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

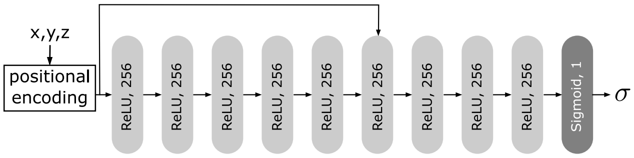


Figure 1. Model architecture of the MLP used to encode the implicit reconstruction. All boxes reference fully connected layers. Light gray boxes use ReLU activation function, while dark gray boxes use sigmoid activation function. One skip connection is used by concatenating the layers output with the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

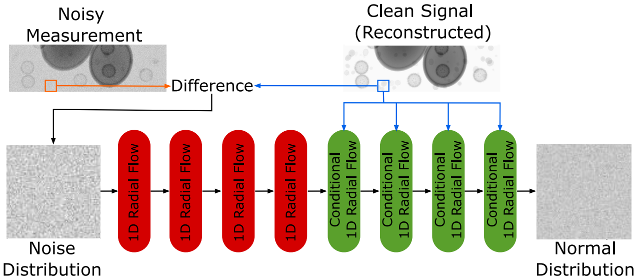

Noisy Measurement
Clean Signal (Reconstructed)
Difference
1D Radial Flow
1D Radial Flow
1D Radial Flow
1D Radial Flow
Conditional 1D Radial Flow
Conditional 1D Radial Flow
Conditional 1D Radial Flow
Conditional 1D Radial Flow
Noise Distribution
Normal Distribution

Figure 2. Model architecture of the Normalizing Flow used to model the noise. Red boxes reference 1D Radial Flow layers. Green boxes reference 1D Radial Flow images, which are conditioned on the clean signal. The noise distribution is retrieved by computing the difference of the clean signal and the noisy signal.

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-

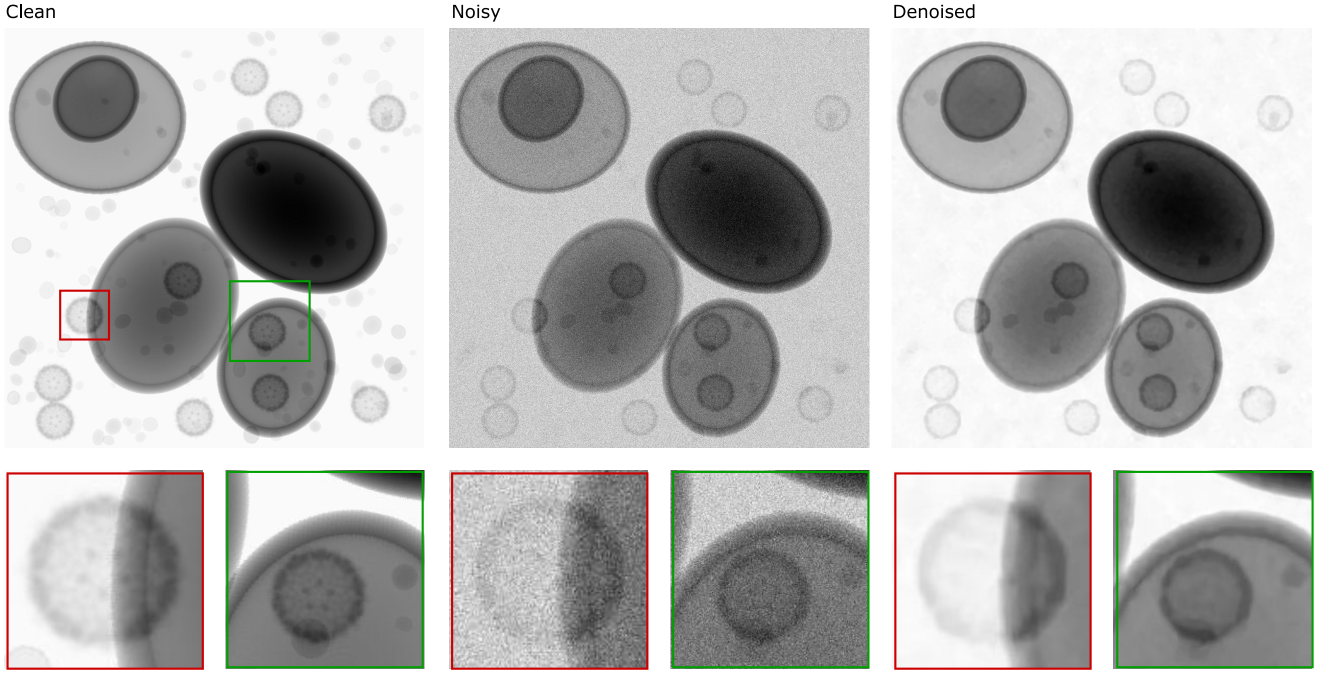


Figure 3. Examples of synthetic data at low tilt angle. **Clean** is generated using our image formation model. **Noisy** adds synthetic noise to the Clean, using Normalizing Flows. **Denoised** is a denoised version of Noisy, using BM3D.

tion is able to model signal dependence of the noise. We compare the resulting noise distributions with the given distribution of the data quantitatively by reporting the Bhattacharyya coefficient and distance, as well as the Jensen-Shannon-Divergence (Table 1). We also provide a qualitative evaluation in Fig. 4. Both baselines seem to fit the true distribution similarly well. While the Poisson distribution prevails according to the Jensen-Shannon-Divergence, the Gaussian distribution has the overhand regarding Bhattacharyya coefficient and distance. Still, the approximation using our noise model fits the real distribution the best in all metrics.

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**.

Methods	JSD	BC	d_{BC}
Poisson	0.96	0.99	0.15
Gaussian	1.07	0.99	0.12
Normalizing Flow	**0.58**	**1.00**	**0.03**

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

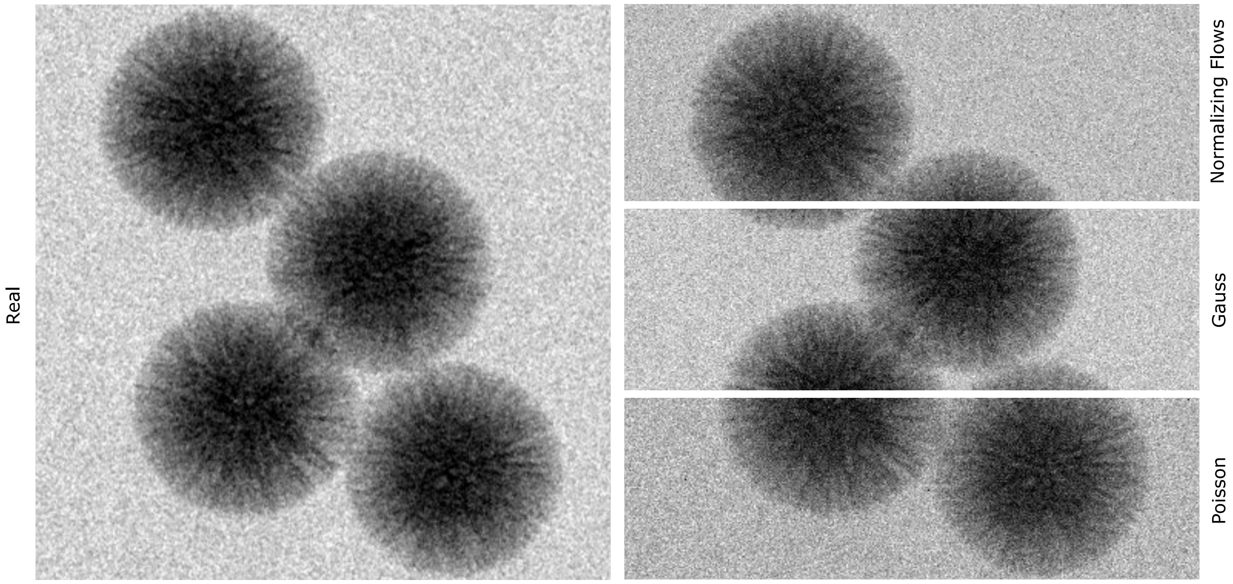
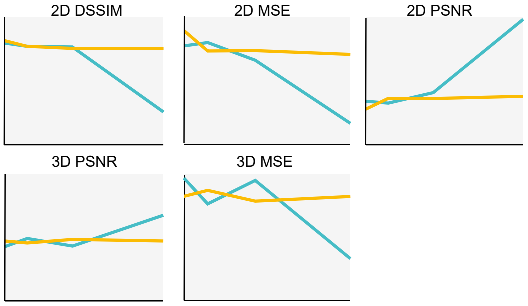


Figure 4. Qualitative results of different methods **(rows)** for modelling the noise. We use MLE to approximate parameters of a Gaussian and Poisson distribution from the data. Normalizing Flow is trained on the data and makes no further assumption of the noise distribution. It outperforms the former.

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.

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



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

4.3. Discussion

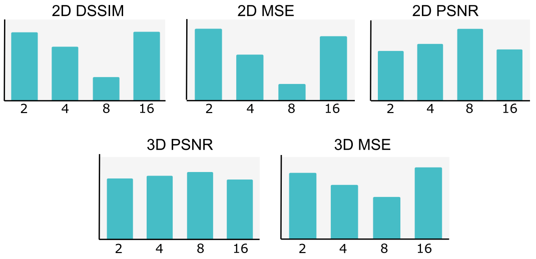
Ours is able to prevent the noise model to learn structures of the 3D signal, since we only condition it on single pixels. On the other hand, we can not prevent the implicit model to incorporate the noise in the reconstruction by mapping it to a cylindrical structure around the region of interest as 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.

Layers	2D			3D	
	PSNR	MSE	DSSIM	PSNR	MSE
2	13.77	4.352	1.861	19.77	1.054
4	15.71	2.792	1.461	20.62	0.866
8	19.93	1.020	0.645	21.75	0.669
16	14.22	3.921	1.865	19.43	1.141



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 κ, depending on the formula:

$$\kappa(\mathbf{x})(\alpha, d) = \exp(-||\mathbf{x}|| \cdot \tan(\alpha) \cdot d) \quad (1)$$

where α 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 outperfroms 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.

Table 4. Main quantitative results of the influence of the out-of-focus effect on the reconstruction. L2Clean functions as an upper bound, as it is trained on synthetic data without out-of-focus effect. L2Blur is trained similar to L2Clean but using synthetic data which contains out-of-focus images. Lastly, L2Blur+ is trained using synthetic data with out-of-focus images, taking this into account during training.

Method	2D			3D	
	PSNR	MSE	DSSIM	PSNR	MSE
L2Clean	20.79	0.838	0.383	21.47	0.712
L2Blur	19.66	1.090	0.514	20.15	0.965
L2Blur+	20.89	0.819	0.412	20.42	0.909

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

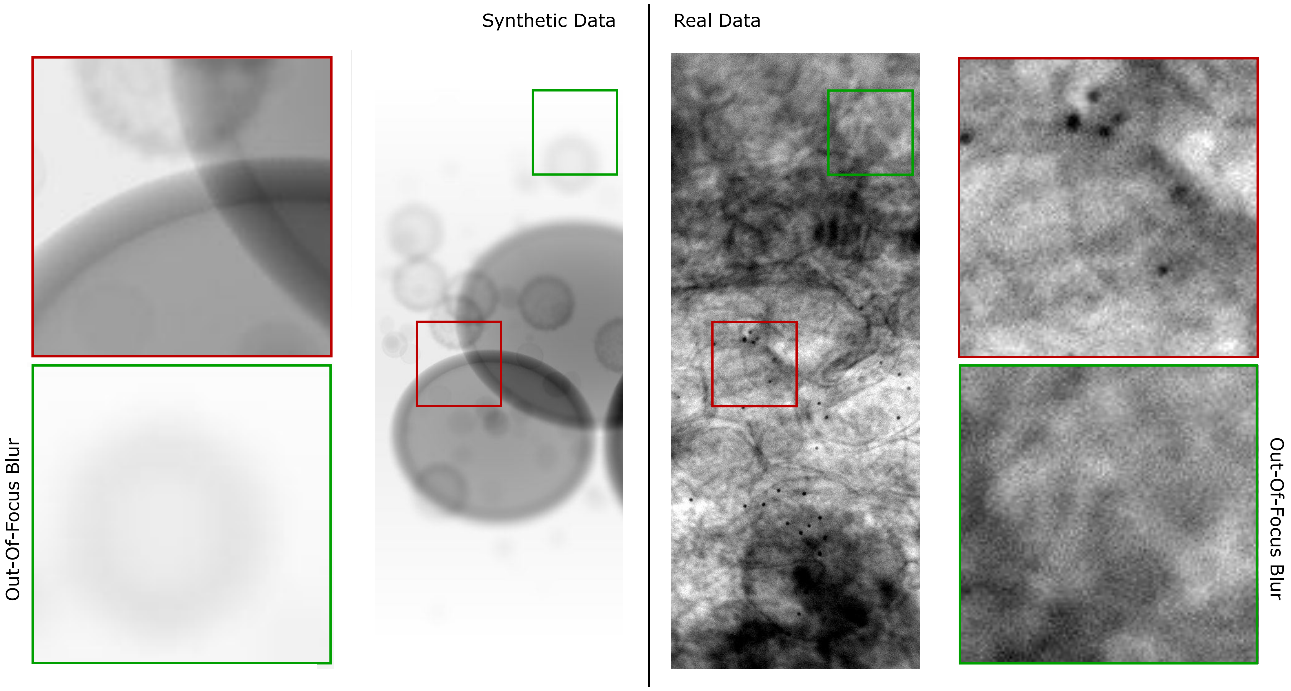


Figure 5. **Left:** Synthetic out-of-focus image at a high tilt angle. Blur is more prone for pixels further away from the tilt axis. **Right:** Out-of-focus real data for image at high tilt angle. Again, the blur is more prone in regions far away from the tilt axis.

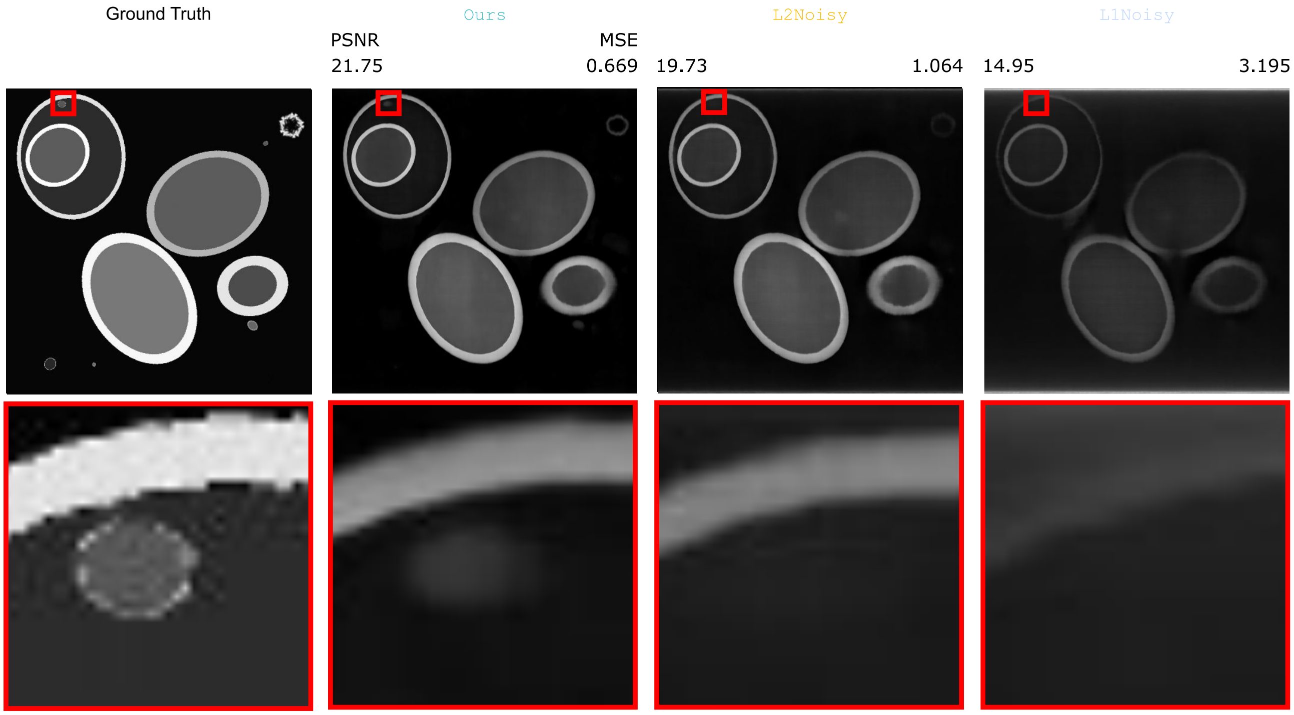


Figure 6. Comparison of loss functions to compute implicit reconstruction with no noise model (L_1, L_2). Ours on the other hand uses a noise model and hence uses an Maximum-likelihood Estimation (MLE) loss. For reconstructions without a noise model we found that L_2 outperforms L_1.

with the use of an implicit reconstruction and an explicit

reconstruction accordingly. We initialize the explicit repre-

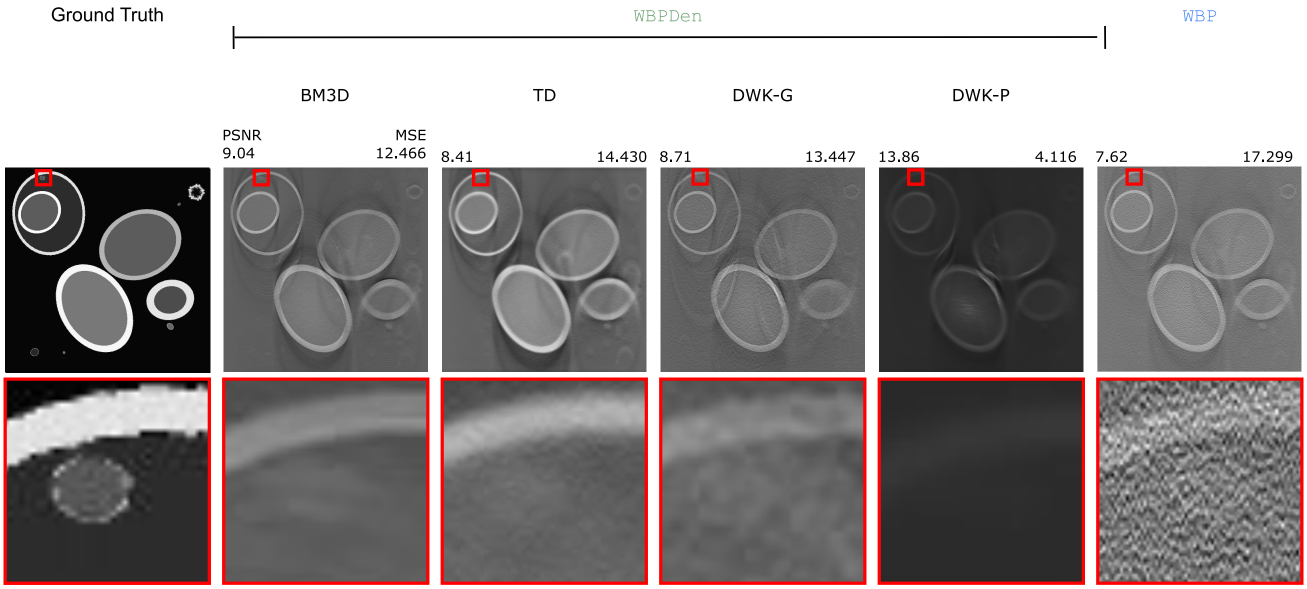


Figure 7. Comparison of denoisers to apply to noisy micrographs before reconstruction with WBP. We found that the reconstruction on micrographs which have been denoised with DWK-P outperforms the other denoisers regarding quantitative evaluation. Still, learned reconstructions outperform WBPDen by a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

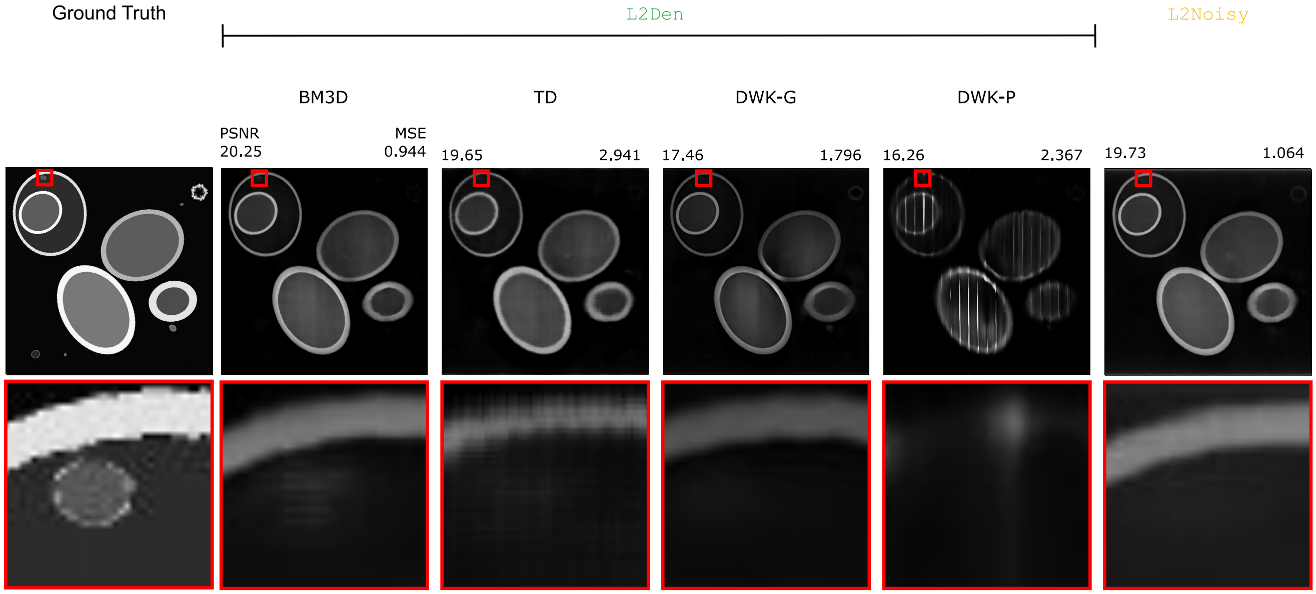


Figure 8. Comparison of denoisers to apply to noisy micrographs before reconstruction with L2Den. We found that the reconstruction on micrographs which have been denoised with BM3D outperforms the other denoisers regarding quantitative as well as qualitative evaluation.

sentation with zeros. During training of the explicit reconstruction we also use total variation (TV) regularization. We hence compute the loss

$$L = L_{\text{network}} + \lambda \cdot L_{\text{TV}} \tag{2}$$

where L_{network} corresponds to the $\mathcal{L}_2$- 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 λ 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 $\mathcal{L}_2$-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

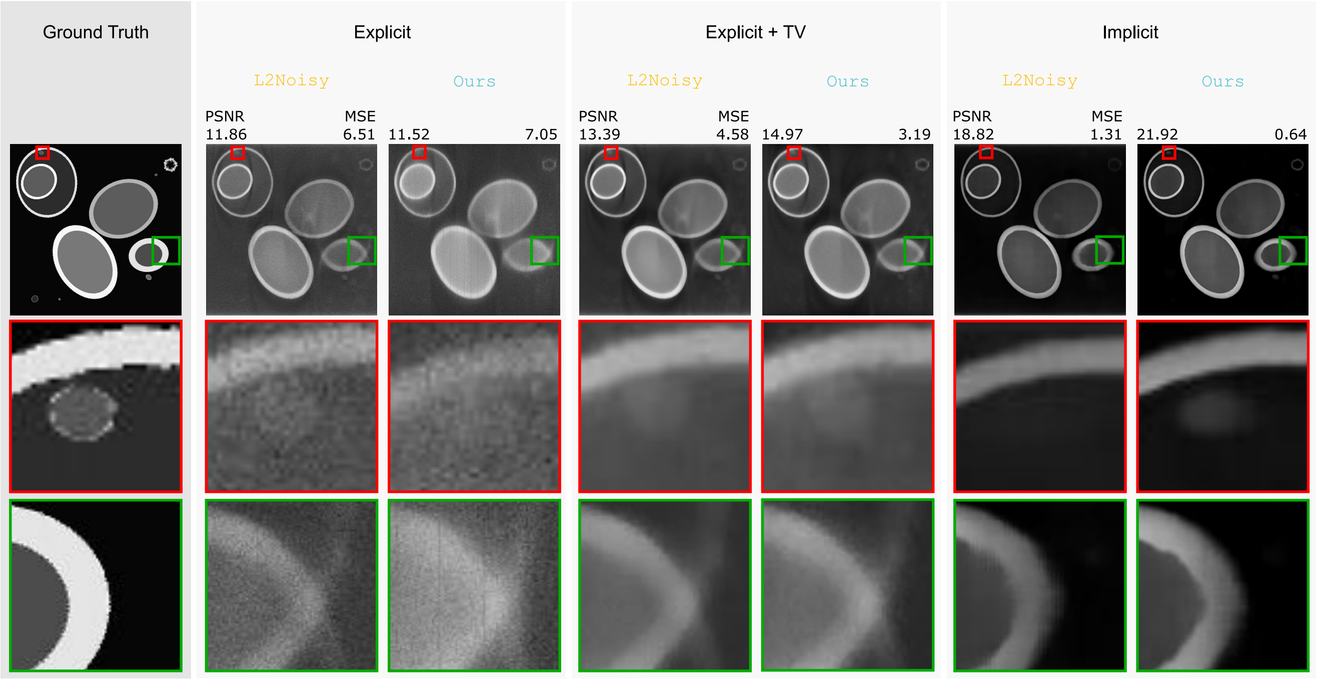


Figure 9. We evaluate the importance of using a noise model during the reconstruction, as well as the influence of using an implicit representation of the reconstruction. We evaluate this by comparing explicit and implicit reconstructions which use a noise model during training `Ours` and which do not 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.

a full sized volume of shape $1000 \times 1000 \times 1000$ voxels, based on limited memory resources. We hence trained the explicit reconstruction as volume of shape $512 \times 512 \times 512$. During evaluation we downsample the ground truth phantom volume and we reconstruct a tomogram of similar size of the implicit model. We report PSNR and MSE in 3D on the provided tomograms. Results can be seen in Fig. 9.

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

[1] Tristan Bepler, Kotaro Kelley, Alex J. Noble, and Bonnie Berger. Topaz-denoise: general deep denoising models for cryoem and cryoet. *Nature Communications*, 2020. 4

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[3] Feng Long, Michael Doyle, Estefania Fernandez, Andrew S Miller, Thomas Klose, Madhumati Sevvana, Aubrey Bryan, Edgar Davidson, Benjamin J Doranz, Richard J Kuhn, et al. Structural basis of a potent human monoclonal antibody against zika virus targeting a quaternary epitope. *PNAS*, 116 (5):1591–1596, 2019. 1

[4] Ymir Mäkinen, Lucio Azzari, and Alessandro Foi. Collaborative filtering of correlated noise: Exact transform-domain variance for improved shrinkage and patch matching. *IEEE Transactions on Image Processing*, 29:8339–8354, 2020. 4

[5] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–21, 2020. 1

[6] Valeriya Pronina, Filippos Kokkinos, Dmitry V Dylov, and Stamatios Lefkimmiatis. Microscopy image restoration with deep wiener-kolmogorov filters. In *European Conference on Computer Vision*, pages 185–201. Springer, 2020. 4

[7] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, pages 1530–1538. PMLR, 2015. 1