Supplementary Material for
Efficient Unified Demosaicing for Bayer and Non-Bayer Patterned Image Sensors

Haechang Lee[1,4,*], Dongwon Park[2,3,*], Wongi Jeong[1,*],
Kijeong Kim[4], Hyunwoo Je[4], Dongil Ryu[4] and Se Young Chun[1,2,3,†]
[1]Dept. of ECE, [2]INMC, [3]IPAI, Seoul National University, Republic of Korea,
[4]SK hynix, Republic of Korea
{harrylee,dong1park,wg7139,sychun}@snu.ac.kr {kijeong1.kim,hyunwoo.je,dongil.ryu}@sk.com

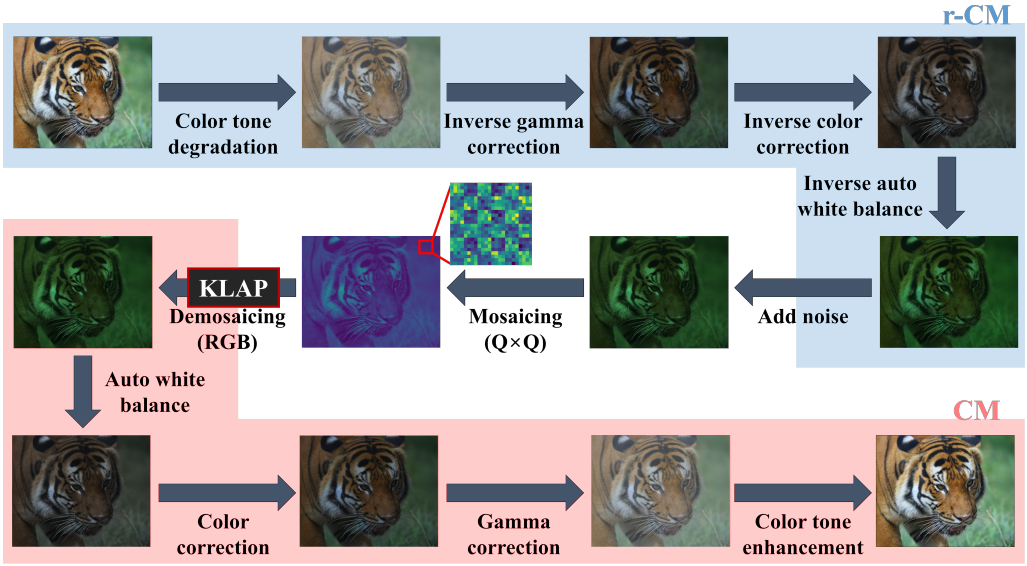


Figure S.1: Overview of our pipeline for synthesizing realistic RAW images, specifically for Q×Q patterns.

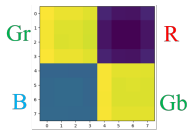
S.1. Detailed Data Synthesis for Demosaicing All CFAs

As described in our paper, we generate synthetic ground truth (GT) by sequentially applying a 4-step **r**everse **C**olor-related **M**apping (r-CM) process. Then, we add mixed Poisson and Gaussian noise and performed mosaicing (*i.e.*, CFA patterning) on the entire image to create synthetic RAW-like images (as shown in the blue shaded area in Fig. S.1). The r-CM process consists of the following modules: color tone degradation, inverse gamma correction, inverse color correction, and inverse auto white balance functions. The Color-related Mapping (CM) process is the inverse of the reverse color matrix (r-CM) and can only be applied to the output of the demosaicing (DM) model.

Note that we need to use r-CM for data synthesis on the open-source dataset to generate GT images, while CM can be "optionally" applied after demosaicing for better visualization in our paper.

Color tone degradation. Typically, color enhancement is performed in the latter part of the ISP chain. Therefore, we position the color tone degradation function at the beginning of r-CM. Inspired by [2], we employ a tone mapping function that utilizes a simple inverse smoothing curve for performing color tone degradation on open-source dataset images during the r-CM process. Note that the color tone enhancement function in CM is the inverse of color tone degradation in r-CM.

Inverse gamma correction. In the ISP chain, gamma correction is applied to image data to correct for the non-linear perception of brightness by the human eye. We use a gamma value setting of 2.2, which is standard for most cameras [6, 16, 24, 17]. In r-CM, the inverse function of gamma correction is applied, while in CM, standard gamma



Cumulative pixel distribution of each pixel unit (Gr, R, B, Gb) in 7 QxQ CIS RAWs

Figure S.2: The cumulative pixel value distribution of each homogeneous pixel unit (Gr, R, B, and Gb) in 7 Q×Q CIS RAW image samples. In our CIS RAW data, we observe a significant difference in signal values between inner and outer pixels in each Gr, R, B, and Gb pixel unit, which is mainly caused by crosstalk effect.

correction is performed.

Inverse color correction. We use a color correction function to adjust the colors captured by a camera's sensor to appear as they would to the human eye. The specific function we used is as follows:

$$\begin{pmatrix} R_{corrected} \\ G_{corrected} \\ B_{corrected} \end{pmatrix} = A \begin{pmatrix} R \\ G \\ B \end{pmatrix},$$

where A is a 3x3 color correction matrix (CCM), which is applied to the pixel values (R, G, and B) to obtain the corrected RGB values ($R_{corrected}$, $G_{corrected}$, and $B_{corrected}$). We obtain the CCM information from the CIS manufacturing company and apply it to our inverse color correction function after calculating the CCM's inverse.

Inverse auto white balance. We empirically adjusted the gains for R, G, and B channels in the auto white balance function to make white portions of the CIS RAW appear white as perceived by the human eye. The inverse auto white balance in r-CM is obtained by reversing the values applied in the white balance module of the CM process.

Noise synthesis. We use the following practical mixed Poisson and Gaussian noise model [19, 2, 25]:

$$\begin{aligned} x_n &= \text{Poisson}(\gamma y_n)/\gamma + \epsilon_n, \\ \epsilon &\sim \mathcal{N}(0, \sigma_\epsilon^2 I), \quad n = 1, \ldots, N, \end{aligned} \tag{1}$$

where y and x are clean image and corrupted image, respectively. Poisson generates pixel intensity-dependent Poisson noise caused by photon sensing, γ is a gain parameter which depends on the sensor and analog gain. ϵ is signal independent Gaussian noise with standard deviation σ, and N is the number of samples. DF2K-CIS train and test datasets are generated using the following imaging parameters: $\gamma = 0.01$ and $\sigma = 0.02$. DF2K-CIS with strong noise test dataset are generated with parameters that are 4 times larger than those of DF2K-CIS: $\gamma = 0.04$ and $\sigma = 0.08$.

S.2. Domain Gap Example: Inherent Grid Artifacts in CIS RAW

The differences in the distribution of pixels within each pixel unit are primarily caused by "crosstalk" effects, which result from mutual interference of each pixel signal in CIS hardware [9, 10, 13]. As shown in Fig. S.2, in CIS QxQ RAW (before demosaicing), we observe that the signals in the center of each pixel unit, especially in the R channel, are stronger than those in the outer pixels, while the edges of each pixel unit, particularly the four corners, are weaker. In addition to the cause of crosstalk phenomenon, the asymmetry between the inner and outer pixels in each pixel unit can vary across CIS devices, and this can manifest in various forms depending on the circuit configuration, component characteristics, product lines, and process capability of the CIS chip. These various sources result in differences in pixel values within each homogeneous color unit in CIS RAW, which is a major cause of the appearance of grid artifacts in CIS RAW.

S.3. Adaptive Discriminative Filter-based Model for Specific CFA Pattern (ADP)

S.3.1. Filter Attribution Integrated Gradients

Xie *et al.* [23] propose FAIG, which identifies discriminative filters of specific degradation in blind super-resolution (SR) by computing integrated gradient (IG) [22, 21] between the baseline and desired models. In FAIG, the baseline model is denoted as θ_{from} and the model being updated is denoted as θ_{to} for each desired task. The function $\rho(\beta)$, where $\beta \in [0, 1]$, represents an uninterrupted straight line between the baseline and target models. In that case, any certain route in $\rho(\beta)$ is represented by $\rho(\beta) = \beta\theta_{from} + (1 - \beta)\theta_{to}$, where $\rho(1) = \theta_{from}$ and $\rho(0) = \theta_{to}$. The FAIG on the continuous line space between two models is discretized as follows:

$$\begin{aligned} &\text{FAIG}_j(\theta_{from}, \theta_{to}, x) \\ &\quad \approx \left| \frac{1}{N} [\theta_{from} - \theta_{to}]_j \sum_{t=0}^{N-1} \left[\frac{\partial \mathcal{L}(\rho(\beta_t), x)}{\partial \rho(\beta_t)} \right]_j \right|, \end{aligned} \tag{2}$$

where N represents the total number of steps used in the integral approximation, and N is set to 100 as in FAIG. β_t and j are t/N and the kernel index, respectively. We apply FAIG, originally proposed for denoising and deblurring, to multiple CFA sensor patterns in our demosaicing tasks.

S.3.2. Mask Ratio of FAIG in ADP

We choose a mask ratio (q) as 1% in ADP for each CFA in our KLAP framework, to balance demosaicing performance and efficiency (as shown in Sec. 4.2 and Fig. 4(b)). Increasing q improves performance but with diminishing

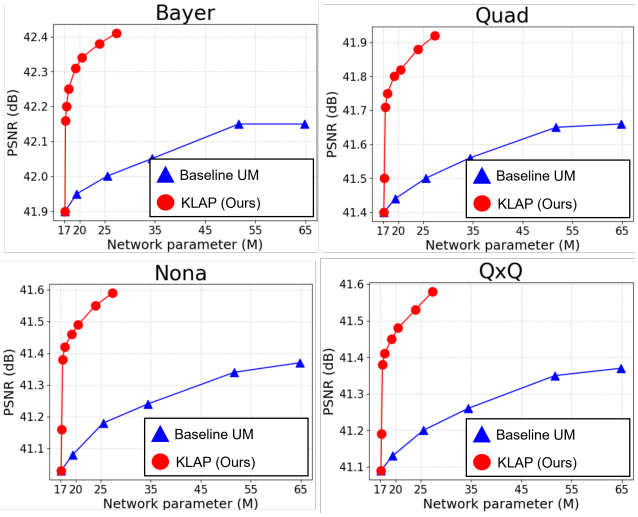


Figure S.3: Performance comparisons between Baseline UM with increased network sizes (17.1M, 19.4M, 25.5M, 34.4M, 51.7M, and 64.8M) and KLAP (Ours) with mask ratios $q\%$ (0%, 0.1%, 0.5%, 1%, 3%, 5%, 10% and 15%, respectively) on DF2K-CIS test dataset. The network size of KLAP (Ours) with mask ratios of $q\%$ (0%, 0.1%, 0.5%, 1%, 3%, 5%, 10%, and 15%) are 17.1M, 17.2M, 17.8M, 17.8M, 19.2M, 20.5M, 23.9M, and 27.4M, respectively. Our approach produces significantly higher performance results even with 3.5 times larger Baseline UM method.

returns and increased parameters (Tab. S.1). Compared to Baseline UM (B.UM), our proposed method using mask ratio 1% for all 4 demosaicing types requires an additional 4% of network parameters.

Furthermore, our KLAP achieves significantly better results even when increasing the size of the Baseline UM method by 3.5 times, as shown in Fig. S.3.

Table S.1: Investigation of experiments according to KLAP (Ours) with filter location selection ratios (*i.e.*, mask selection ratios in FAIG [23]; $q\%$) in the DF2K-CIS test dataset. B.UM denotes the Baseline UM. Note that Avg. and Par. denotes mean of all CFAs' PSNR (dB) and the required number of parameters (M).

Method	q	Ba.	Qu.	No.	QxQ	Avg.	Par.
B.UM	0	41.90	41.40	41.03	41.09	41.35	17.1
KLAP	0.1	42.16	41.50	41.16	41.19	41.50	17.2
KLAP	0.5	42.20	41.71	41.38	41.38	41.67	17.4
KLAP	1	42.25	41.75	41.42	41.41	41.71	17.8
KLAP	3	42.31	41.80	41.46	41.45	41.75	19.2
KLAP	5	42.34	41.82	41.49	41.48	41.78	20.5
KLAP	10	42.38	41.88	41.55	41.53	41.83	23.9
KLAP	15	42.41	41.92	41.59	41.58	41.87	27.4

S.4. Meta-learning during Inference

S.4.1. Definition of the term "meta-test"

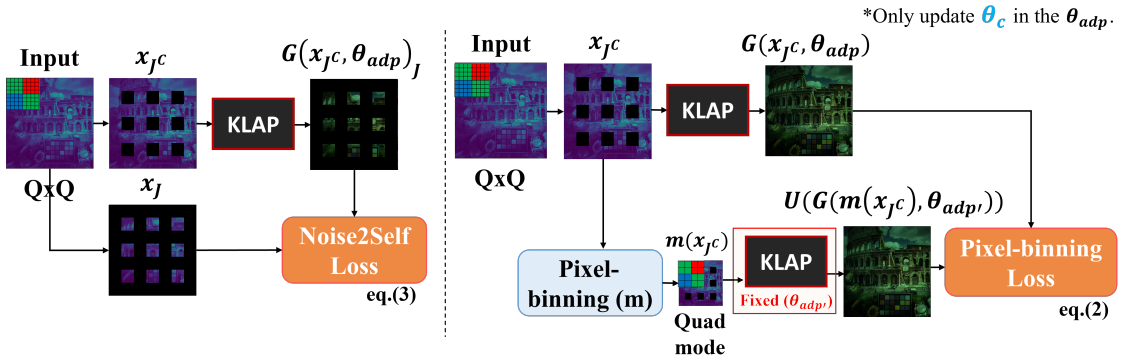
In our paper, we named the process of fine-tuning with meta-learning during inference as KLAP-M. The term "meta-test" typically refers to the process of improving performance on various generalization scenarios with only a few trials on unseen data [26, 7, 28, 14, 18, 20]. In general, the meta-test process works in conjunction with the meta-training process. The meta-training process optimizes the model to improve the accuracy of meta-test samples using source data. We use ADP in the second step of our KLAP framework to only adjust the unimportant kernel for each CFA demosaicing during training in order to improve the accuracy of meta-test. This can be seen as a type of meta-training process. In our paper, we define the process of fine-tuning only unimportant kernel in KLAP during model inference as meta-test learning to achieve robust results even for undefined artifacts caused by CIS device features and shooting environments.

S.4.2. Noise2Self and Pixel Binning Loss.

To aid in a more thorough understanding in our meta-test learning process, KLAP-M, we provide a more detailed explanation of Noise2Self (N2S) [1] loss and pixel-binning loss in Fig. S.4 (a) and (b).

Noise2Self (N2S) loss. We choose Noise2Self [1] among many self-supervised denoising methods [30, 29, 11, 8, 1, 3]. To calculate N2S loss, the L1 loss is computed between an output of an image inputted into KLAP, where the empty pixels of x_{J^c} are interpolated, and x_J. In our study, we utilize the same masking scheme for each J as outlined in the N2S paper [1]. Each J samples a single pixel selected within each 4×4 window (*i.e.*, 6.25% of the number of pixels in each image). In the original N2S method, the interpolation function for x_{J^c} use a 3×3 kernel to compute the average value of the surrounding pixels for interpolation. However, we consider the characteristics of the RGB channel and calculate the average value of the surrounding values corresponding to that channel for interpolation. In the case of Bayer, we set a size of window to 6×6 and use 5×5 kernel for interpolation to prevent overlap. The use of N2S loss term has the effect of removing independent noise.

Pixel-binning loss. As mentioned in Sec. 3.1 in our paper, pixel binning is applied differently depending on the input pattern status of the CFA. Similarly, the proposed pixel binning loss based on CIS domain knowledge is also applied differently according to the CFA pattern. When using the average-based pixel binning operation (m), the Q×Q CFA pattern is converted to Quad or Bayer pattern. Nona and Quad patterns are converted to Bayer pattern. Note that pixel binning operation (m) does not exist in the Bayer pattern. The upsampling operation (U) employs a bilinear



(a) Noise2Self (N2S) loss (b) Pixel-binning loss

Figure S.4: The specific processes for calculating 2 loss functions in our proposed method, KLAP-M, which is KLAP with meta-test learning: (a) Noise2Self (N2S) loss, and (b) Pixel-binning loss.

function to restore the original resolution, which may have been altered due to the pixel binning operation (m).

S.5. Implementation Details

In our experiments, we use a patch size of 240×240 to cover all of Bayer, Quad, Nona, and Q×Q CFAs. The model is trained using the ADAM optimizer with a batch size of 32 and an initial learning rate of 2×10^{-4}. We apply the cosine annealing learning rate decay technique with a minimum learning rate of 1×10^{-6}. To ensure a fair comparison, we evaluate the proposed method on the same conditions with an NVIDIA A6000 GPU using PyTorch [15]. Our architecture is similar to the NAFNet [4] architecture, which has state-of-the-art performance in IM-based image restoration. We use the official codes provided by the authors of Chen [5] and Li [12]. In the Table 1, TKL denotes the method of applying NAFNet-based TKL, and Chen denotes the method of applying MBSDN-based TKL.

S.6. Results

More results on CIS RAW. Fig. S.5 illustrates the qualitative results of the Baseline-UM (2nd column) with NAFNet, existing methods (Chen (3rd column), Li (4th column)) and our proposed KLAP (5th column), evaluated on the synthetic RAW (DF2K-CIS) test dataset. Also, Fig. S.6 presents the qualitative results of prior arts (Chen (2nd column), Li (3rd column)) and our proposed KLAP (4th column) and KLAP-M (5th column) on the synthetic RAW (DF2K-CIS) with 'strong noise' test dataset. Our proposed KLAP method visually outperforms other state-of-the-art methods on DK2K-CIS test dataset, and our proposed KLAP-M method shows visually superior results compared to other state-of-the-art methods on the DK2K-CIS test with strong noise dataset, thanks to meta-learning during inference.

Fig. S.7 shows our proposed KLAP-M inference output on the real CIS RAW data. Before applying meta-learning (in KLAP), artifacts exist; however, after applying meta-learning (in KLAP-M), the artifacts are effectively relieved. We selected 45 iterations for real RAW. Note that we observed that similar results were obtained even with further increases in iterations. Fig. S.8 presents additional KLAP-M inference output samples and their color-related mapped images (*i.e.*, CM-applied version) on real CIS RAW data.

Additional evaluation on 'another' RAW dataset. To further evaluate the robustness of our methods, we conduct additional inference on another open-source RAW dataset from the MIPI 2022 remosaic challenge [27]. This competition emphasizes Quad-to-Bayer *re-mosaicing*, not demosaicing, so the definition of ground truth (GT) differs from our research focus. Nevertheless, we conducted inference on the MIPI inputs using KLAP and KLAP-M, as shown in Fig. S.9, effectively reducing visual artifacts and validating their performance. This emphasizes the robustness and stability of our self-supervised learning approach, showcasing the effectiveness of our meta-test framework for various real sensor RAW data in real-world scenarios.

References

[1] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *ICML*, 2019.

[2] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019.

[3] Jaeseok Byun, Sungmin Cha, and Taesup Moon. Fbi-denoiser: Fast blind image denoiser for poisson-gaussian noise. In *CVPR*, 2021.

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *ECCV*, 2022.

[5] Wei-Ting Chen, Zhi-Kai Huang, Cheng-Che Tsai, Hao-Hsiang Yang, Jian-Jiun Ding, and Sy-Yen Kuo. Learning

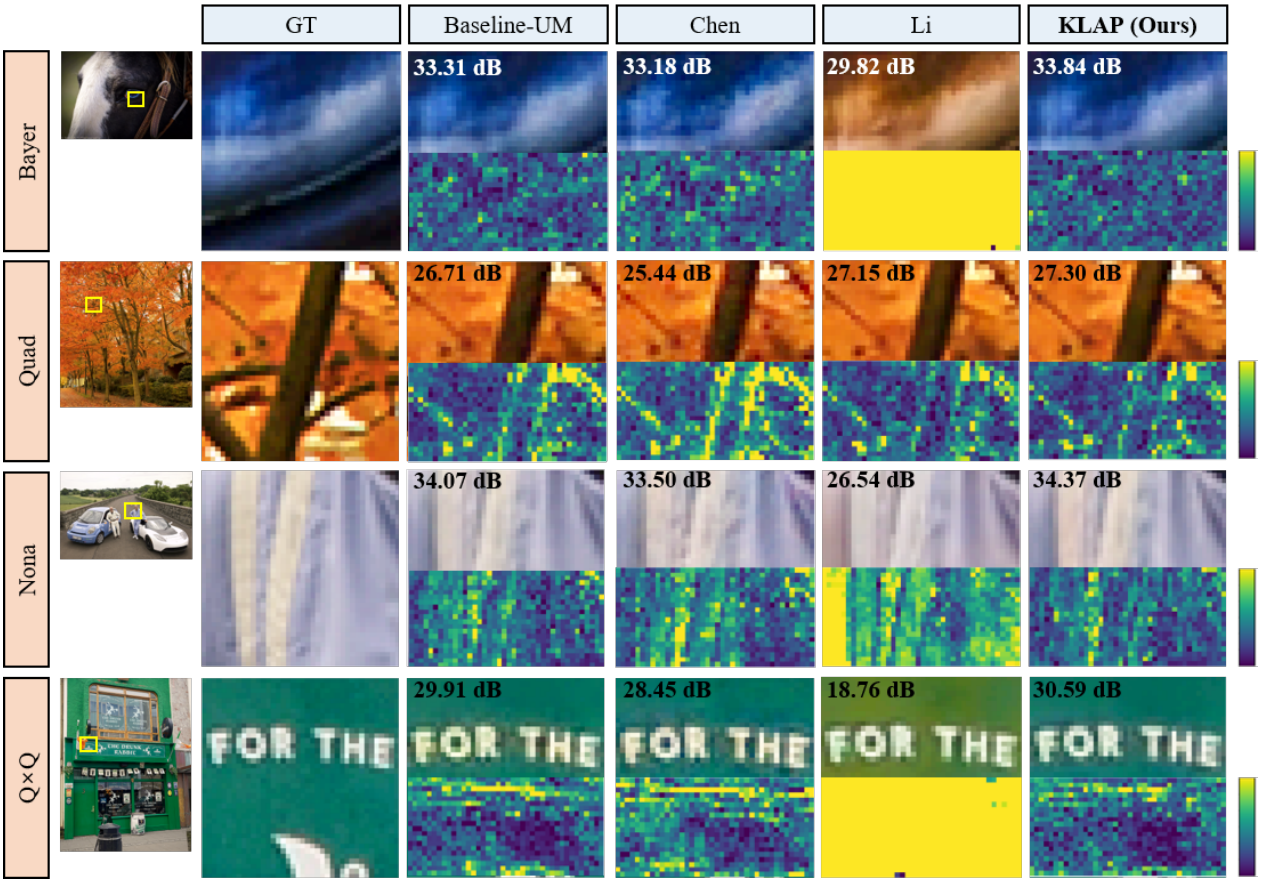


Figure S.5: Comparisons of demosaiced images **(top)** from different methods and their difference maps **(bottom)** on the synthetic RAW (DF2K-CIS) test dataset. The PSNR (dB) values displayed in the top-left corner of each image are calculated using the entire image.

multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: Toward a unified model. In *CVPR*, 2022.

[6] Egor Ershov, Alex Savchik, Denis Shepelev, Nikola Banić, Michael S Brown, Radu Timofte, Karlo Koščević, Michael Freeman, Vasily Tesalin, Dmitry Bocharov, et al. Ntire 2022 challenge on night photography rendering. In *CVPR*, 2022.

[7] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *NeurIPs*, 2018.

[8] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021.

[9] Mehdi Khabir, Hamzeh Alaibakhsh, and Mohammad Azim Karami. Electrical crosstalk analysis in a pinned photodiode cmos image sensor array. *Applied Optics*, 2021.

[10] Mehdi Khabir and Mohammad Azim Karami. Characterization and analysis of electrical crosstalk in a linear array of cmos image sensors. *Applied Optics*, 2022.

[11] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *ICML*, 2018.

[12] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *CVPR*, 2022.

[13] Yihao Li, Junyu Long, Yun Chen, Yan Huang, and Ni Zhao. Crosstalk-free, high-resolution pressure sensor arrays enabled by high-throughput laser manufacturing. *Advanced Materials*, 2022.

[14] Hao Ni, Jingkuan Song, Xiaopeng Luo, Feng Zheng, Wen Li, and Heng Tao Shen. Meta distribution alignment for generalizable person re-identification. In *CVPR*, 2022.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[16] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *CVPR*, 2022.

[17] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.

[18] Vitchyr H Pong, Ashvin V Nair, Laura M Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement

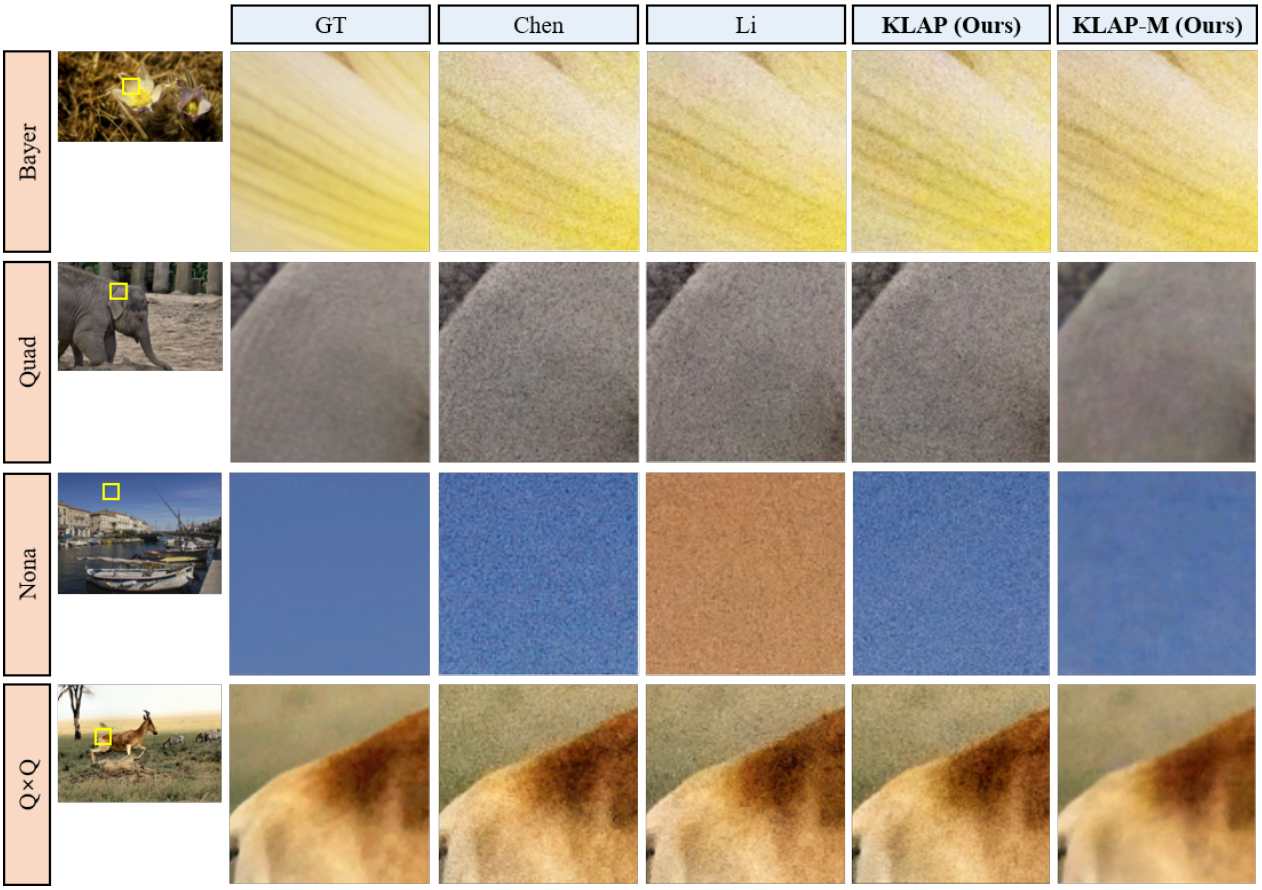


Figure S.6: Comparisons among different methods of robustness on DF2K-CIS with strong noise test dataset. The noise parameters used in the test are four times larger than the noise parameters used in the training. The number of meta-learning iterations in KLAP-M is set to 10, based on empirical determination through visualization of outputs in our experiments.

learning with online self-supervision. In *ICML*, 2022.

[19] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. *ECCV*, 2022.

[20] Corban G Rivera, David A Handelman, Christopher R Ratto, David Patrone, and Bart L Paulhamus. Visual goal-directed meta-imitation learning. In *CVPR*, 2022.

[21] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639*, 2016.

[22] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[23] Liangbin Xie, Xintao Wang, Chao Dong, Zhongang Qi, and Ying Shan. Finding discriminative filters for specific degradations in blind super-resolution. *NeurIPs*, 2021.

[24] Ying Xiong, Kate Saenko, Trevor Darrell, and Todd Zickler. From pixels to physics: Probabilistic color de-rendering. In *CVPR*, 2012.

[25] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE TIP*, 2020.

[26] Fengxiang Yang, Zhun Zhong, Zhiming Luo, Yuanzheng Cai, Yaojin Lin, Shaozi Li, and Nicu Sebe. Joint noise-tolerant learning and meta camera shift adaptation for unsupervised person re-identification. In *CVPR*, 2021.

[27] Qingyu Yang, Guang Yang, Jun Jiang, Chongyi Li, Ruicheng Feng, Shangchen Zhou, Wenxiu Sun, Qingpeng Zhu, Chen Change Loy, Jinwei Gu, et al. Mipi 2022 challenge on quad-bayer re-mosaic: Dataset and report. In *ECCVW*, 2022.

[28] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *CVPR*, 2021.

[29] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Extending stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. *NeurIPs*, 2019.

[30] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior. In *CVPR*, 2019.

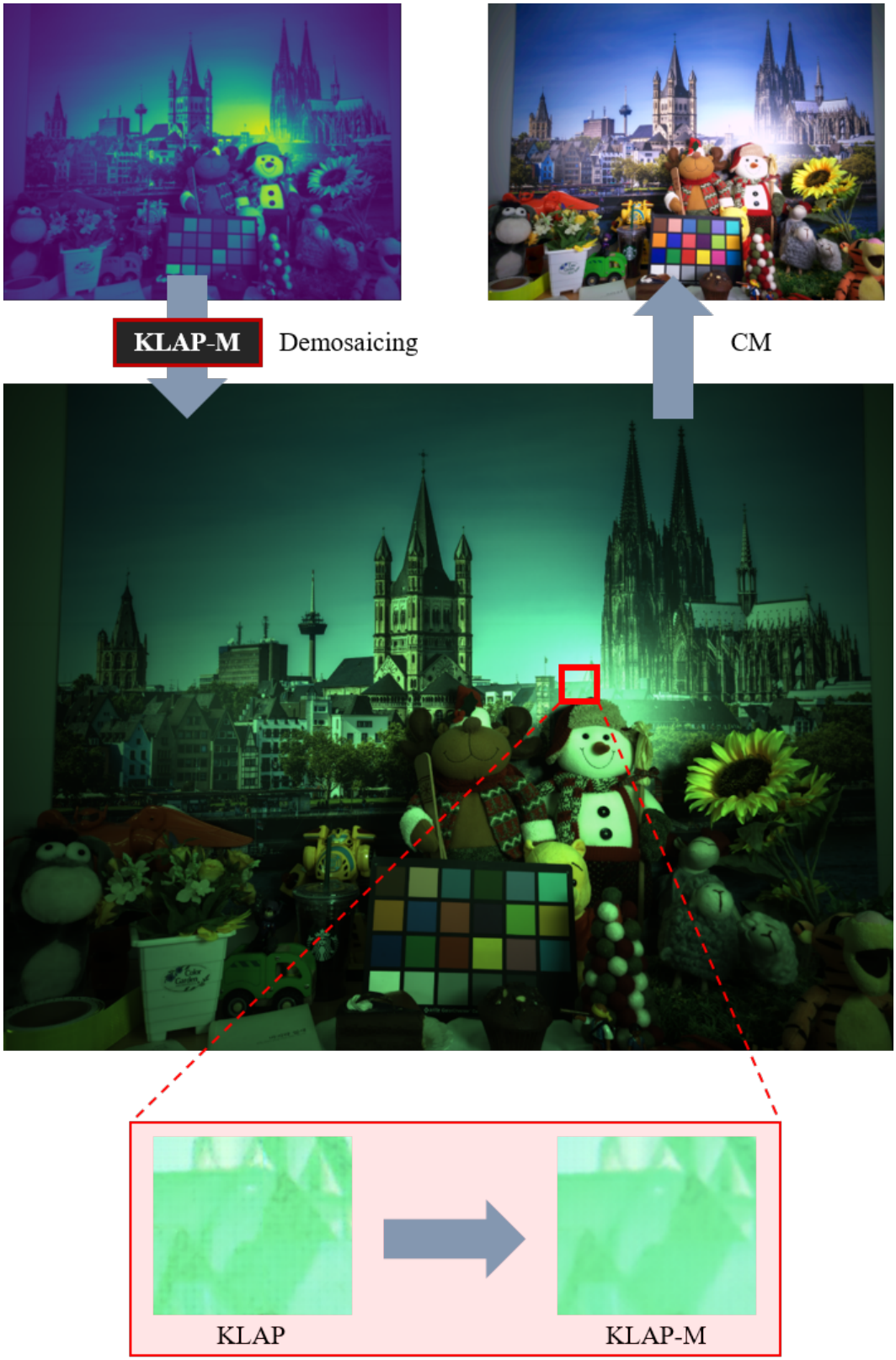
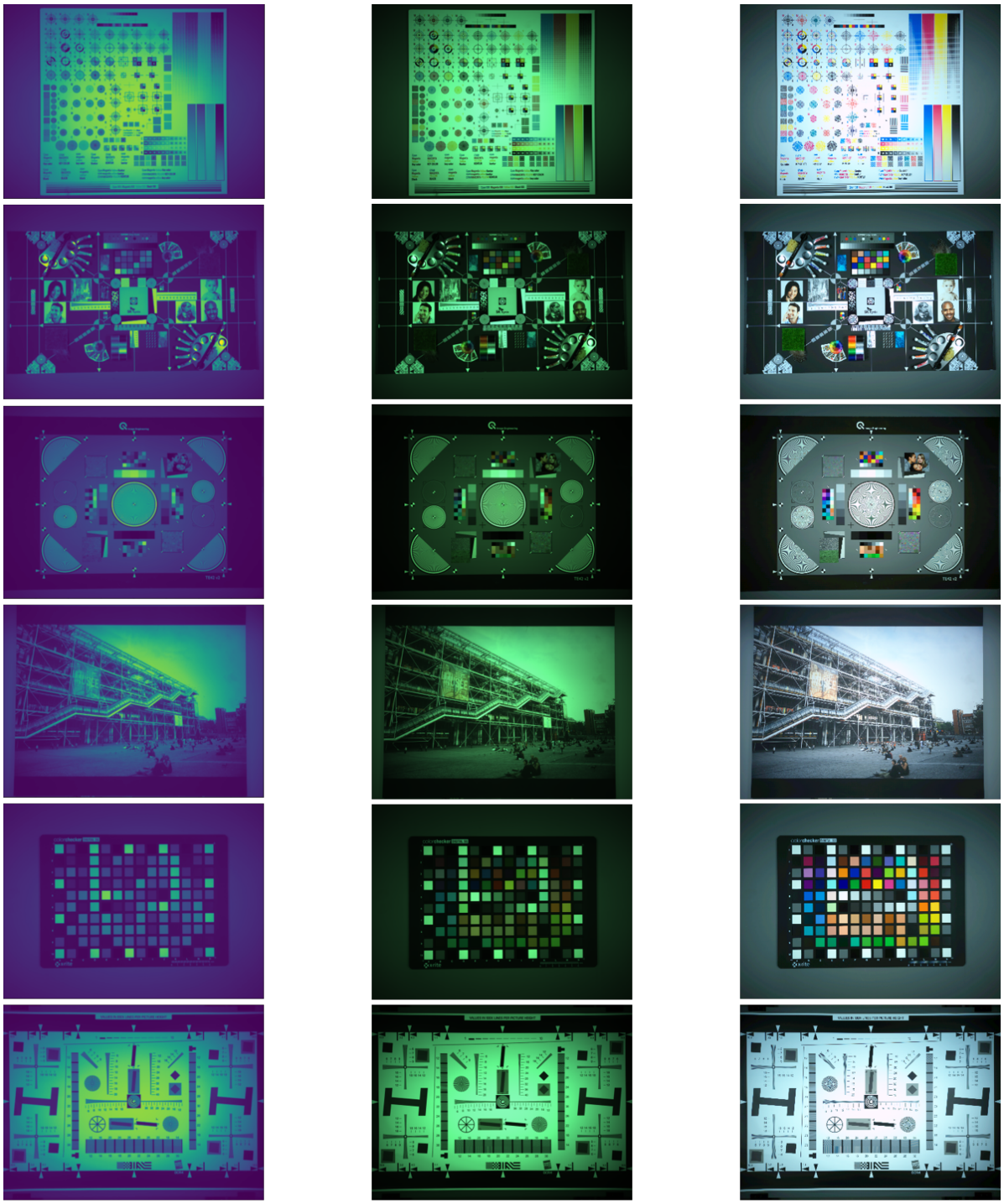


Figure S.7: The sample of demosaiced output images of Q×Q CIS RAW (48MP) with KLAP and KLAP-M.



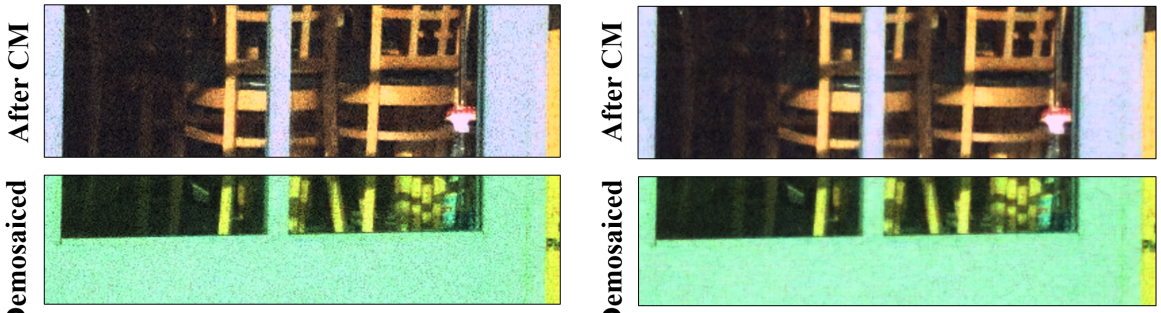
(a) CIS RAW (Q×Q)

(b) Demosaiced images (KLAP-M)

(c) Color-related Mapped images

Figure S.8: Additional CIS RAW and its inference outputs. (a) CIS Q×Q RAW data, (b) demosaiced output images obtained using KLAP-M inference, and (c) the same images as in (b) after applying CM (Color-related Mapping function). Note that in (c), it can be perceptually observed that CM works well not only on synthetic RAW images but also on real CIS RAW images.

(a) DM results of MIPI (b) DM results after CM



(c) KLAP (d) KLAP-M

Figure S.9: Additional inference images of CIS RAW, MIPI 2022 Quad remosaic challenge data [27] using KLAP (trained by our synthetic dataset, DF2K-CIS) and KLAP-M. (a) demosaiced output images obtained using KLAP-M inference, and (b) the same images as in (b) after applying CM (Color-related Mapping function). As shown in the figures above, KLAP with meta-test learning (KLAP-M) shows robust performance in another real CIS RAW, MIPI 2022 Quad data, despite of existence of sensor-generic unknown artifacts.