Report for semester project

Zhou Haitao

Supervisor: Dr. Carlo Gigli, Prof. Demetri Psaltis

Introduction

MaxwellNet is a novel scheme to train a DNN that solves Maxwell's equations speedily and accurately without relying on other computational electromagnetic solvers. The approach is to train a DNN using the residual of Maxwell's equations as the physics-driven loss function for a network that finds the electric field given the spatial distribution of the material property. It can be used to solve forward problems and do inverse design.[1]

Use MaxwellNet to solve problems

Predicting Mie scattering with MaxwellNet

2D case

Problem descrption

There is a silicon rod with circular section in the xz plane and infinitely extended along the y direction. We label the radius with r and set the refractive index of the material $n_m = 3.5$. We want to study how the scattering efficiency differs with different r when the cylider is excited with an external plane wave $E_i = E_0\, exp(jk_0 z)\hat{x}$ (TM-polarization), $\lambda_0 = 1550nm, n_b = 1$.

Experiment design

To calculate scattering power, we need to use the following equation

$$W_{sca} = \oint_{\Sigma} < \mathbf{S}_{sca} > \cdot \mathbf{n} d\sigma = \frac{1}{2}\oint_{\Sigma} \mathrm{Re}(\mathbf{E}_S \times \mathbf{H}_S^*) \cdot \mathbf{n} d\sigma$$

We do research on 2D TM polarization case, we can simplify the equation based on it. The $d\sigma$ in the equation can be simplified to dl due to 2D limitation. TM polarization means Es only contains x and z components, and Hs only contains y component, so Es = [Esx, 0, Esz], Hs = [0, Esy, 0].

The simplied Equation is

$$\frac{1}{2}\oint[-Re(Esz * Hsy), \qquad 0, \qquad Re(Esx * Hsy)]\,.\,\vec{n}dl$$

If we use a square window to do calculation, $\vec{n}$ of left side is [0, 0, -1], $\vec{n}$ of upper side is [1, 0, 0], $\vec{n}$ of bottom side is [-1, 0, 0], $\vec{n}$ of right side is [0, 0, 1]. We can get Es directly from the output of a well trained MaxwellNet. Hs can be calculated following the equation

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = j\omega\mu H_y$$

Result and analysis

Train dataset contains 1000 samples whose radius is randomly from 0.1um to 0.4um are generated to form train dataset. dpl is 150 (The distance per wavelength can cover 150 pixels). pml thickness (Perfectly Matched Layer) is 75. Final average train loss over these 1000 samples stops at 0.3.

The trained MaxwellNet has the ability to predict scattering field E_S when radius of the silicon rod lies between 0.1um and 0.4um, so we can calculate corresponding scattering power and plot scattering efficiency curve as the following

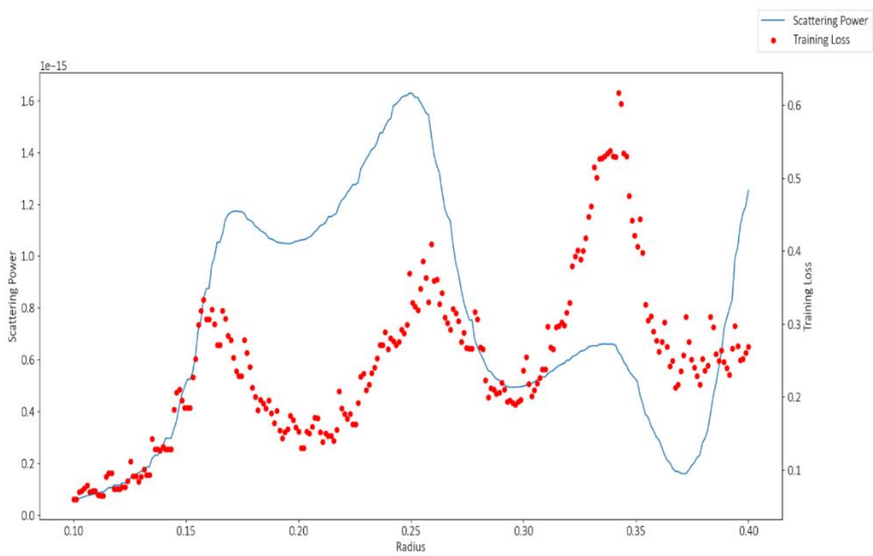

Figure 1: Scattering efficiency curve based on MaxwellNet

Use COMSOL to get the ground truth plot

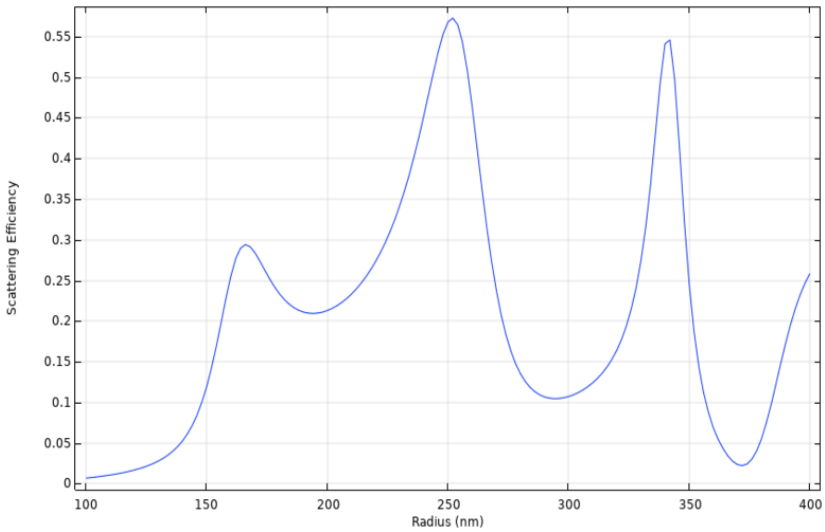

Figure 2: Ground truth scattering efficiency curve based on COMSOL

The scattering efficiency curve based on MaxwellNet doesn't match the ground truth well. The relative height of peaks has some errrors.

The errors may come from 3 aspects:

1.The implementation of scattering power calculation

2.The implementation of loss function to guide the train of MaxwellNet

3.The network architecture

After analysis, the errors are most likely to come from the poor performance of network architecture, because the overall loss of scattering efficiency is about 0.3 which is a lot higher than the requirement loss.

The compare of three plot with different loss is as the following. The first one is ground truth of total field when the radius of the rod is 250nm. The second one is MaxwellNet output with loss=0.12. The third one is MaxwellNet output with loss=0.0156. It can again show that 0.3 is not a low enough loss to get an accurate result.

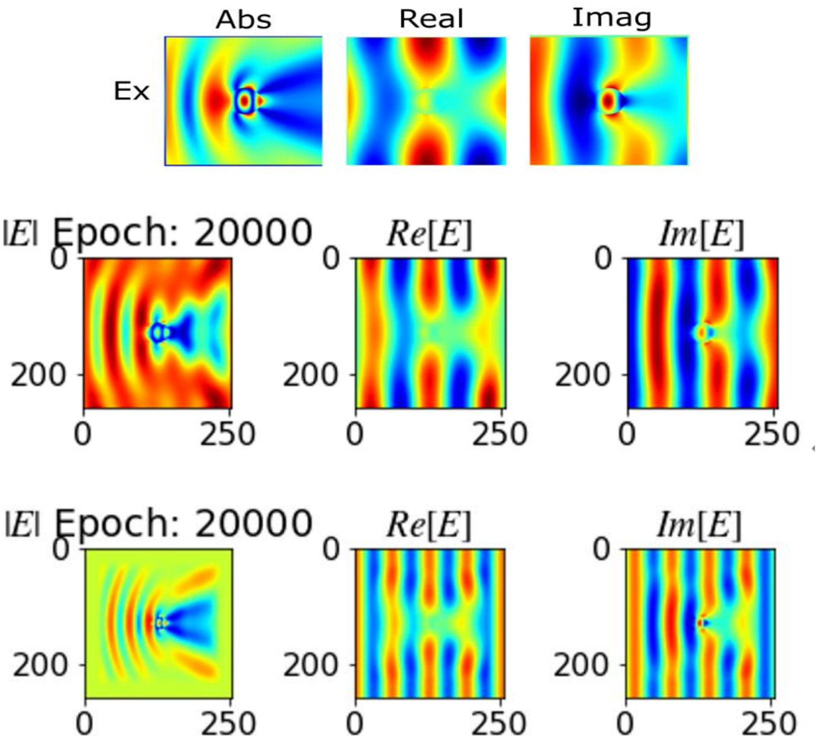

Figure 3: Ground truth, MaxwellNet output with loss=0.12, MaxwellNet output with loss=0.0156 are listed from the top to the bottom.

During the experiment, some interesting phenomenons are observed.

$$\text{dpl} \downarrow \begin{cases} good: \ loss\ can\ saturate\ at\ a\ lower\ value \\ bad: more\ details\ are\ lost \end{cases}$$

$$\text{pmlthickness} \uparrow \begin{cases} good: quality\ of\ field\ improves \\ bad: pixels\ may\ be\ not\ enough \end{cases}$$

A trade-off choice should be made for the best performance of MaxwellNet.

3D case

The method for 3D case is quite similar as that of 2D case, and the result is as the following

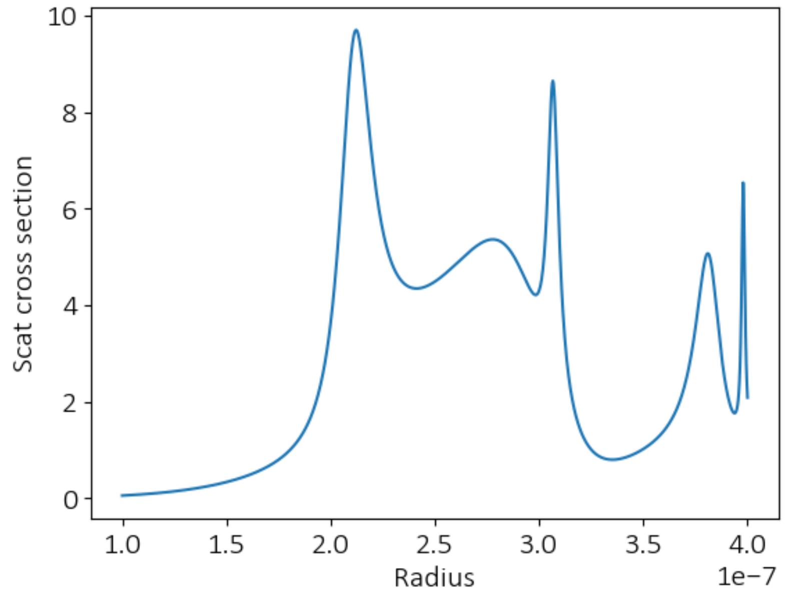

Figure 4: Scattering efficiency of a silicon sphere for TM polarization based on COMSOL

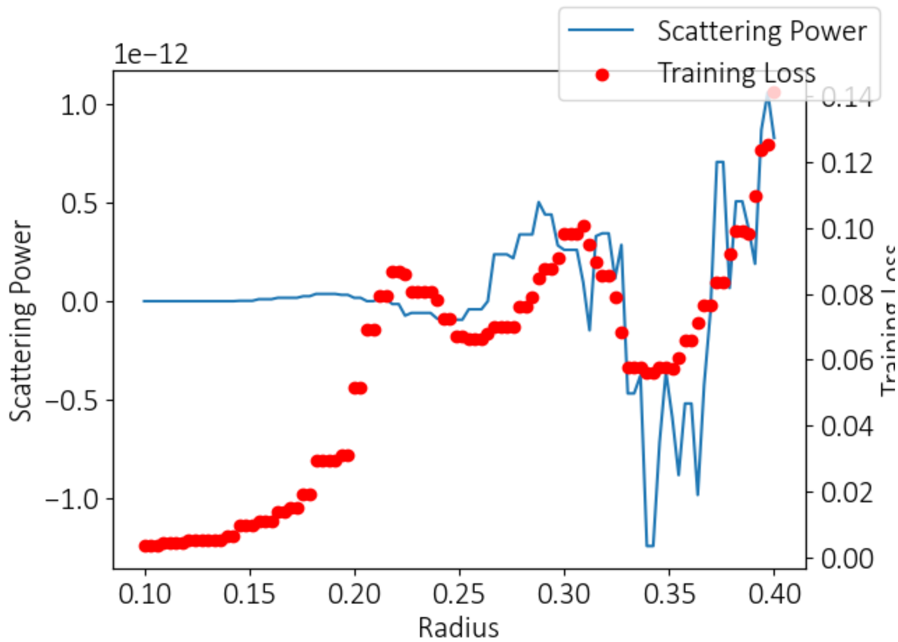

Figure 5: Calculate and plot are based on the output of MaxwellNet. Train dataset includes 200 sphere samples whose radius is randomly from 0.1 to 0.4, dpl is 32, and pml thickness is 32, and final average train loss is 0.052.

For 3D case, the scattering efficiency curve based on MaxwellNet still has a big dismatch with the ground truth coming from COMSOL. The final average train loss of 3D case is 0.052 which seems very low compared to that of 2D case (0.3), but 3D case requires lower loss to have an accurate prediction. It is also shown through the compare between ground truth and prediction result with different train loss.

Parameters are set as R=300nm, dpl=32, pmlthickness=32, and the plot shows the x-y plane of real(Ex)

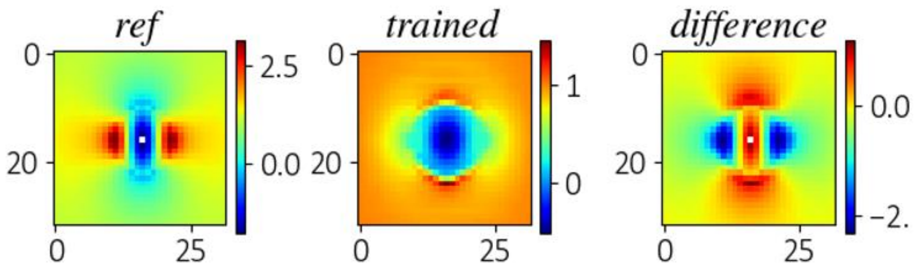

Figure 6: the left one is the ground truth got from MatScat matlab package, the middle one is the output of MaxwellNet, the right one is the difference of them. Train loss is 0.098

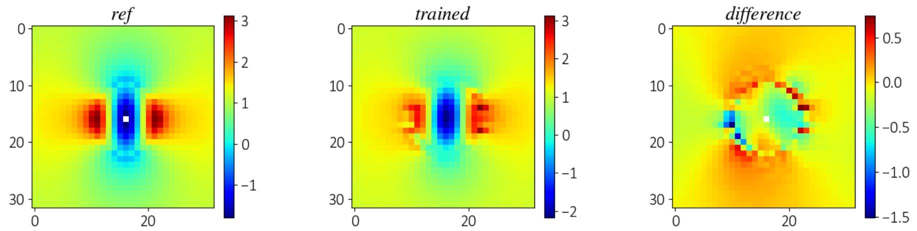

Figure 7: Train loss is 0.0027

From the compare, we can conclude that 0.052 is not a low enough loss to get an accurate result. Something on network optimization should be done.

Inverse design based on MaxwellNet

Problem description

Given a rod with diffuser section in the xz plane infinitely extended along the y direction, we want to find the most suitable TE polarization incidence that can be focused by the diffuser on a point we define.

Experiment design

When we use MaxwellNet to predict Mie scattering, the dataset only contains the refractive distribution, and incidence grid is a fixed one that is defined directly in MaxwellNet. Now we need to add incidence grid to train dataset for the need of inverse design. The structure of dataset is [N, C, W, H], the first channel holds refractive distribution and other channels hold incidence grid.

The steps of achieving inverse design are as followings:

1.Define grid matrix for planewave1…planewaven whose incident angle lies between $(-\frac{\pi}{2}, \frac{\pi}{2})$, and the sample at high angle should be relatively sparse to be more similar as natural case. In coding work, we use the following sentences:

```
k = 2*pi/wavelength
kix = fftshift(pi/Lx * linspace(-Nx // 2, Nx // 2, Nx))
kiz = nan_to_num(sqrt(k**2-kix**2))
```

Lx is the length of the whole field window.

With incidence at different angle and different refractive distribution (For cylinder, it means different center and radius. For diffuser, it means different minimum average thickness, maximum average thickness, maximum interface modulation frequency and center.), we can form train dataset that covers enough cases needed for inverse design.

2. After training with the dataset, MaxwellNet can predict total field for the refractive distribution appearing in the dataset under planewave incidence with different angle. Choose a shape that we want to research and get its output under different planewaves.

Take MaxwellNet as f(.)

Output1 = f(planewave1)

...

Outputn = f(planewaven)

Output = a1*output1 + a2*output2 + … + an*outputn

If output1 corresponds to planewave1 … outputn corresponds to planewaven, then a1*output1 + a2*output2 + … + an*outputn corresponds to a1*planewave1 + a2*planewave2 + … + an*planewaven (Maxwell equations are linear equations when n is not affected by light intensity).

3.Optimize Output to the field wanted through changing [a1, a2 ... an]. The incidence of optimized output is a1*planewave1 + a2*planewave2 + … + an* planewaven

Result and analysis

Test is done on a diffuser

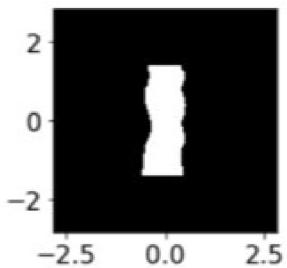

Figure 8: The diffuser that we do test on.

Our target is to find the incidence that can focus the most energy on point $(\frac{3}{4}N_z, \frac{1}{2}N_x)$. Use Adam to do the optimization. We can get the optimization result as the following

```
Physical Loss: 2.17e-01
Equivalent Loss: 1.10e-02
optimal a is

tensor([[0.2412],
        [0.1615],
        [0.3781],
        [0.5263],
        [0.5151],
        [0.3680],
        [0.4348],
        [0.5515],
        [0.5674],
        [0.4371],
        [0.2178],
        [0.0348]], device='cuda:0', requires_grad=True)
optimal phi is

tensor([[-0.5338],
        [-0.1964],
        [-0.1034],
        [-0.0833],
        [ 0.0679],
        [ 0.4890],
        [ 0.4054],
        [ 0.0281],
        [-0.1796],
        [-0.3152],
        [-0.5322],
        [ 0.1984]], device='cuda:0', requires_grad=True)
<matplotlib.collections.PathCollection at 0x7f31dbf201f0>
```

Figure 9: Optimization result of [a1 … an]

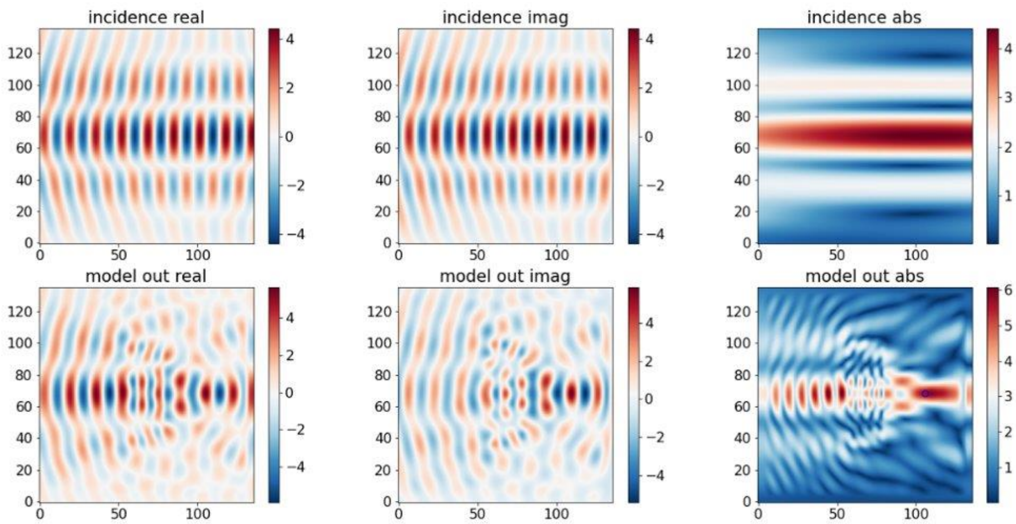

Figure 10: Optimized incidence and its output

Figure 11: Compare with the ground truth got from COMSOL

In the ground truth making process, we also tried package ceviche, and we got big-dismatch result, but we got good results using COMSOL, so this package may have some errors.

The U-Net architecture doesn't show enough generality in the experiment, and we need to make subsets (one subset covers all diffuser shapes, but the incident angle only covers a small range.) to train several models to get acceptable loss for all samples.

When we do research on circle case (different radius, different center and different incident angle), it struggles to get a good train loss even using subset. It indicates that U-Net architecture doesn't have good generality and we need to do something on network optimization.

Network Optimization

From the experiment of using MaxwellNet to solve problems above, we can see that the U-Net architeture doesn't have good ability to provide low enough train loss for dataset covering different shapes (or together with different planewave incidence) especially circles with different radius and center. In that case, more architectures should be explored.

Deeper U-Net

Dataset has 40000 diffusers with different shapes and different incident angles, test U-Net with 3 sizes: depth=6, filter=16; depth=7, filter=17; depth=8, filter=18. Simply increasing the size of U-Net doesn't increase the generality ability of architecture. Always, the saturate train loss of subsets with small incident angles is around 0.05, and the saturate train loss of subsets with big incident angles is around 0.2. It shows use deeper U-Net doesn't give a better generality performance.

Vision transformer family

Masked autoencoders (MAE)[2], Self-supervised vision Transformer (SiT)[3], Global Filter Network (GFNet)[4], Vision Attention Network (WAN)[5] are architectures based on the vision transformer method (Just change one or several blocks compared to initial vision transformer method). They perform very poor on our task.

Use only one sample as dataset to train GFNet, and the train loss is as the following. Delete the loss during the beginning 100 epochs, because it is too high and will hide the loss later in the plot

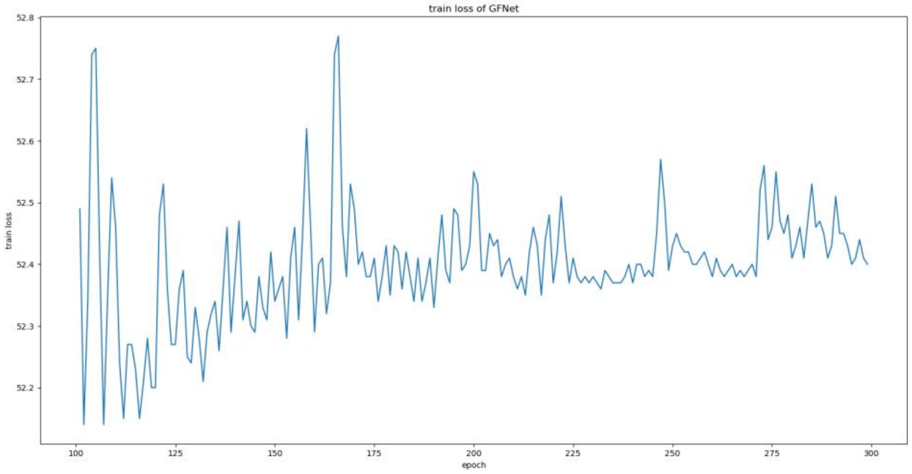

Figure 12: Train loss of GFNet using one-sample train dataset

Changing parameter pairs of GFNet still doesn’t provide an acceptable result. The tests of MAE, SiT, WAN give similar results. The saturate train loss is even hard to go below 10 which is quite big.

Architectures in vision transformer family are not suitable for our task.

CNN

Lippman-Schwinger equation describes the map relationship from a spatial refractive index distribution and the total electromagnetic field to the total electromagnetic field in the same domain when we impose an incident field as the following:

$$E_t(\mathbf{r}) = E_i(\mathbf{r}) + \int_\Omega G(\mathbf{r}-\mathbf{r}')f(\mathbf{r}')E_t(\mathbf{r}')d\mathbf{r}$$

where is $G(r-r')$ the dyadic Green tensor and $f(r)$is the scattering potential $(\frac{4\pi^2}{\lambda}(\frac{\varepsilon}{\varepsilon_b}-1))$.

For circular section case, if R=600nm (center is at the middle of the square window), wavelength=680nm (incident angle is 0), n=1.05, nb=1, the scattering field is weak compared to incident field, and we can use Born approximation. The equation becomes:

$$E_t(\mathbf{r}) = E_i(\mathbf{r}) + \int_\Omega G(\mathbf{r}-\mathbf{r}')f(\mathbf{r}')E_i(\mathbf{r}')d\mathbf{r}$$

The problem mapping incidence to scattering field is linear, we try to use a multilayer perceptron with no activation functions to fullfill the map. When R becomes small, or the difference of n and n_b becomes big, we can not use Born approximation, and the problem is not linear, we try to solve it by adding activation functions to introduce nonlinearity.

The project guideline asks to simplify the loss function based on Born approximation, but we prove that it is not accessible in the experiment process.

The full loss function is

$$\nabla\times\nabla\times\mathbf{E}_s - k_0^2n^2\mathbf{E}_s - k_0^2(n^2-n_b^2)\mathbf{E}_i = 0$$

The method in the project guideline is to delete the term $k_0^2n^2E_s$ based on $|E_s| \ll |E_i|$, so the equation can be simplified to

$$\nabla\times\nabla\times\mathbf{E}_s = k_0^2(n^2-n_b^2)\mathbf{E}_i$$

However, when the practical experiment based on the simplified equation is done, the output prediction field goes very far from the ground truth. After data analysis, it is found that term $k_0^2n^2E_s$ and term $k_0^2(n^2-n_b^2)E_i$ are of the same order of magnitude. Although $|E_s| \ll |E_i|$, $(n^2-n_b^2)$ is also much smaller than n^2, so the simplification given in the project guideline is wrong.

Skip the simplification of loss function, still use the full loss function and test a multilayer perceptron with no activation functions。 It is found that the simple CNN shows quite poor map ability. The parameters of the CNN are: in_channels=1, out_channels=2, hidden_layers=5, hidden_channels=32, img_size=256, ker_size=3, padding=0, no normalization and activation functions.

The test result is as the following

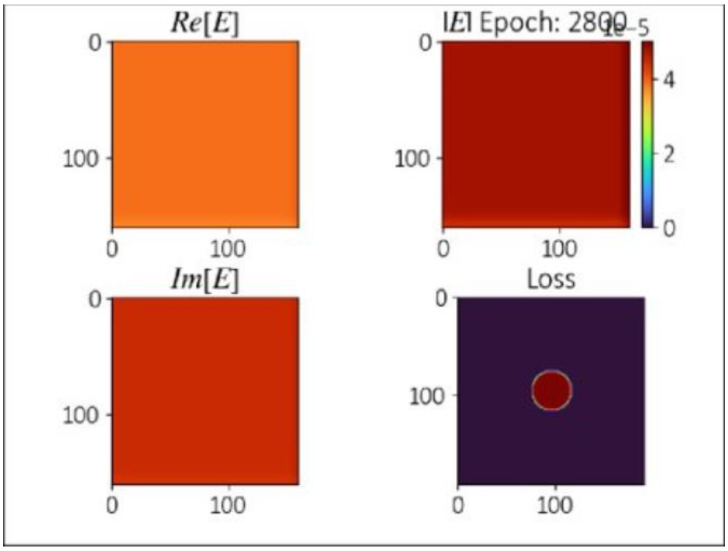

Figure 13: The map result of a multilayer perceptron

Some different parameter pairs are also tested, and there is no improvement for nearly all of them except one way. It will be talked about in the next section. If activation functions are added (nearly all kinds of activation functions have been tested), the result also has no improvement. This simple CNN architecture proposed in project guideline is just not suitable for our task.

Reframe-CNN

Although the test of Lippman-Schwinger model with MaxwellNet is not successful, it is found that if the kernel size of convolution layer is set to a very big value, like 513 for case image_size=256, an acceptable output can appear. The problem of this action is that the size of model could be very big and the training speed is very slow. However, it still gives some ideas to design the architecture. Through the big-kernel convolution layer, every output pixel is got from the combination of all pixels rather than pixels in a small local region. This follows the physical Lippman-Schwinger equation that performs a global integration over all points rather than only the points in a local region.

The architecture of Reframe block we propose is as the following

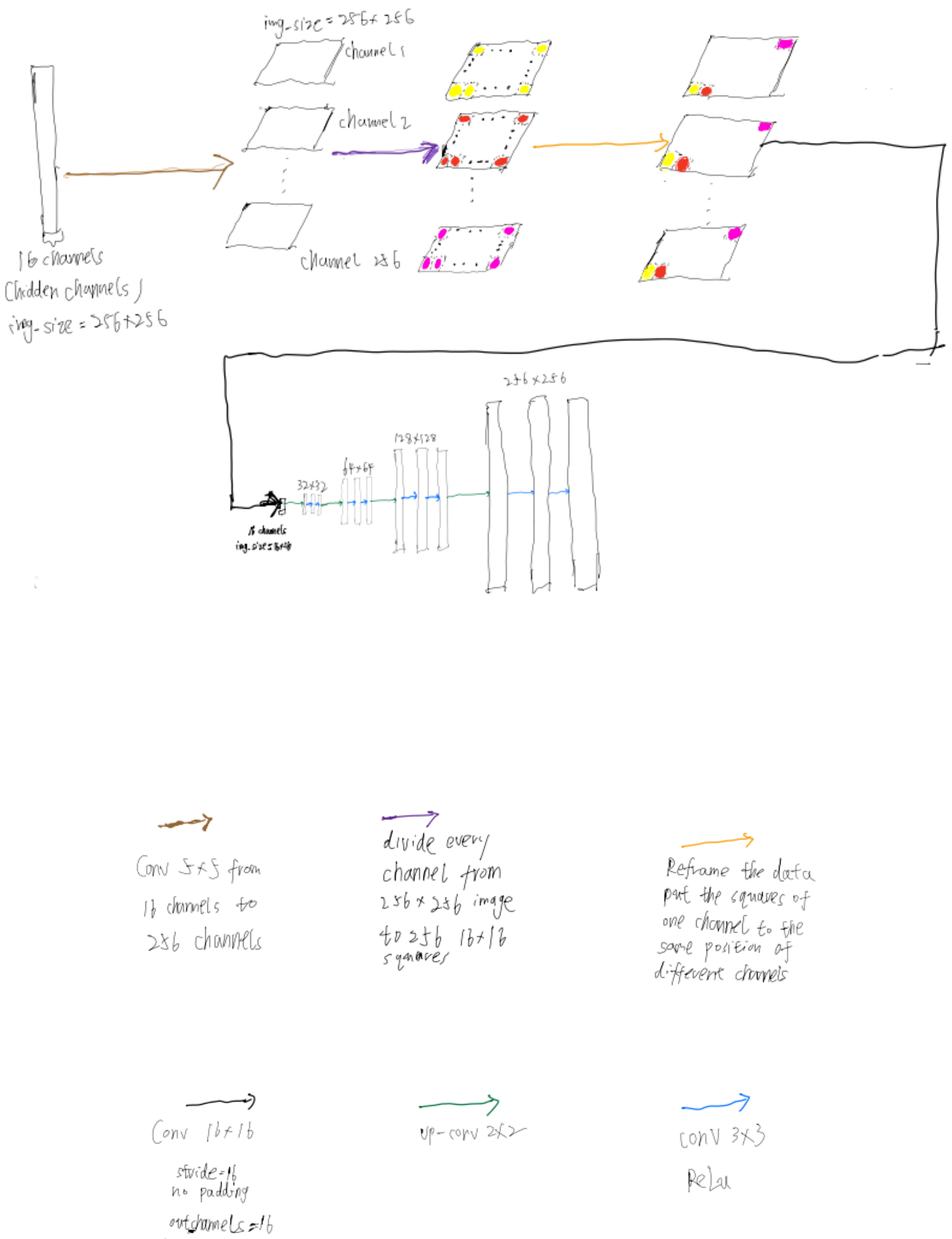

Figure 14: Architecture of Reframe block

Through the steps symboled by brown, purple, orange and black arrows, a data block with shrinked image size is got. Every pixel in the new data block is a combination of global pixels rather than local pixels.

Steps symboled by green and blue arrows are used to recover the image size. If we directly use up-conv 16x16 rather than four up-conv 4x4, square blurs will appear in th output and heavily affect the quality as the following

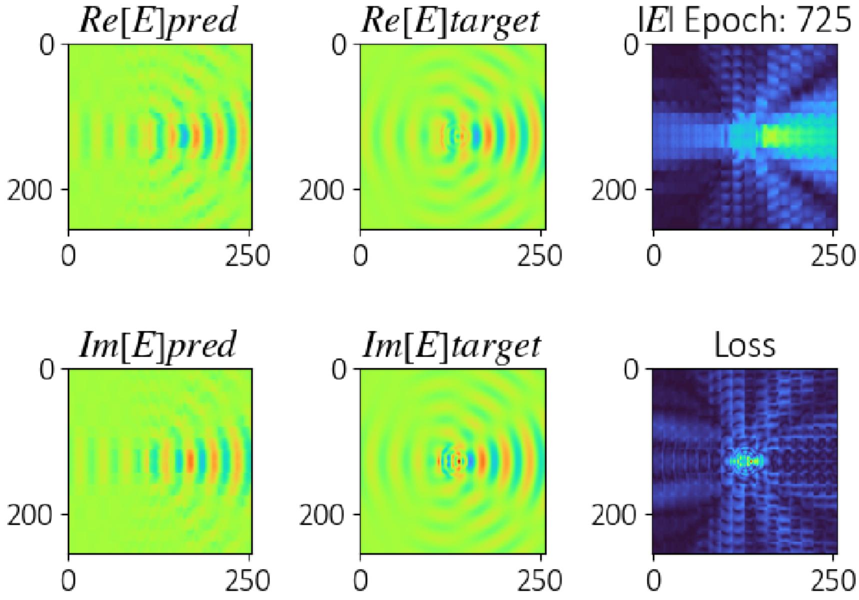

Figure 15: Output with square blurs using up-conv 16x16

Using Reframe-CNN to predict the total field of circular section. To better show the performance of network architecture, we don’t use the loss function and directly use data got from COMSOL to do a data-driven train.

For case radius=600nm, wavelength=680nm, n=1.05(weak scattering), the train result is

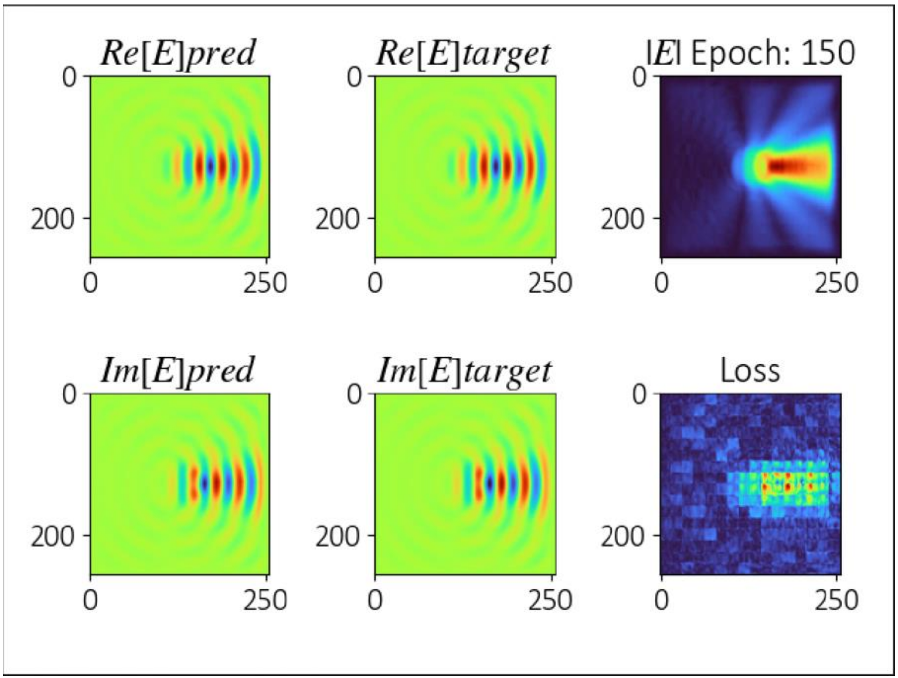

Figure 16: Prediction of Reframe-CNN for weak sacttering

For case radius=300nm, wavelength=680nm, n=3.5(strong scattering), the train result is

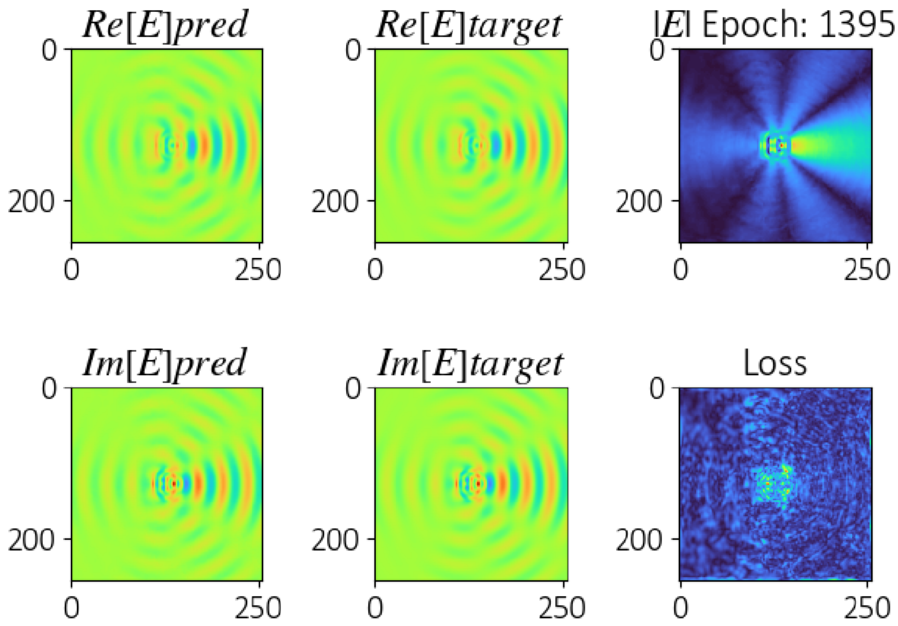

Figure 17: Prediction of Reframe-CNN for strong sacttering

Also, do a compare study between U-Net and Reframe-CNN for strong scattering case, the result is

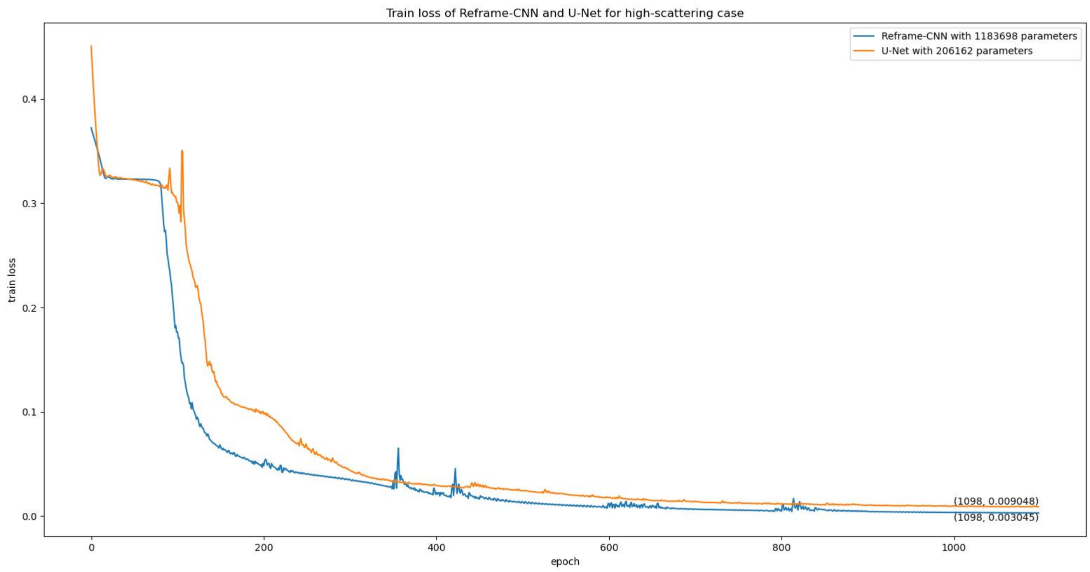

Figure 18: Compare between Reframe-CNN and U-Net

Reframe-CNN shows very good performance in the test of single-sample dataset for both weak scattering case and high scattering case. Compared to U-Net, Reframe-CNN we propose has faster converge speed and lower saturate train loss, but the size of it is bigger than that of U-Net.

Due to time limitation, now we only finish the test on single-sample dataset. Further architecture improvement should be done for Reframe-CNN based on the test result of multi-sample dataset.

References

[1] Lim J, Psaltis D. MaxwellNet: Physics-driven deep neural network training based on Maxwell's equations[J]. Apl Photonics, 2022, 7(1): 011301.

[2] He K, Chen X, Xie S, et al. Masked autoencoders are scalable vision learners[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 16000-16009.

[3] Ahmed S A A, Awais M, Kittler J. Sit: Self-supervised vision transformer[J]. ArXiv abs/2104.03602, 2021, 5.

[4] Rao Y, Zhao W, Zhu Z, et al. Global filter networks for image classification[J]. Advances in neural information processing systems, 2021, 34: 980-993.

[5] Guo M H, Lu C Z, Liu Z N, et al. Visual attention network[J]. arXiv preprint arXiv:2202.09741, 2022.