Deep Optimization Prior for THz Model Parameter Estimation (Supplementary Material)

Tak Ming Wong*, Hartmut Bauermeister†, Matthias Kahl‡,

Peter Haring Bolívar‡, Michael Möller†, Andreas Kolb*

Center for Sensor Systems (*ZESS*), University of Siegen, Germany

{tak.wong, hartmut.bauermeister, matthias.kahl, peter.haring, michael.moeller, andreas.kolb}@uni-siegen.de

In this supplementary material, we provide more details regarding training, ablation study on the network architecture, experimental setup and evaluation. We also extend figures from the main paper with more qualitative comparisons.

In this paper and supplementary material, all computation time are recorded by NVIDIA® GTX 1080Ti GPU, using PyTorch 1.9.0 version. The source code is available at https://github.com/tak-wong/Deep-Optimization-Prior.

1. Training of Deep Optimization Prior

Algorithm 1 Training procedure of deep optimization prior

Input: Data tensor G, forward model A, z-direction sampling grid $\vec{z}$, iteration M

function AUTOENCODER($G, A, \vec{z}, M$)
  $\mathcal{N}$, θ = NN(); ▷ initialize a neural network model
  for $i = 1$ to M **do**
    $u = \mathcal{N}(G; \theta)$; ▷ network prediction
    Model = $A(u|\vec{z})$; ▷ physical model
    Loss = $\mathcal{L}(\text{Model}, G)$; ▷ loss function
    θ = $\mathcal{N}$.train(Loss); ▷ backpropagation
  end for
return θ, u
end function

In contrast to the classical *training-then-prediction* approach by per-pixel autoencoder, we propose to utilize a *deep optimization prior* training approach which trains a neural network as an optimizer. The pseudo-code of the unsupervised deep prior training procedure of the proposed method is shown in Algorithm 1, where the inputs are the measurement data tensor G, the physical forward model A, depth z-direction sampling vector $\vec{z}$, the number of iterations (*i.e.* epochs) M. Note that unlike the random sampled per-pixel approach in [8], the network $\mathcal{N}$ is trained, *i.e.* optimized, to predict parameters u at all lateral pixel (x, y) based on the entire 4-D tensor G.

2. Ablation Study on the Network Architecture

The introduction of concatenating skip connections in U-net that often exceed the number of channels as well as the additional intermediate bottleneck in the encoder appear unintuitive from a supervised learning perspective. Thus, let us investigate their effect by considering the standard U-net (*Unet*), a U-net with large skip-connections by a standard encoder (*Unet+Skip*), a U-net with standard skip-connection but the encoder we proposed (*Unet+Bottleneck*), and our proposed U-net architecture. Table 1 shows their optimal performance on the real measurement *MetalPCB* dataset and on the synthetic datasets *SynthUSAF* and *SynthObj* at various noise level respectively. All network architecture are trained for 4 learning rates using the real measurement *MetalPCB* dataset respectively. This optimal learning rate is applied for all datasets.

As we can see, the large skip connection Unet (*Unet+Skip*) and the proposed Unet architectures achieves the lowest loss for almost all datasets. However, the variance of loss by *Unet+Skip* is significantly higher than the proposed Unet for some datasets (*e.g. MetalPCB*, *SynthUSAF* at 0dB, *SynthObj* at -10dB). It indicates that the proposed Unet architecture is the most robust architecture among these 4 network structures.

*Computer Graphics and Multimedia Systems Group

†Computer Vision Group

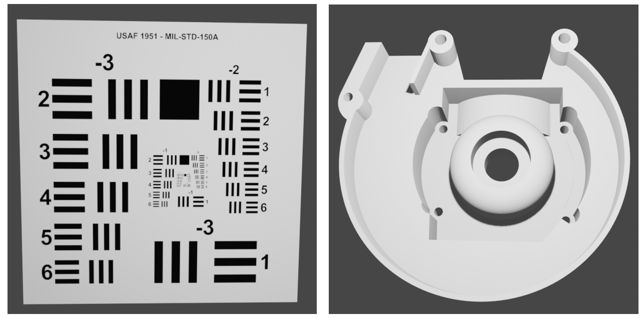
‡Institute for High Frequency and Quantum Electronics (*HQE*)

Noise Level (PSNR)	Normalized Loss ($\times 10^{-6}$)				
	Network	**Unet**	**Unet+Skip**	**Unet+Bottle**	**Proposed**
MetalPCB					
measured	**Optimal LR**	0.1	0.1	0.01	0.01
measured	**Min.**	58.87	**56.55**	60.16	56.78
	Mean	62.95	57.63	63.81	**57.56**
	Max.	69.33	61.20	67.04	**58.25**
MetalPCB+AWGN					
-20dB	**Min.**	32365.39	30851.79	32064.11	**30797.56**
	Mean	47598.25	35825.00	32121.31	**30871.59**
	Max.	58154.44	54943.34	32196.63	**30918.26**
-10dB	**Min.**	3303.06	**3264.36**	3311.45	3267.41
	Mean	3362.96	8401.39	3369.38	**3271.89**
	Max.	3465.76	28943.87	3431.25	**3278.23**
0dB	**Min.**	422.88	**395.64**	415.69	397.34
	Mean	5552.23	5523.62	423.50	**400.09**
	Max.	26033.73	26032.38	428.74	**403.63**
10dB	**Min.**	117.14	**109.12**	118.03	109.69
	Mean	5241.37	**109.93**	121.15	111.22
	Max.	25729.74	**110.91**	124.82	113.76
SynthUSAF+AWGN					
-20dB	**Min.**	31145.93	29799.38	30948.27	**29746.34**
	Mean	37539.14	44513.64	31022.22	**29802.03**
	Max.	62445.44	66699.85	31080.66	**29819.02**
-10dB	**Min.**	3216.02	**3020.85**	3196.11	3042.54
	Mean	10928.21	**3031.51**	10624.38	3058.49
	Max.	41705.13	**3050.10**	40263.39	3080.95
0dB	**Min.**	343.63	**313.34**	342.51	315.10
	Mean	345.74	517.48	347.60	**317.82**
	Max.	349.03	1324.67	354.28	**320.96**
10dB	**Min.**	43.01	**35.99**	54.54	38.18
	Mean	46.56	**38.94**	60.47	40.82
	Max.	48.75	**41.94**	73.15	45.26
SynthObj+AWGN					
-20dB	**Min.**	32059.66	29860.43	31813.17	**29668.97**
	Mean	38232.22	30048.30	31947.22	**29729.65**
	Max.	60389.10	30240.42	32187.62	**29823.52**
-10dB	**Min.**	3422.17	3095.52	3535.84	**3088.49**
	Mean	9481.55	10106.76	3633.48	**3276.31**
	Max.	33048.26	30638.85	3696.07	**3343.63**
0dB	**Min.**	406.96	323.21	387.25	**323.11**
	Mean	612.38	**381.05**	460.56	387.28
	Max.	885.45	**571.84**	673.18	588.90
10dB	**Min.**	75.92	**40.02**	90.64	48.85
	Mean	98.85	**101.67**	100.37	106.93
	Max.	**112.52**	317.85	121.63	289.59

Table 1: Comparison of normalized ℓ^2-squared loss by standard U-net architecture (*Unet*), standard encoder with large skip-connection (*Unet+Skip*), proposed encoder with standard skip-connection (*Unet+Bottleneck*) to the proposed U-net architecture. The best optimizers (lower is better) are highlighted.

3. Experimental Setup

In this section, technical details of experiment setup are described. Fig. 1a and Fig. 1b shows 3D objects that generate synthetic datasets *SynthUSAF* and *SynthObj* respectively.



(a) *SynthUSAF* (b) *SynthObj*

Figure 1: Synthetic dataset is generated by 3D objects from a USAF resolution target and an engine bearing part from MVTec ITODD dataset [2]

3.1. Choice of Optimizer

Commonly used optimization methods for THz inverse problem can be categorised as:

- Hessian based methods (second order gradient), which include Levenberg Marquardt [6], Trust Region Algorithm [1], and LBFGS [4].

- Gradient descent methods (first order gradient), which include gradient descent, and steepest gradient descent.

We select AdamW [5] method as a first order gradient descent method, and LBFGS as a second order method for comparison.

We optimize the deep optimization prior loss functions using the AdamW optimizer as implemented in PyTorch with GPU acceleration. To ensure a fair comparison we phrase the classical optimization as the minimization of a "network" that does not receive any input node, but instead only outputs the learnable parameters u to avoid any differences in implementation. As a second baseline, we additionally evaluate the LBFGS [4] optimizer for the classical approach to exclude an systematic advantage of the specific AdamW method for optimization problems with a deeply nested structure. All formulations and optimizers are run for 1200 iterations (i.e. *full-batch epochs* in machine learning terminology). Moreover, we compare to the per-pixel autoencoder [8]. In order to have a fair comparison, we changed the optimization algorithm of the per-pixel autoencoder from Adam in [8] to AdamW.

To project parameters onto the non-negative orthant, the network predicted parameters $\mathcal{N}(G;\theta)$ are projected to $[u_{min}, u_{max}]$ using sigmoid function. Similarly, for LBFGS and AdamW optimizers, we project parameters u to the non-negative orthant, except that the linear bounded function is adopted instead of sigmoid function:

$$\hat{u}_{x,y} = \min\left(\max(u_{min}, u_{x,y}), u_{max}\right) \tag{1}$$

This is because by empirical comparison, the linear bounded projection function performs better than sigmoid function in terms of the minimized loss value.

However, because of the non-differentiable zero-point, the projection of LBFGS and AdamW optimizers is implemented after the gradient descent update for each iteration.

3.2. Initialization

Descent-based nonconvex optimization methods depend on their respective starting point, commonly known as their initialization. In our numerical experiments we evaluate two types of initializations for the classical approach of minimizing loss function $\mathcal{L}$ directly: The first is to choose every parameter u at every pixel from a uniform random distribution over $[u_{\min}, u_{\max}]$ where $u_{\min}$ and $u_{\max}$ are estimates of the reasonable minimum and maximum values these parameters should attain. As we will see in Section 4, such an initialization is too crude for classical optimization to yield reasonable results. Secondly, we try to exploit physical knowledge about each application in order to provide reasonably accurate initial guesses for the parameters at each pixel. In this paper, the initialization method in [7] is adopted as *physics based initialization*, and a *random parameter initialization* is tested for comparison.

Since classical approaches greatly benefit from a good initialization, we tried to benefit from good initial guesses for our network-based reparametrizations: by adding the initial parameters to the network prediction. However, this approach did not improve our numerical results in comparison to the usual random initialization of network parameters which is why we discarded this approach.

For the network initialization, we adopted initialization method from [3] for per-pixel autoencoder and the proposed method. In order to verify the robustness to random initialization, each setting that is related to random initialization of the model parameters or to random initialization of the network parameters is run for 5 times. Note that we run this sanity check for per-pixel autoencoder [8] to verify its robustness.

3.3. Hyperparameter Optimization

In THz imaging, the numerical value represents a specific physical meaning individually, *e.g*. THz time signal data represents reflective power at individual frequency and μ represents the physical depth position in terms of millimeters. In order to respect these physical meanings, we retain the original data scale for training and optimization. However, the large variety of numeric range leads to a diverging optimal hyperparameter for network training and optimizer.

To have a fair comparison, we optimize the hyperparameters via a grid search for 4 learning rates from 10^{-3} to 10^{0}, *i.e*. all optimizers and networks are trained and optimized for 4 different learning rates and for the measurement dataset *MetalPCB*, synthetic datasets (*SynthUSAF* and *SynthObj*) at $0dB$ PSNR noise level respectively as a reference optimal learning rate.

For the regularization coefficients λ, we empirically maximize the coefficients as long as no blurring of the parameter images based on visual inspection for all optimizers (LBFGS, AdamW and the proposed method) occurs for the shot noise model.

4. Evaluation on loss

Noise Level (PSNR)	Normalized Loss ($\times 10^{-6}$)						
	Optimizer	**PPAE**	**LBFGS**		**AdamW**		**Proposed**
	Initialization	**Random**	**Physics**	**Random**	**Physics**	**Random**	**Random**
MetalPCB							
measured	**Optimal LR**	0.001	0.01	0.1	0.001	0.01	0.01
measured	**Min.**	56.89	218.02	15008.87	61.32	12665.03	**56.78**
	Mean	3372.63	218.02	15465.89	61.32	12677.52	**57.56**
	Max.	16615.59	218.02	15904.82	61.32	12688.49	**58.25**
MetalPCB+AWGN							
-20dB	**Min.**	30800.43	39766.81	73667.70	36100.08	49546.92	**30797.56**
	Mean	34927.94	39766.81	105352.48	36100.08	49608.79	**30871.59**
	Max.	51007.51	39766.81	126876.18	36100.08	49643.10	**30918.26**
-10dB	**Min.**	**3217.44**	10488.53	59122.06	7380.64	21559.17	3267.41
	Mean	**3232.18**	10488.53	85814.26	7380.64	21591.09	3271.89
	Max.	**3252.32**	10488.53	107167.15	7380.64	21636.72	3278.23
0dB	**Min.**	**395.70**	1967.64	60312.11	965.00	18182.68	397.34
	Mean	408.61	1967.64	63289.90	965.00	18226.71	**400.09**
	Max.	436.22	1967.64	66334.08	965.00	18247.48	**403.63**
10dB	**Min.**	**108.96**	240.86	22642.91	135.92	17422.87	109.69
	Mean	112.16	240.86	27453.51	135.92	17439.26	**111.22**
	Max.	114.37	240.86	32264.10	135.92	17464.49	**113.76**

Table 2: Comparison of ℓ^2-squared loss using *MetalPCB* and *MetalPCB+AWGN* datasets by optimizers per-pixel autoencoder, LBFGS and AdamW to the proposed method. All optimizers are tested for 4 learning rate individually using *MetalPCB* dataset, and the corresponding optimal learning rates are shown in the first row. Then, this optimal learning rate is applied for different noise level. Note that the ℓ^2-squared loss is normalized by the signal power. The best optimizers (lower is better) are highlighted.

Table 2 shows ℓ^2-squared loss using *MetalPCB* and *MetalPCB+AWGN* datasets by optimizers per-pixel autoencoder, LBFGS and AdamW to the proposed method.

Table 3 shows ℓ^2-squared loss using *SynthUSAF* and *SynthObj* datasets at noise level $-20dB$ to $10dB$ AWGN by optimizers per-pixel autoencoder, LBFGS, AdamW and the proposed method.

Table 4 shows loss with regularization using *MetalPCB*, *SynthUSAF* and *SynthObj* datasets at $0dB$ AWGN and 10% shot noise respectively.

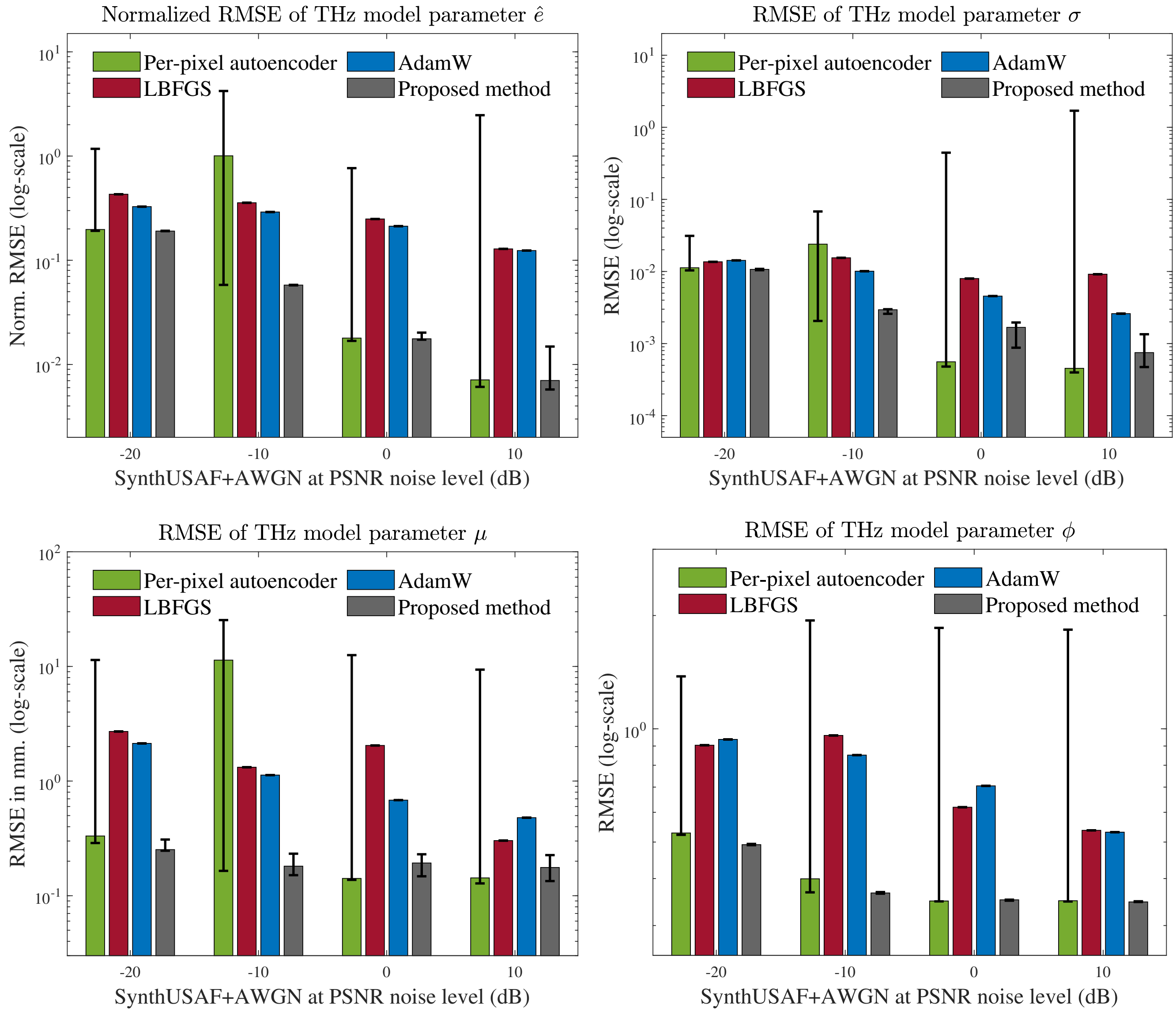


Figure 2: Comparison of RMSE of THz model parameters by optimizers per-pixel autoencoder, LBFGS and AdamW to the proposed method method using dataset *SynthUSAF+AWGN* at noise level from -20 to 10dB AWGN. Bars and whiskers indicate minimum, median and maximum RMSE among 5 runs for each optimizer respectively.

5. Evaluation on Parameter Accuracy

Numerical comparison Fig, 2 plots the RMSE of model parameters $\hat{e}$, μ, σ and ϕ estimated by per-pixel autoencoder, LBFGS, AdamW and the proposed method.

Fig, 3 plots the RMSE of model parameters $\hat{e}$, μ, σ and ϕ estimated by per-pixel autoencoder, LBFGS, AdamW and the proposed method.

Qualitative Comparison Fig. 4, 5, 6 and 7 show the corresponding model parameter images $\hat{e}$ (top row), μ, σ and ϕ (bottom row) of ground truth (first column) and estimation by per-pixel autoencoder, AdamW and the proposed method (last column).

Fig. 4 shows the model parameter images using *SynthUSAF* dataset with AWGN noise model at 0dB noise level, where the images are selected by median RMSE (*median quality run*) individually.

Fig. 5 shows the model parameter images using *SynthObj* dataset with AWGN noise model at 0dB noise level, where the images are selected by the highest RMSE (*worst quality run*) individually.

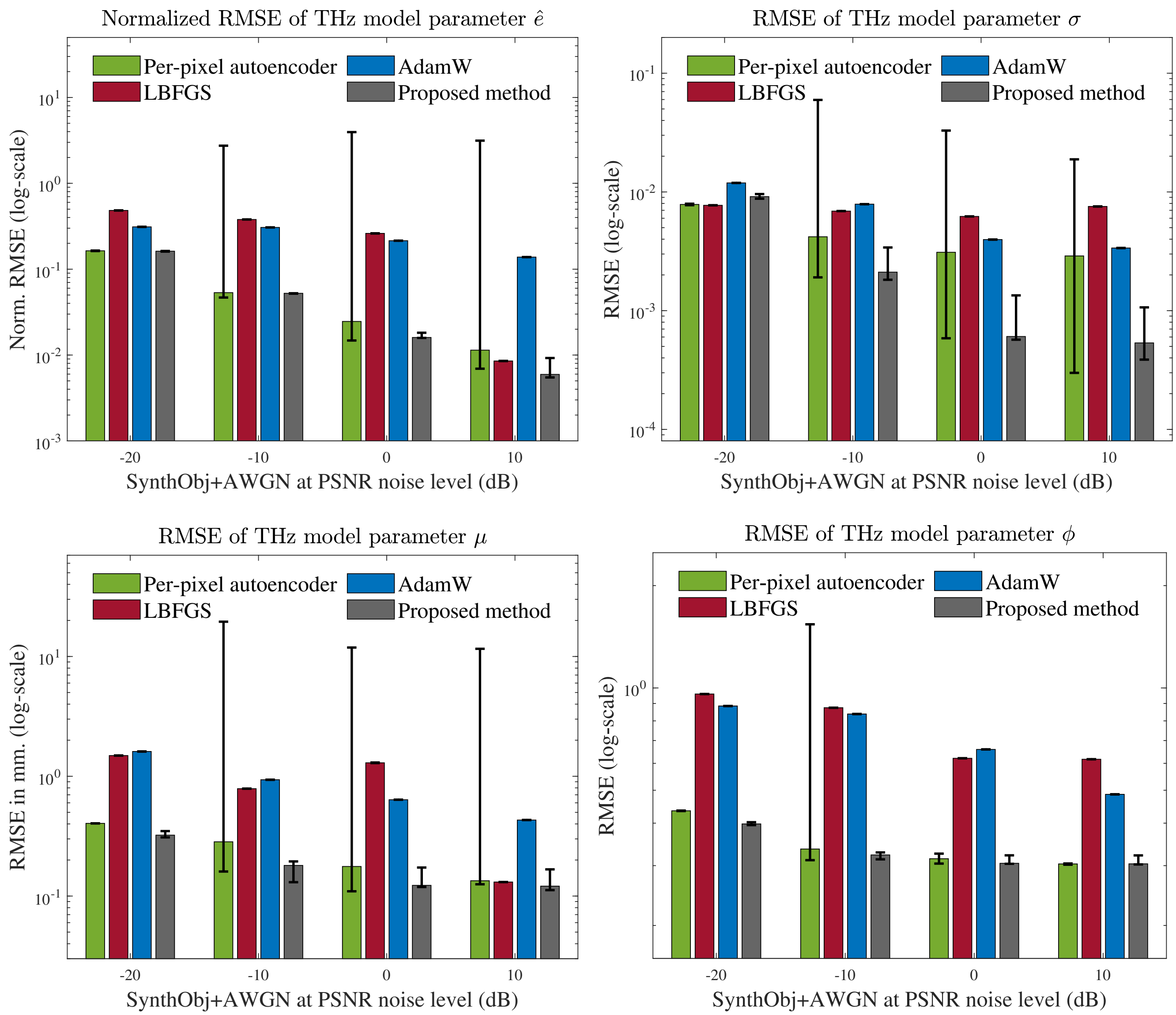


Figure 3: Comparison of RMSE of THz model parameters by optimizers per-pixel autoencoder, LBFGS and AdamW to the proposed method method using dataset *SynthObj+AWGN* at noise level from -20 to 10dB PSNR AWGN. LBFGS and AdamW are initialized by physics based initialization in Sec. 3. Bars and whiskers indicate minimum, median and maximum RMSE among 5 runs for each optimizer respectively.

Fig. 6 shows the model parameter images using *SynthUSAF* dataset with shot noise model at 0dB noise level, where the images are selected by median RMSE (*median quality run*) individually.

Fig. 7 shows the model parameter images using *SynthObj* dataset with shot noise model at 0dB noise level, where the images are selected by median RMSE (*median quality run*) individually.

References

[1] Thomas F Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on optimization*, 6(2):418–445, 1996.

[2] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing mvtec itodd-a dataset for 3d object recognition in industry. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2200–2208, 2017.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

Noise Level (PSNR)	Normalized Loss ($\times 10^{-6}$)				
	Optimizer	**PPAE**	**LBFGS**	**AdamW**	**Proposed**
	Initialization	**Random**	**Physics**	**Physics**	**Random**
SynthUSAF+AWGN					
0dB	**Optimal LR**	0.001	0.01	0.01	0.01
-20dB	**Min.**	**29498.45**	38624.05	35838.50	29746.34
	Mean	38843.38	38624.05	35838.50	**29802.03**
	Max.	65652.00	38624.05	35838.50	**29819.02**
-10dB	**Min.**	**3020.54**	15620.10	9243.03	3042.54
	Mean	19326.69	15620.10	9243.03	**3058.49**
	Max.	35783.14	15620.10	9243.03	**3080.95**
0dB	**Min.**	**312.73**	3730.37	3698.46	315.10
	Mean	8107.89	3730.37	3698.46	**317.82**
	Max.	39273.80	3730.37	3698.46	**320.96**
10dB	**Min.**	45.22	1220.81	1190.69	**38.18**
	Mean	7750.62	1220.81	1190.69	**40.82**
	Max.	38507.01	1220.81	1190.69	**45.26**
SynthObj+AWGN					
0dB	**Optimal LR**	0.001	0.001	0.01	0.01
-20dB	**Min.**	**30073.29**	49725.60	36259.96	29668.97
	Mean	30163.70	49725.60	36259.96	**29729.65**
	Max.	30254.10	49725.60	36259.96	**29823.52**
-10dB	**Min.**	**3091.60**	13370.01	10202.86	3088.49
	Mean	13342.05	13370.01	10202.86	**3276.31**
	Max.	28511.46	13370.01	10202.86	**3343.63**
0dB	**Min.**	**340.81**	7027.83	4032.16	323.11
	Mean	5170.08	7027.83	4032.16	**387.28**
	Max.	19096.94	7027.83	4032.16	**588.90**
10dB	**Min.**	44.90	16733.44	1711.49	**48.85**
	Mean	2741.44	16733.44	1711.49	**106.93**
	Max.	8798.07	16733.44	1711.49	**289.59**

Table 3: Comparison of ℓ^2-squared loss using *SynthUSAF+AWGN* and *SynthObj+AWGN* datasets by optimizers per-pixel autoencoder, LBFGS and AdamW to the proposed method. The optimal learning rate is selected by optimizing the loss by $0dB$ noise level. Note that the ℓ^2-squared loss is normalized by the signal power. The best optimizers (lower is better) are highlighted.

Noise Level (PSNR)	Normalized Loss $\times 10^{-6}$			
	Optimizer	**LBFGS**	**AdamW**	**Proposd**
	Initialization	**Physics**	**Physics**	**Random**
MetalPCB+ShotNoise				
0dB	**Learning Rate**	1	0.1	0.01
	Min.	2034.9	1670.4	**941.6**
	Mean	2034.9	1670.4	**967.3**
	Max.	2034.9	1670.4	**980.8**
SynthUSAF+ShotNoise				
0dB	**Learning Rate**	1	0.1	0.01
	Min.	10953.1	5036.5	**4819.6**
	Mean	10953.1	5036.5	**4831.2**
	Max.	10953.1	5036.5	**4842.9**
SynthObj+ShotNoise				
0dB	**Learning Rate**	1	0.1	0.01
	Min.	7771.8	4329.7	**4253.8**
	Mean	7771.8	4329.7	**4271.7**
	Max	7771.8	4329.7	**4289.1**

Table 4: Comparison of loss with regularization using *MetalPCB+ShotNoise*, *SynthUSAF+ShotNoise* and *SynthObj+ShotNoise* datasets by optimizers, LBFGS and AdamW to the proposed method. The learning rate is selected based on optimal learning rate of *MetalPCB+ShotNoise* using $0dB$ PSNR noise level. Note that the loss is normalized by the signal power. The best optimizers (lower is better) are highlighted.

Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[4] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018.

[6] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[7] Tak Ming Wong, Matthias Kahl, Peter Haring Bolívar, and Andreas Kolb. Computational image enhancement for frequency modulated continuous wave (fmcw) thz image. *Journal of Infrared, Millimeter, and Terahertz Waves*, 40(7):775–800, 2019.

[8] Tak Ming Wong, Matthias Kahl, Peter Haring Bolívar, Andreas Kolb, and Michael Möller. Training auto-encoder-based optimizers for terahertz image reconstruction. In *German Conference on Pattern Recognition*, pages 93–106. Springer, 2019.

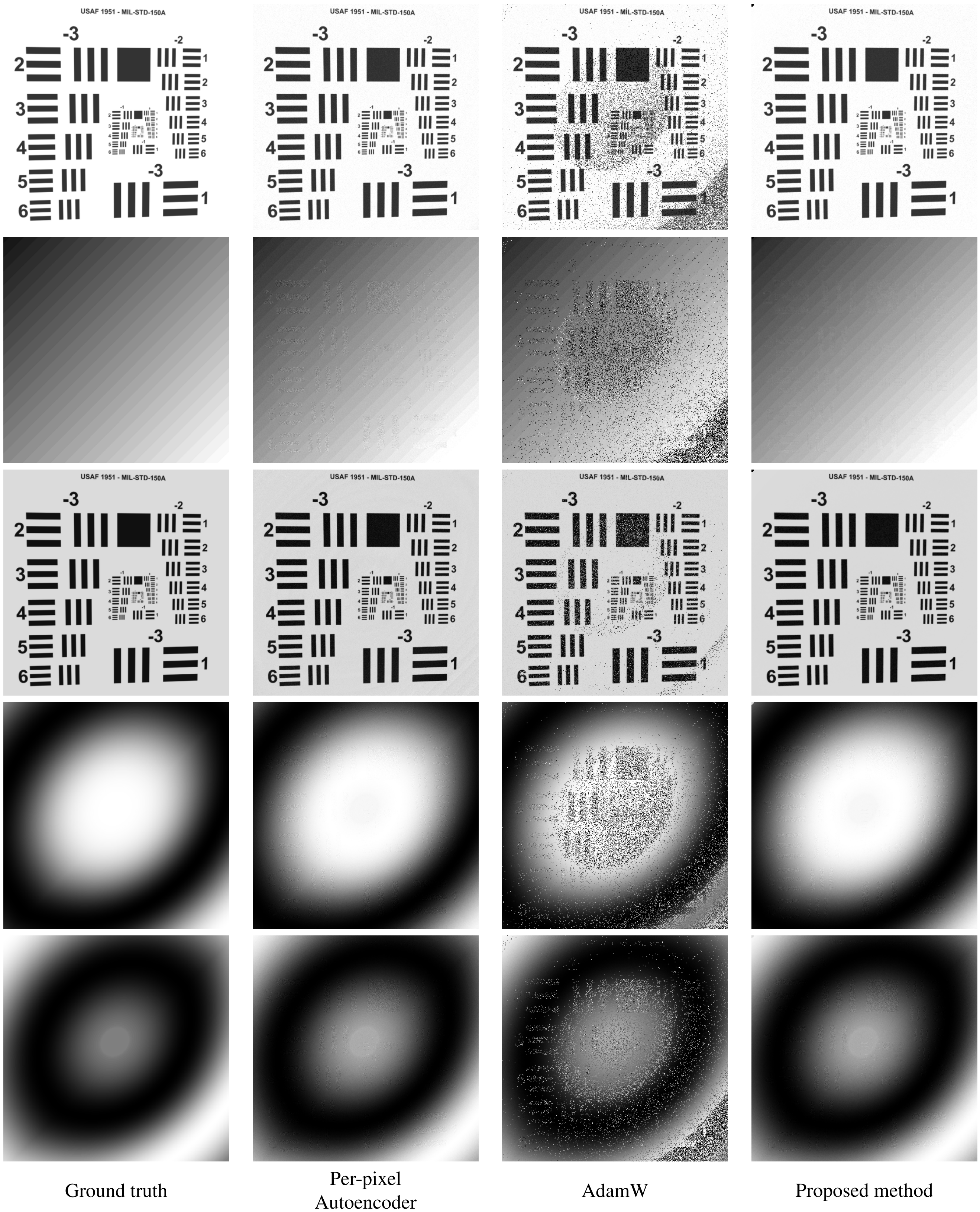


Figure 4: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, per-pixel autoencoder, AdamW and the proposed method using dataset *SynthUSAF+AWGN* at $0dB$ PSNR. All images are selected by the median RMSE (*median quality run)* among 5 runs respectively.

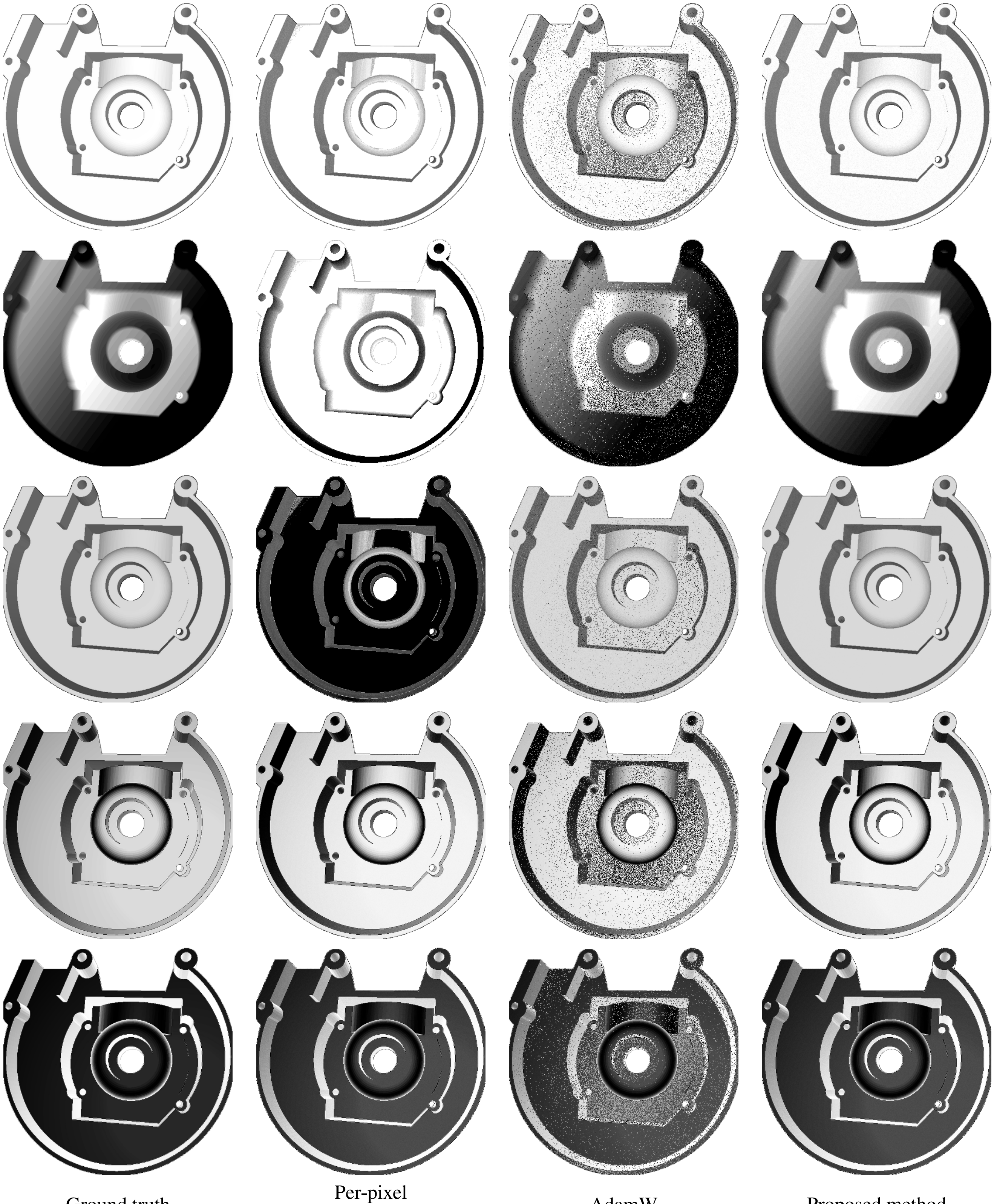
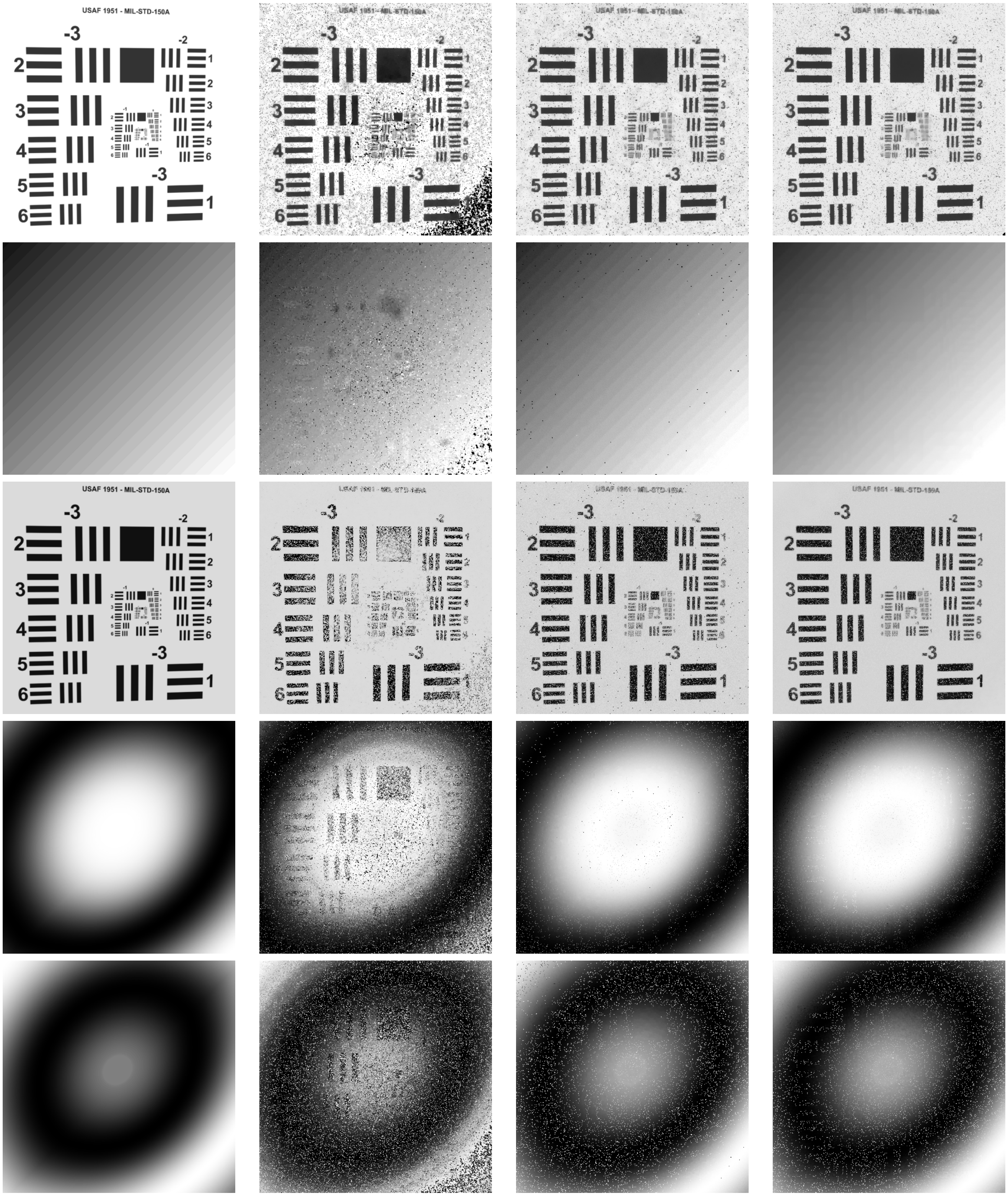


Figure 5: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, per-pixel autoencoder, AdamW and the proposed method using dataset *SynthObj+AWGN* at $0dB$ PSNR. All images are selected by the maximum RMSE among 5 runs respectively (*worst quality run*).



Ground truth | LBFGS | AdamW | Proposed method

Figure 6: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, LBFGS, AdamW and the proposed method using dataset *SynthUSAF+ShotNoise* at $0dB$ PSNR. All images are selected by the median RMSE (*median quality run)* among 5 runs respectively.

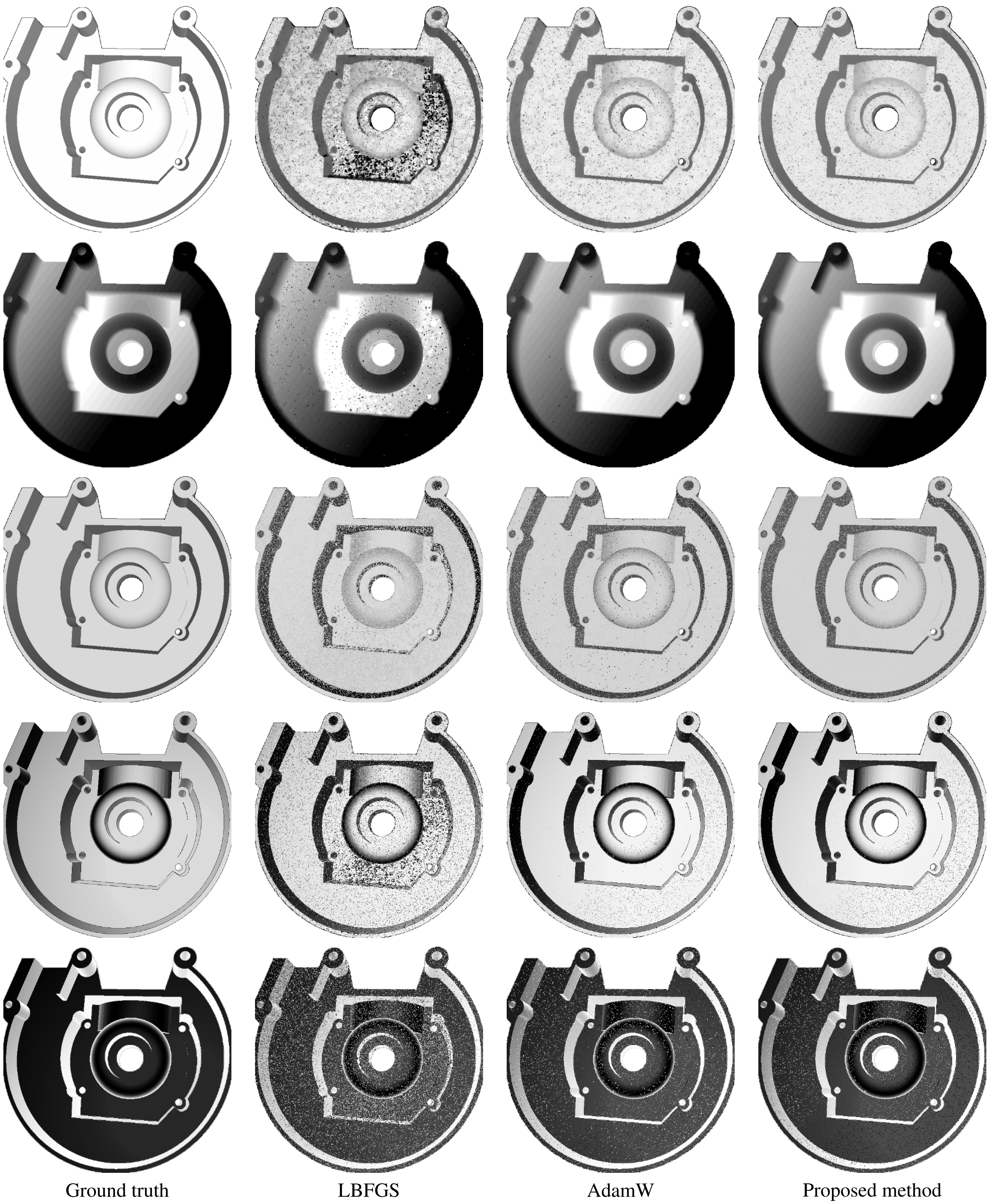


Figure 7: Comparison of model parameters $\hat{e}$ (first row), μ (second row), σ (third row), $\cos(\phi)$ (fourth row) and , $\sin(\phi)$ (last row) by ground truth, LBFGS, AdamW and the proposed method using dataset *SynthObj+ShotNoise* at $0dB$ PSNR. All images are selected by the median RMSE (*median quality run)* among 5 runs respectively.