FlatNet: Towards Photorealistic Scene Reconstruction from Lensless Measurements —Supplementary—

Abstract—In this supplementary material, we provide some additional details. We provide details about the display captured setup, the qualitative performance of FlatNet-gen-UC on both cropped and full measurements, the variation of performance of the deep networks with respect to the number of parameters, additional detail on the trainable inversion stage, the performance of FlatNet-gen finetuned on unconstrained cropped indoor captures and the performance of both FlatNet-sep and FlatNet-gen on scenes containing bright objects.

Index Terms—lensless imaging, image reconstruction

1 Display Capture Setup

To capture a display-captured image using FlatCam [1] and PhlatCam [2], the image is resized so as to occupy the biggest central square on a 24-inch monitor using bicubic interpolation. The monitor was placed at appropriate distance so that the image occupied the field of view of the cameras. For FlatCam, this was around 1 foot, while for PhlatCam, this was around 16 inches. This setup is fixed for all image captures such that the alignment of the monitor pixels to the camera pixels is uniform throughout both training and test. The white balance setting for FlatCam is fixed to be the white balance setting obtained in the FlatCam's (i.e. PointGrey Flea3) automatic white balance mode when an all-white image is displayed on the monitor. The exposure time is set to PointGrey's automatic mode, and the camera's gain is set to 0dB. For PhlatCam prototype using a Basler ace camera, the white balance setting was estimated once before the capture began by capturing a demo picture. The exposure was set at 10000 microseconds. Figure 1 shows the setup for FlatCam capture. The setup for PhlatCam is similar.

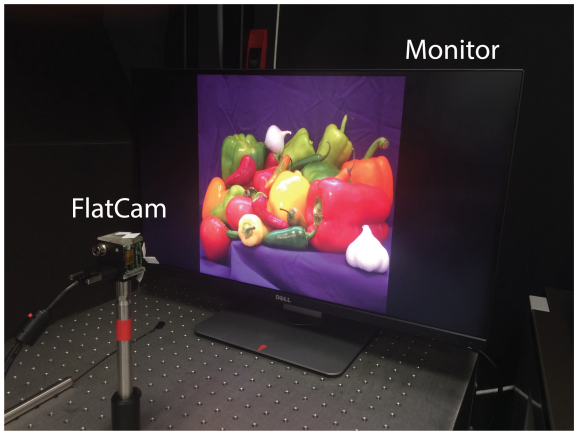


Fig. 1. **The display capture setup for FlatCam.** A similar setup was used for PhatCam.

2 Qualitative Comparison for Uncalibrated PSF Case

In Section 4.3.2 and 4.4.1 of the main paper, we provided the quantitative comparison for FlatNet-gen with Le-ADMM and Tikh+U-Net. In this section, we provide the visual results for the uncalibrated versions of the same. In particular, we use PSF simulated using the method described in Section 3.1.2 and use this PSF for learning Le-ADMM, Tikh+U-Net and FlatNet-gen. We provide the comparison for both full measurement in Figure 2 and cropped measurement in Figure 3. We can see clearly that the performance of FlatNet-gen-UC is very close to its calibrated counterpart i.e. FlatNet-gen-C. However, this is not the case with Le-ADMM and Tikh+U-Net.

3 Effect of Parameters on Performance of FlatNet-gen

In this section, we investigate how FlatNet-gen compares against Le-ADMM and Tikh+U-Net in terms of performance for different parameter count. In particular, we train FlatNet-gen, Tikh+U-Net and Le-ADMM for different variants of U-Net, keeping the number of learnable parameters constant in the trainable inversion stage for FlatNet-gen and unrolled ADMM block for Le-ADMM. U-Net-N refers to the variant of U-Net for which the number of filters in a convolutional block increases from N to 8N and reduces back to N. We perform this experiment for N = 32, 64 and 128. Table 1 provides the variation of the average PSNR and LPIPS for Tikh+U-Net, Le-ADMM and FlatNet-gen against the total number of learnable parameters. It is clear that FlatNet-gen outperforms both Tikh+U-Net and Le-ADMM for different parameter counts at the cost of slight increase in the relative number of learnable parameters. In the main text, we report the best model for each approach i.e. with U-Net-128.

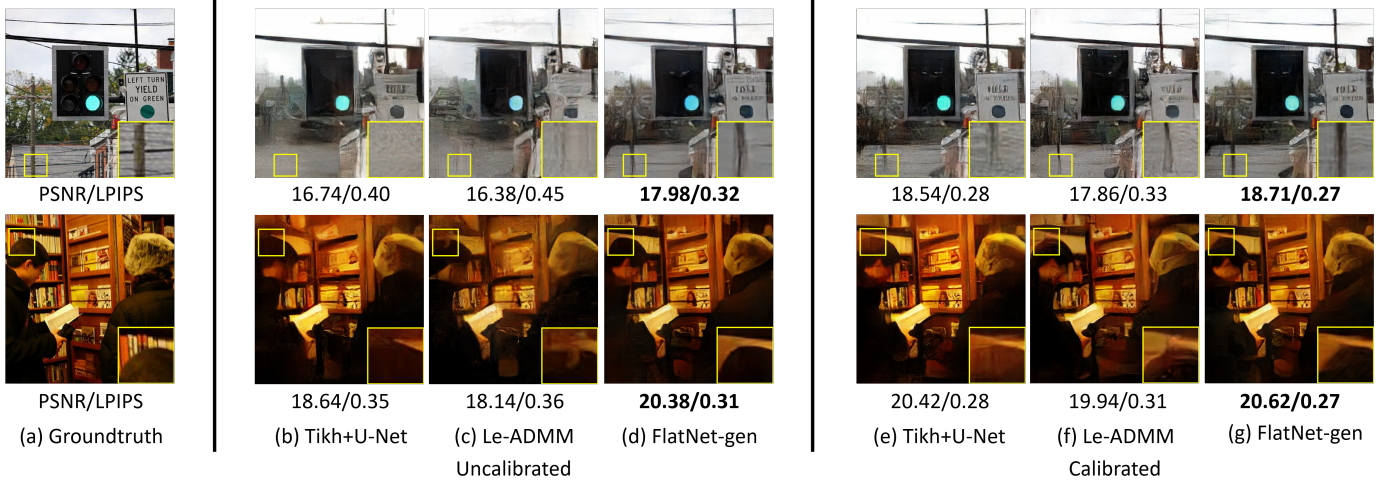


Fig. 2. **Comparison between uncalibrated and calibrated learning based approaches for full PhlatCam measurement**. Tikh+U-Net and Le-ADMM rely on accurate estimation of PSF while FlatNet-gen relies on PSF only for initialization and rather learns the inverse of the PhlatCam forward model. FlatNet-gen higher quality reconstructions with finer details for both calibrated and uncalibrated case. This is not the case for Le-ADMM or Tikh+U-Net.

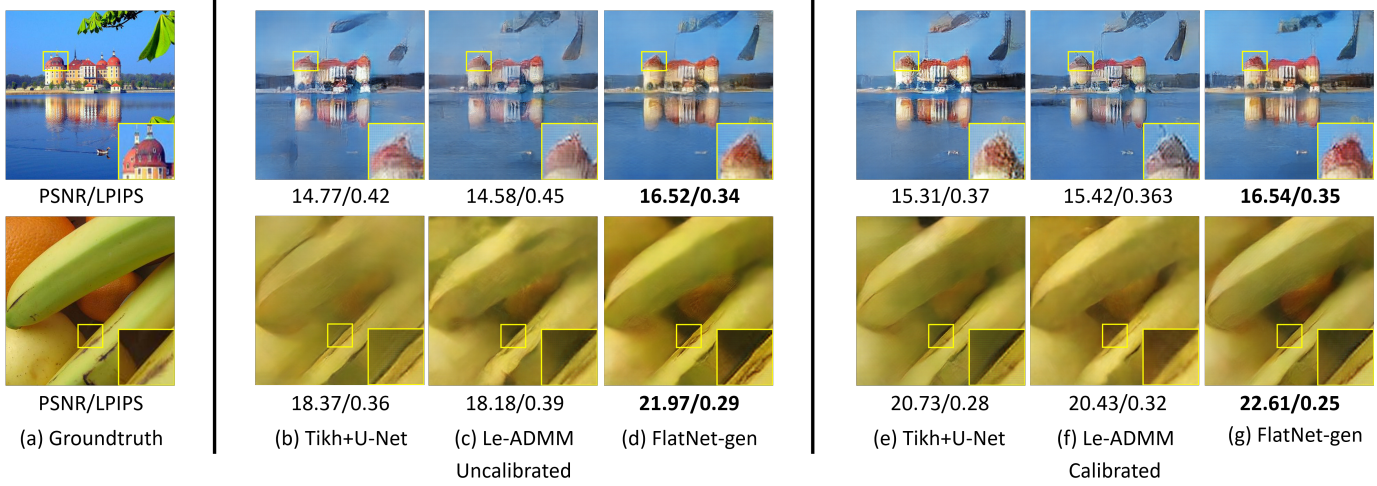


Fig. 3. **Comparison between uncalibrated and calibrated learning based approaches for cropped PhlatCam measurement**. FlatNet-gen provides higher quality reconstruction for both calibrated and uncalibrated case even when the measurement is extensively cropped. This indicates that FlatNet-gen can be used for small sensor setup without accurately estimating the PSF.

4 Details of Trainable Camera Inversion

In this section, we provide some additional details regarding the trainable camera inversion stage.

4.1 Initial weights in FlatNet-sep

The dimensions of W_1 and W_2 are 256×500 and 620×256, given that the measurement dimensions are $500 \times 620 \times 4$ and the reconstruction dimensions are 256×256. For calibrated initialization of FlatNet-sep, we use Φ_L^T to initialize W_1 and Φ_R to initialize W_2. Similarly, for the uncalibrated initialization, we first generate random toeplitz matrices of slope that matches that of Φ_L and Φ_R. Once these matrices are generated, they are used for initialization in a way similar to the calibrated case i.e. the transpose of the random toeplitz matrix corresponding to the Φ_L is used to initialize W_1 and the random toeplitz matrix corresponding to Φ_R is used to initialize W_2. Figure 4 presents a visual representation of how the initialized weights W_1 and W_2 look for both calibrated and uncalibrated case.

4.2 Generation of random toeplitz matrices for FlatNet-sep

For this subsection, please refer to Figure 5 which provides a 1-D version of the geometry we are considering. Let us assume that a scene of dimension $H \times W$ fills up the entire FoV of the camera and the the scene is discretized into $h \times w$ dimensional pixels. The corresponding scene maps to a

TABLE 1
Variation of performance against the total number of learnable parameters. FlatNet-gen outperforms both Le-ADMM and Tikh+U-Net under all parameter counts.

Methods	PSNR (in dB)	LPIPS	Learnable Parameters
Tikh+U-Net			
U-Net-32	18.74	0.384	2.4M
U-Net-64	19.83	0.341	12.9M
U-Net-128	20.60	0.298	51.5M
Le-ADMM			
U-Net-32	15.72	0.448	2.4M
U-Net-64	17.20	0.407	12.9M
U-Net-128	20.29	0.333	51.5M
FlatNet-gen			
U-Net-32	18.83	0.379	4.2M
U-Net-64	19.92	0.336	14.7M
U-Net-128	**20.94**	**0.296**	**53.3M**

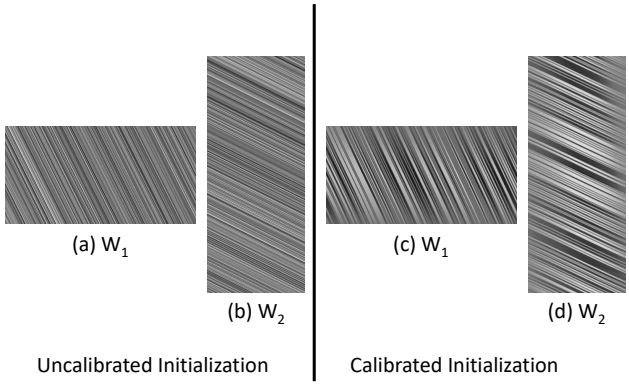


Fig. 4. **Initialized trainable inversion weights for FlatNet-sep**. (a) Initialized W_1 for uncalibrated case. (b) Initialized W_2 for uncalibrated case. (c) Initialized W_1 for calibrated case. (d) Initialized W_2 for calibrated case.

region of dimension $M \times N$ in the sensor and the sensor has a pixel pitch of p. The slope of the calibration matrix Φ_L is then defined as follows,

$$m_L = \frac{H/h}{M/p} \tag{1}$$

This slope measures the ratio of the number of pixels (row) in the scene to the number of pixels (rows) in its projection at the sensor or in other words, how many row pixels in the scene correspond to a row pixel at the sensor.

If we assume our monitor for calibration or data capture is at z distance from the camera and the mask to sensor distance is d, then,

$$M = Hd/z \tag{2}$$

Plugging 2 into 1, and assuming a scene of dimension $P \times Q$ pixels (i.e. $(H/h) \times (W/w)$), the slope for Φ_L becomes

$$m_L = \frac{P}{Hd/(pz)} \tag{3}$$

Similarly, the slope for Φ_R can be shown to be,

$$m_R = \frac{Q}{Wd/(pz)} \tag{4}$$

For the FlatCam prototype we use in the experiments, $p = 10.6\mu m$ (this pixel pitch is for each channel and is therefore twice the actual pixel pitch of the sensor) and $d = 1.5mm$. We placed the monitor at a distance $z = 31.75cm$ and projected on the screen, a scene of dimension $H \times W = 29cm \times 29cm$. If we assume our scene reconstruction to be of size $P \times Q = 256 \times 256$ pixels, then $m_L = m_R \approx 2$.

To generate toeplitz matrix of shape $S \times P$ where $P < S$ and with a slope that matches that of Φ_L, we first generate a random vector of length S and form a circulant matrix of dimension $S \times S$ corresponding to it. Then, using bilinear/nearest-neighbor interpolation, we resize this circulant matrix to $S \times m_L S$. We then arbitrarily crop a submatrix of size $S \times P$ from the resized matrix to match the dimension of Φ_L. Similar process is followed for generating a toeplitz matrix that matches the dimension and slope Φ_R as well. Figure 6 shows an example toeplitz matrix along with the calibrated Φ_L matrix and the generated slope-matched random toeplitz matrix after estimating the slope using 3.

4.3 Evolution of the parameters

Figure 7 shows the evolution of trainable inversion parameters for both FlatNet-sep and FlatNet-gen. Specifically, we plot the product $W_1\Phi_L$ for FlatNet-sep and the convolution output $\mathcal{F}^{-1}(\mathcal{F}(W) \odot H)$ for FlatNet-gen. Here, H is the Fourier transform of the PSF. For, FlatNet-sep the product is an identity matrix while the convolution output for FlatNet-gen is close to an impulse, indicating that the trainable camera inversion has learned to invert the forward process. The effect of learning is more prominent in FlatNet-sep compared to FlatNet-gen for two reasons: (a) the weights W_1 and W_2 were initialized with adjoint of Φ_L and Φ_R as compared to the pseudo-inverse of the PSF in case of FlatNet-gen, (b) owing to the superior mask properties of PhlatCam, the pseudo-inverse of the PSF is of high quality already. Similarly, the effect of learning is more prominent in case of uncalibrated initialization for FlatNet-gen compared to the calibrated counterpart. This is again due to the fact that pseudo-inverse of the calibrated PSF accurately inverts the forward model while the pseudo-inverse of the simulated PSF is unable to capture some of the non-idealities of the capturing process. As a result, it gets refined through learning to accurately invert the forward model. The prominence of the inversion stage learning in FlatNet-gen, however, is evident in the case of cropped measurements (main text section 4.4.2), as shown in Figure 8. It can be seen that learning gets rid of majority of the artifact making it easier for the perceptual enhancement to extract meaningful features that help with higher quality final reconstruction.

4.4 Additional intermediate reconstructions

In this subsection, we present more intermediate results for Le-ADMM, FlatNet-gen and FlatNet-sep. In Figure 9, we show the intermediate outputs for three scenes by Le-ADMM, FlatNet-gen and FlatNet-sep. The intermediate output for Le-ADMM corresponds to the output of the unrolled ADMM block while that for FlatNet corresponds to the output of the trainable inversion block. For the non-separable models (Le-ADMM and FlatNet-gen), we show

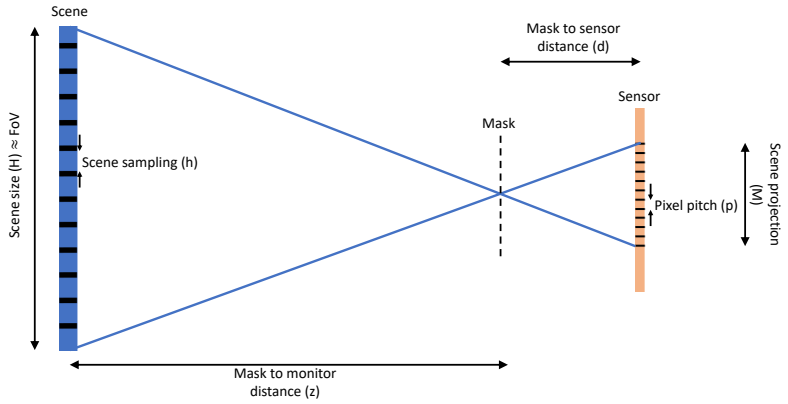
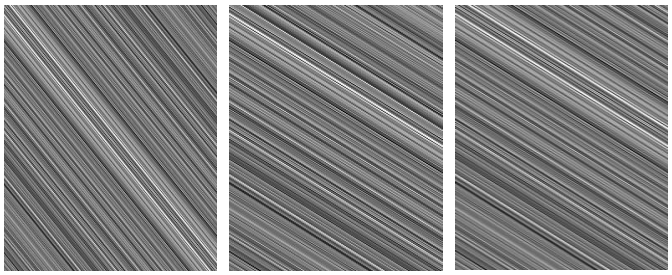


Fig. 5. **Geometry considered for the generation of random toeplitz matrices**. 1-D representation of the geometry considered for the generation of matrices for uncalibrated FlatNet-sep.



(a) Toeplitz Matrix (b) Calibrated Matrix Φ_L (c) Slope-matched

Fig. 6. **Generation of slope-matched random toeplitz matrices**. (a) Shows a conventional random toeplitz matrix. (b) Shows the calibrated matrix Φ_L. (c) Shows the slope-matched random toeplitz matrix. The slope in (c) matches to that in (b).

the intermediates for both cropped and full measurement. We can clearly see that learning has significant impact on the intermediates especially for FlatNet-gen trained on cropped measurements.

5 Reconstruction of unconstrained indoor scenes for small sensor

It is interesting to observe the effectiveness of the finetuned FlatNet for cropped unconstrained indoor scenes. In Figure 10, we provide visual comparison for the reconstructions from cropped measurement and full measurement along with the webcam capture. We show result for crop sizes of 990×1254. It should be noted that in an unconstrained setup, there may be large signals (due to bright objects) outside the field of view described by the CRA which would result in strong line artifacts in the reconstructions produced by model without finetuning.

6 Effect of Bright Object

For a highly multiplexed lensless imager, every pixel receives light from every point in the scene. Hence, if there is any really bright object (like a highly reflective object or a lamp) in the scene, the light from the object can dominate the pixel intensities and result in severe reconstruction artifacts on the dimmer objects. We show that, using FlatNet, the artifacts are minimized resulting in a higher quality reconstruction of the scene.

We show the bright object problem by introducing an LED into the scene. Figure 11 shows the reconstruction for FlatCam [1] and PhlatCam [2]. We can observe that FlatNet-sep and FlatNet-gen reconstructions have significantly fewer artifacts than other traditional and learning based approaches.

References

[1] M. S. Asif, A. Ayremlou, A. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk, "Flatcam: Thin, lensless cameras using coded aperture and computation," *IEEE Transactions on Computational Imaging*, vol. 3, no. 3, pp. 384–397, 2017.

[2] V. Boominathan, J. Adams, J. Robinson, and A. Veeraraghavan, "Phlatcam: Designed phase-mask based thin lensless camera," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

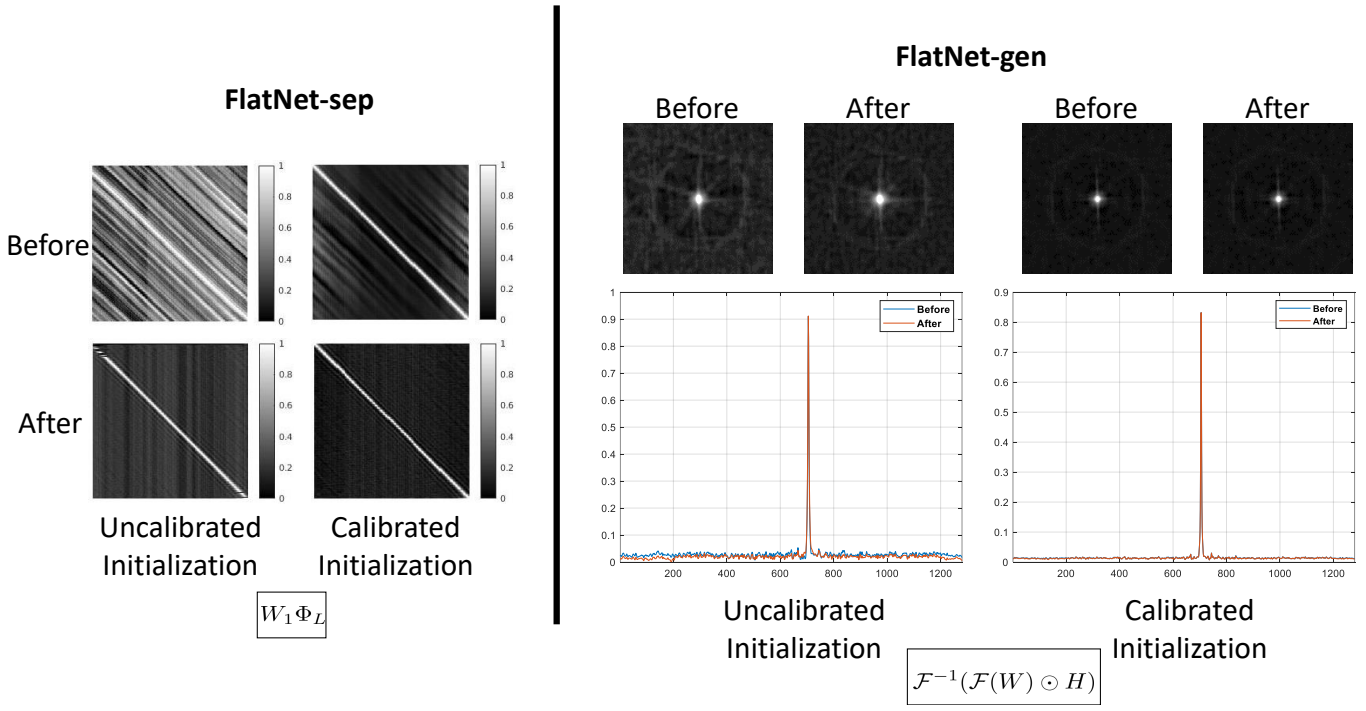
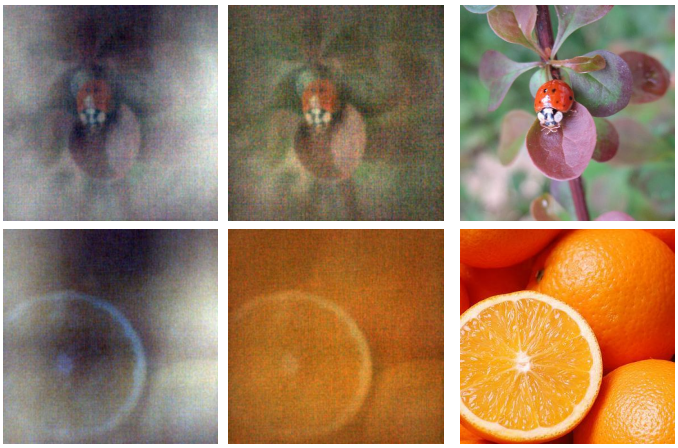


Fig. 7. **Evolution of trainable camera inversion stage**. Left: $W_1\Phi_L$ is shown as an image for both uncalibrated and calibrated scenario for the inversion layer of FlatNet-sep. Eventually, the product becomes an identity matrix, indicating that the learning has led to an inversion of the forward model for FlatNet-sep. Right: $\mathcal{F}^{-1}(\mathcal{F}(W) \odot H)$ is shown for the inversion stage of both uncalibrated and uncalibrated scenario for the inversion layer of FlatNet-gen. Here H is the Fourier transform of the PSF. Learning helps W in inverting the PSF resulting in the impulse shown in the top figures. The bottom row shows a horizontal slice from the impulse image. The effect of learning is more prominent in the case for uncalibrated FlatNet-gen compared to the calibrated counter part due to the superior nature of the mask and the resulting Wiener filter for the calibrated case.



(a) Before (b) After (c) Groundtruth

Fig. 8. **Evolution of trainable inversion output of FlatNet-gen for cropped measurement**.The effect of learning of trainable inversion is more prominent for cropped measurement as can be seen here. In (a) we show the trainable inversion output at the beginning of training and in (b) we show the trainable inversion output at the end of training. It can be observed that learning has removed a majority of the artifacts. (c) Groundtruth is also shown for reference.

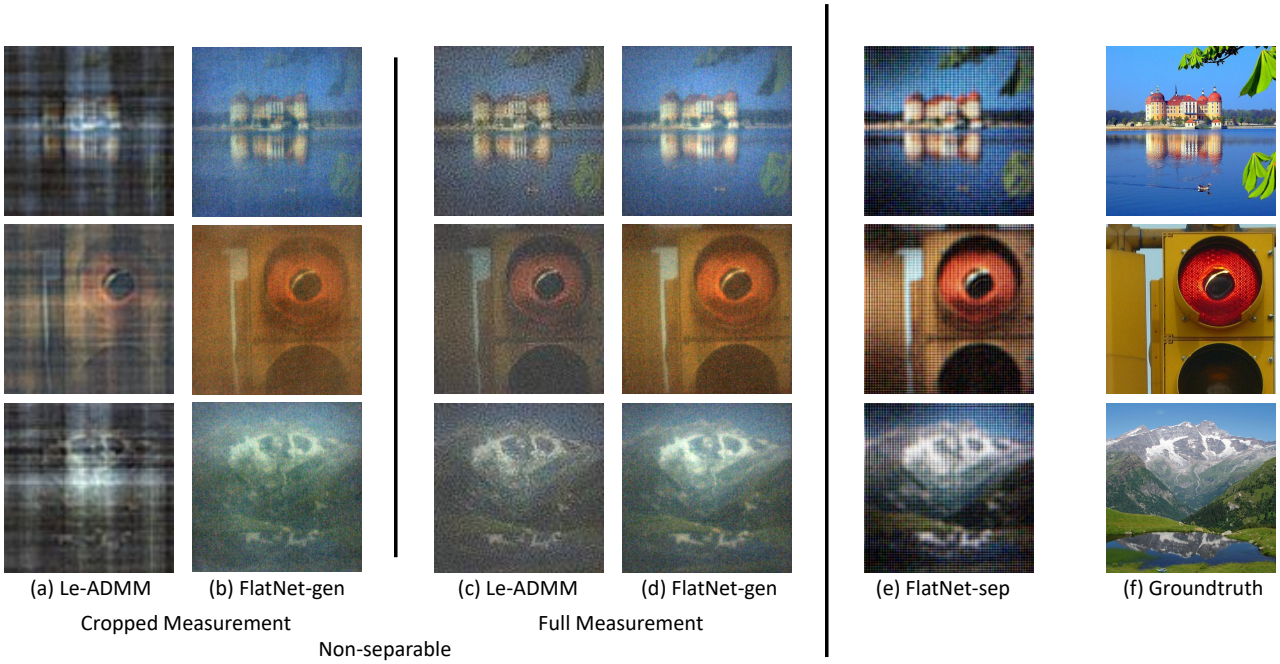


Fig. 9. **Intermediate outputs before perceptual enhancement block**. (a) Intermediate output of Le-ADMM for cropped measurement. (b) Trainable inversion output of FlatNet-gen for cropped measurement. (c) Intermediate output of Le-ADMM for full measurement. (d) Trainable inversion output of FlatNet-gen for full measurement. (e) Trainable inversion out of FlatNet-sep. (f) Groundtruth for reference.

Fig. 10. **Cropped measurements for Unconstrained Indoor Scenes**. We can observe that FlatNet-gen finetuned on unconstrained scenes provides reasonable reconstruction quality even for cropped measurements

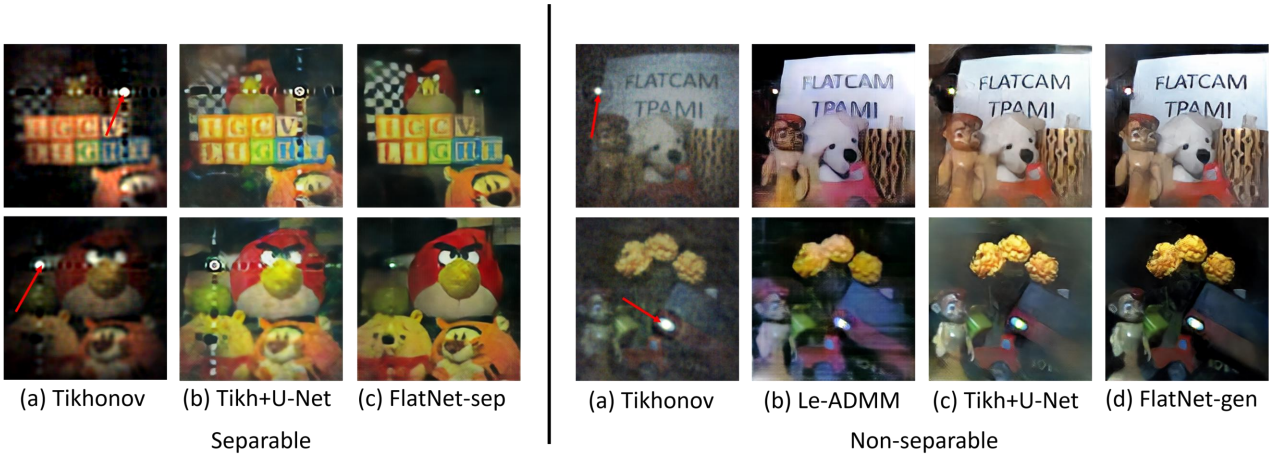


Fig. 11. **Reconstruction of scenes with bright objects (LED) using FlatCam and PhlatCam**. Artifacts occuring in Tikhonov reconstructions are amplified by Tikh+U-Net reconstruction. While Le-ADMM performs slightly better than Tikh+U-Net for PhlatCam, it is outperformed by FlatNet-gen