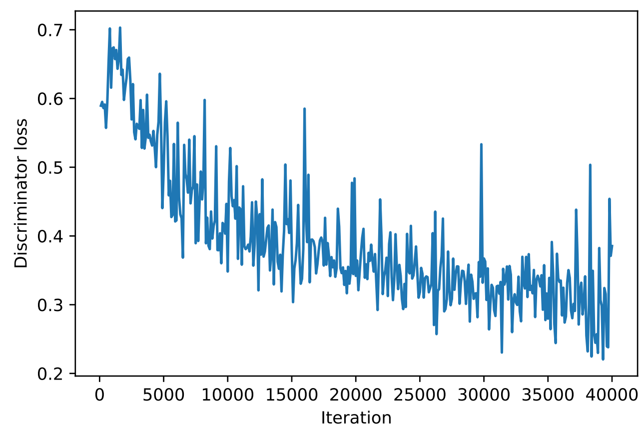
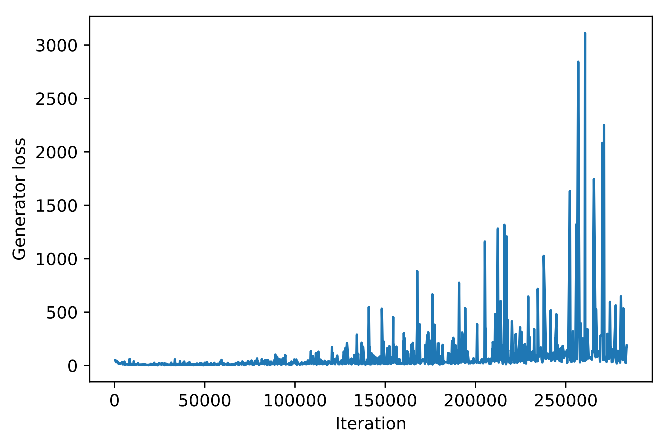


Figure 8: PSNR and sharpness difference changes during recursive prediction



(b) GAN loss of discriminative model in early time



(c) GAN loss of generative model in late time