Clean Implicit 3D Structure from Noisy 2D STEM Images  
Supplemental Material  

Hannah Kniesel  
Ulm University  

Timo Ropinski  
Ulm University  

Tim Bergner  
Ulm University  

Kavitha Shaga Devan  
Ulm University  

Clarissa Read  
Ulm University  

Paul Walther  
Ulm University  

Tobias Ritschel  
University College London  

Pedro Hermosilla  
Ulm University  

Figure 1. Model architecture of the MLP used to encode the implicit reconstruction. All boxes reference fully connected layers. Light gray boxes use ReLU activation function, while dark gray boxes use sigmoid activation function. One skip connection is used by concatenating the layers output with the models input.

1. Model Architectures

1.1. Implicit Model

The MLP Architecture is inspired by the architecture used in NeRF by Mildenhall et al. [5]. We forward the positional encoding of the sample’s position in model space through nine fully connected layers with 256 features and a ReLU activation function. We retrieve the output by a sigmoid activation function in the output layer to predict densities in the range \([0, 1]\). We use one skip connection, which concatenates the output of the previous layer with the input, as seen in Fig. 1.

1.2. Noise Model

The noise model consists of a Normalizing Flow network. This network comprises eight 1D Radial Flow layers [7], of which four layers are conditioned on the clean signal. To condition the layers on the signal we use a MLP with one hidden layer with 16 features and ReLU activation. The output layer uses Tanh activation to fit the parameter range \([-1, 1]\). This MLP then predicts parameters of the 1D Radial Flow layer based on the input pixel intensity. To retrieve the noise distribution, which we want to transform into a normal distribution using the noise model, we compute the difference of the predicted clean signal and the known noisy signal. In Ours, the clean signal is retrieved from the prediction of the implicit model since only the noisy measurement is available.

2. Synthetic Data

To generate synthetic data we randomly place ellipsoidal shells and a density model of the ZIKV (i.e., Zika) virion at 15Å by Long et al. [3] in a cubic volume. In Fig. 3 we show an overview of the used projections for the different methods.

3. Noise Synthesis

To generate the noise of the projections we train our noise model in a supervised fashion from pairs of long and short exposure STEM images, as already described in the main paper Sec. 4. We here evaluate the trained noise model in comparison to two baseline approaches: First, we assume a Gaussian distribution and optimize it’s parameters from the long and short exposure data using MLE. Second, we assume a Poisson distribution and again optimize it’s parameters from the long and short exposure data using MLE. Compared to the Gaussian, the Poisson distribu-
Figure 3. Examples of synthetic data at low tilt angle. **Clean** is generated using our image formation model. **Noisy** adds synthetic noise to the Clean, using Normalizing Flows. **Denoised** is a denoised version of Noisy, using BM3D.

Table 1. Main quantitative results of different methods for noise modeling. We report Jensen-Shannon-Divergence (JSD), Bhattacharyya coefficient (BC) and Bhattacharyya distance ($d_{BC}$). We optimize the parameters of the distributions Poisson and Gaussian using MLE. Our approach using Normalizing Flows outperforms the baseline methods in all metrics. The best method is shown in **bold**.

<table>
<thead>
<tr>
<th>Methods</th>
<th>JSD</th>
<th>BC</th>
<th>$d_{BC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>0.96</td>
<td>0.99</td>
<td>0.15</td>
</tr>
<tr>
<td>Gaussian</td>
<td>1.07</td>
<td>0.99</td>
<td>0.12</td>
</tr>
<tr>
<td>Normalizing Flow</td>
<td><strong>0.58</strong></td>
<td><strong>1.00</strong></td>
<td><strong>0.03</strong></td>
</tr>
</tbody>
</table>

4. Model Capacity

We further investigate the influence of model capacity on the performance of our method. Therefore, we separately investigate the capacity of the implicit model, and the noise model.

4.1. Implicit Model Capacity

To investigate the effect of the capacity of the implicit model we increase and decrease the number of features in the hidden layers. The capacity of the noise model remains fixed during these experiments. We compare the performance of **Ours** and **L2Noisy**. We investigate performance for 32, 64, 128 and 256 features in all hidden, fully connected layers. We find that increasing capacity slightly improves performance of **L2Noisy** and **Ours**. Specially for small model capacities **Ours** is not able to outperform **L2Noisy** (see Table 2).

4.2. Noise model Capacity

To tune the capacity of the noise model we increase and decrease the number of layers used. In this experiment, we reduce/increase the eight layers of the model by multiples of two. Still, the number of conditional and unconditional layers is always balanced. We fix the capacity of the implicit model to the one described in Sec. 1.1 and then train the model using **Ours**. We investigate performance of the noise model using 2, 4, 8 and 16 layers. Quantitative evaluation (see Table 3) shows that increasing the capacity of
Figure 4. Qualitative results of different methods (rows) for modelling the noise. We use MLE to approximate parameters of a Gaussian and Poisson distribution from the data. Normalizing Flow is trained on the data and makes no further assumption of the noise distribution. It outperforms the former.

Table 2. Main quantitative results of the influence of MLP Capacity. For both methods the reconstruction accuracy seems to improve with increased model capacity. Especially noteworthy is the finding, that L2Noisy outperforms Ours for small model capacities by a slight margin. We argue that this finding is mostly accountable to the imbalance of the noise model and the implicit model, since capacity of the noise model was fixed for all experiments on the MLP Capacity. Experiments on the noise model capacity underline this assumption.

<table>
<thead>
<tr>
<th>Method</th>
<th>Features</th>
<th>2D PSNR</th>
<th>MSE</th>
<th>DSSIM</th>
<th>3D PSNR</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2Noisy</td>
<td>32</td>
<td>12.84</td>
<td>5.397</td>
<td>2.038</td>
<td>19.73</td>
<td>1.065</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>13.69</td>
<td>4.433</td>
<td>1.925</td>
<td>19.57</td>
<td>1.103</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>13.68</td>
<td>4.450</td>
<td>1.884</td>
<td>19.85</td>
<td>1.034</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>13.86</td>
<td>4.271</td>
<td>1.885</td>
<td>19.73</td>
<td>1.064</td>
</tr>
<tr>
<td>Ours</td>
<td>32</td>
<td>13.47</td>
<td>4.672</td>
<td>1.990</td>
<td>19.28</td>
<td>1.181</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>13.32</td>
<td>4.830</td>
<td>1.925</td>
<td>19.93</td>
<td>1.017</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>14.15</td>
<td>3.996</td>
<td>1.912</td>
<td>19.33</td>
<td>1.166</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>19.93</td>
<td>1.020</td>
<td>0.645</td>
<td>21.75</td>
<td>0.669</td>
</tr>
</tbody>
</table>

only the noise model does not necessarily improve reconstruction accuracy of Ours.

4.3. Discussion

Ours is able to prevent the noise model to learn structures of the 3D signal, since we only condition it on single pixels. On the other hand, we can not prevent the implicit model to incorporate the noise in the reconstruction by mapping it to a cylindrical structure around the region of interest as done by L2Noisy. Hence, we need to find a balance in training to restrict the implicit model. The balance between noise model and implicit model can be influenced by many different factors such as learning rate, optimizer choice, used model capacities and loss regularization terms. Also, approaches like alternate training, which is commonly used for the training of GAN [2] models, to train Generator and Discriminator networks, might help to find a suitable balance between the noise model and the implicit model. We leave the investigation of these factors for future work.

5. Defocus

During data acquisition of STEM, out-of-focus areas can occur especially at high tilt angles and at distances far away from the tilt axis. This can influence the reconstruction process, since observations of the same point in world space appear differently, when seen from different angles. We
Table 3. Main quantitative results of the influence of noise model Capacity. The experiment underlines the importance of balance between the noise model and the implicit model. Increasing the capacity of the noise model will not compulsorily improve performance.

<table>
<thead>
<tr>
<th>Layers</th>
<th>2D PSNR</th>
<th>2D MSE</th>
<th>2D DSSIM</th>
<th>3D PSNR</th>
<th>3D MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13.77</td>
<td>4.352</td>
<td>1.861</td>
<td>19.77</td>
<td>1.054</td>
</tr>
<tr>
<td>4</td>
<td>15.71</td>
<td>2.792</td>
<td>1.461</td>
<td>20.62</td>
<td>0.866</td>
</tr>
<tr>
<td>8</td>
<td>19.93</td>
<td>1.020</td>
<td>0.645</td>
<td>21.75</td>
<td>0.669</td>
</tr>
<tr>
<td>16</td>
<td>14.22</td>
<td>3.921</td>
<td>1.865</td>
<td>19.43</td>
<td>1.141</td>
</tr>
</tbody>
</table>

show, using synthetic data, that accounting for this blur during reconstruction can help to improve the reconstruction. Therefore, we apply a Gaussian blur with a variable kernel size $\kappa$, depending on the formula:

$$\kappa(x) = \exp\left(-||x|| \cdot \tan(\alpha) \cdot d\right)$$

(1)

where $\alpha$ is the tilt angle and $d$ the distance in image space from the tilt axis. This formula assigns a larger kernel size to areas with high tilt angle and far distance from the tilt axis. An example of the synthetic data in comparison to real data can be seen in Fig. 5.

During reconstruction, we apply Monte Carlo integration over the defocus area in the image by sampling multiple rays. However, for computational reasons, we use only one sample during training. This setup converges more slowly than using multiple samples but allows for sampling more pixels in each batch. We can show in a quantitative evaluation (see Table 4) that, assuming the emergence of out-of-focus blur is known, handling this blur during training improves the reconstruction quality.

6. Comparison of L1 and L2 Loss

For learned approaches which do not use a noise model, we compare the use of $L_1$ and $L_2$ loss. Therefore, similar to $L2Noisy$ we train an implicit model using $L_1$ loss. We will refer to this model as $L1Noisy$. We found that $L2Noisy$ outperforms $L1Noisy$ by a large margin. We hence used $L_2$ loss for all learned reconstructions without a noise model.

Qualitative as well as quantitative evaluation can be seen in Fig. 6. Here, we also compare to the use of the noise model by comparing to $Ours$.

7. Denoising of Projections

We explore different denoising algorithms in order to train $L2Den$. We further evaluate the impact of denoising using WBP for reconstruction. We compare BM3D [4], Deep Wiener-Kolmogorov Filters [6] and Topaz Denoise (TD) [1]. We used the provided code by the authors to apply denoising to the synthetic micrographs. For BM3D denoising we used the provided python package. Regarding Deep Wiener-Kolmogorov Filters we assume a poisson (DWK-P) as well as a gaussian (DWK-G) noise distribution.

7.1. WBP

We investigate the influence of denoising the micrographs before applying WBP for reconstruction. We will call this method $WBPDen$. For results see Fig. 7.

We found that reconstruction quality was improved by all denoisers. Especially DWK-P was outperforming all other denoisers regarding quantitative evaluation. Still, the reconstruction quality using WBP was worse compared to all considered learned approaches. Moreover, we found that especially small details are not well recovered when working on denoised micrographs.

7.2. Learned Reconstruction

Similar to Sec. 7.1 we investigate the influence of denoisers on $L2Den$. See results in Fig. 8.

We found that BM3D outperforms all other denoisers. Hence, for all considered experiments of $L2Den$ we used BM3D denoising. Still, similar to $WBPDen$, small details are not well recovered when working on denoised micrographs.

8. Comparison of Implicit and Explicit Reconstruction

We compare the benefits of using an implicit representation of the reconstruction with and without the use of a noise model. We therefore compare $Ours$ and $L2Noisy$.
Figure 5. **Left:** Synthetic out-of-focus image at a high tilt angle. Blur is more prone for pixels further away from the tilt axis. **Right:** Out-of-focus real data for image at high tilt angle. Again, the blur is more prone in regions far away from the tilt axis.

Figure 6. Comparison of loss functions to compute implicit reconstruction with no noise model ($L_1$, $L_2$). **Ours** on the other hand uses a noise model and hence uses a Maximum-likelihood Estimation (MLE) loss. For reconstructions without a noise model we found that $L_2$ outperforms $L_1$.

<table>
<thead>
<tr>
<th></th>
<th>Ground Truth</th>
<th>Ours</th>
<th>L2Noisy</th>
<th>L1Noisy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>21.75</td>
<td>0.69</td>
<td>1.064</td>
<td>3.195</td>
</tr>
<tr>
<td>MSE</td>
<td>19.73</td>
<td>14.95</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with the use of an implicit reconstruction and an explicit reconstruction accordingly. We initialize the explicit repre-
Figure 7. Comparison of denoisers to apply to noisy micrographs before reconstruction with WBP. We found that the reconstruction on micrographs which have been denoised with DWK-P outperforms the other denoisers regarding quantitative evaluation. Still, learned reconstructions outperform WBP Den by a large margin.

Figure 8. Comparison of denoisers to apply to noisy micrographs before reconstruction with L2Den. We found that the reconstruction on micrographs which have been denoised with BM3D outperforms the other denoisers regarding quantitative as well as qualitative evaluation.

\[ L = L_{\text{network}} + \lambda \cdot L_{TV} \]  

where \( L_{\text{network}} \) corresponds to the \( L_2 \)- or MLE-loss, depending on the used method. \( L_{TV} \) corresponds to the TV regularization which we compute on the 3D reconstruction volume. We use \( \lambda \) to weight the regularization term. Based on the different scopes of the loss functions, we found that \( \lambda = 0.05 \) performed the best for the \( L_2 \)-loss, while \( \lambda = 50 \) performed the best for the MLE loss.

For training of the explicit model without a noise model, we use an ADAM optimizer with a learning rate of \( 5^{-5} \). For the training of the explicit model using a noise model, we again use an ADAM optimizer, with a learning rate of \( 5^{-6} \). The noise model uses a SGD optimizer with a learning rate of \( 1^{-4} \). We again train all models for 400,000 iterations and report the test error on the model with the highest validation accuracy.

We were not able to train the explicit reconstruction on
Figure 9. We evaluate the importance of using a noise model during the reconstruction, as well as the influence of using an implicit representation of the reconstruction. We evaluate this by comparing explicit and implicit reconstructions which use a noise model during training Ours and which do not use a noise model during training L2Noisy. We found that the use of an implicit representation helps to suppress artefacts generated by the missing wedge effect. Moreover, the use of a noise model seems to improve reconstruction quality.

We found that the explicit representation is susceptible in regard of the missing wedge effect. Further, without the use of a regularization term, we observe that the explicit representation is more prone to overfit to the noise in the projections than the implicit representation. Moreover, without the use of TV regularization, the use of a noise model does not help the reconstruction. However, with the application of the TV regularization, the noise model helps the reconstruction quality. Still, the combined use of implicit representation and noise model outperforms all other baselines.
References


