RAW-Domain Degradation Models for Realistic Smartphone Super-Resolution

Supplementary Material

This supplementary material presents additional details regarding the proposed degradation modeling for RAW SR and the experiments presented in the main paper.

S1. RAW Capturing Setup

Fig. S1 shows a snapshot of our capturing setup, which consists of a smartphone mounted on a tripod that captures an image displayed on the display prototype. This system is used to collect images for kernel modeling (Sec. 3.1 of the main paper) and paired LR-HR images for reference-based evaluation metrics (Sec. 4.2).

Camera-display framing. To effectively model subsampling performed in the camera imaging pipeline during the kernel modeling and evaluation data collection, the target display and camera are framed such that the spatial resolution of the displayed image exceeds the target resolution of the sensor. To this end, we display a HR image on the monitor with 1:1 scaling to prevent interpolation or stretching in the display domain. The distance d between the display and camera is adjusted based on the target SR scale.

Specifically, the scaling factor s is defined as the ratio between the displayed image resolution M and the captured image resolution N, where M corresponds to the spatial resolution of the HR image displayed on the monitor, and N corresponds to the resolution captured by the camera sensor. The distance d is then determined such that $s = \frac{f}{d}$, where f is the focal length of the camera. By adjusting the distance d, the camera captures the displayed image at the target resolution, simulating the effect of subsampling and ensuring the captured data is consistent with the intended target scale for kernel modeling and evaluation. This approach helps avoid unwanted interpolation and aliasing, providing more accurate data that reflects real-world imaging scenarios.

Camera specifications. We provide a list of the mobile cameras used in our experiments (Sec. 4). As shown in Tab. S1, these cameras have varying optical properties and sensor sizes, resulting in diverse degradation kernels and noise profiles. Cameras that are common across multiple devices are excluded. For example, the Main and Tele 1 cameras of Samsung S24U are identical to S23U Main and Tele 1, making separate degradation modeling unnecessary. Additionally, ultra-wide cameras are excluded from the degradation modeling, as they are not commonly used for digital zoom applications.

[1]Specifications obtained from: https://www.phonearena.com/phones

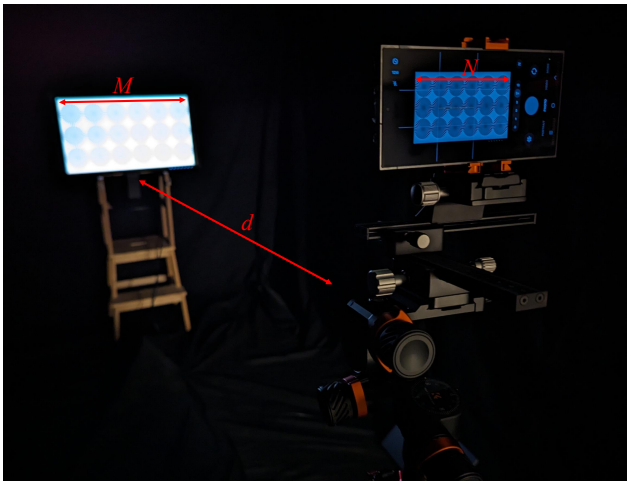

Figure S1. Our calibration setup and display prototype.

Camera	Optical Zoom	Focal Length	Sensor Size	Pixel Size	Sensor Resolution
S23+ Main	1x	23 mm	1/1.56”	1.0 μm	4080x3060 (50MP)
S23+ Tele	3x	69 mm	1/3.94”	1.0 μm	3648x2736
S23U Main	1x	24 mm	1/1.3”	0.6 μm	4080x3060 (200MP)
S23U Tele 1	3x	69 mm	1/3.52”	1.12 μm	3648x2736
S24U Tele 2	5x	111 mm	1/2.52”	0.7 μm	4080x3060 (50MP)
Pixel 6 Main	1x	26 mm	1/1.31”	1.2 μm	4080x3072 (50MP)
Pixel 9 Pro Main	1x	25 mm	1/1.31”	1.2 μm	4080x3072 (50MP)
Pixel 9 Pro Tele	5x	113 mm	1/2.55”	0.7μm	4032x3024 (48MP)
Mi 11	1x	25 mm	1/1.52”	0.7μm	6016x4512 (108MP)

Table S1. Specifications of all the cameras used in our experiments. Note that some of the listed sensors are designed with pixel-binning capability, but the capture app cannot access the unbinned version of the RAW captures. The Sensor Resolution column indicates the captured RAW resolution in binned (regular Bayer) format, while the number in parentheses indicates the actual sensor resolution before pixel-binning.[1]

RAW capture application. The native camera applications on mobile phones do not necessarily provide direct access to mosaicked RAW captures. For obtaining RAW data, a dedicated RAW capturing tool is required that meets the following criteria: (1) RAW access of the cameras listed in Sec. 4 and Tab. S1, (2) Manual ISO, shutter speed, and focus controls, and (3) Burst imaging capability in mosaicked RAW format. While burst capturing can be simulated by sequentially triggering the shutter to capture a scene, this may cause misalignment between the captures due to random spatial shifts caused by the optical image stabilizer (OIS), even in vibration-free remote capturing setups. For RAW data capturing, including noise calibration, kernel modeling, and evaluation data collection, we use the

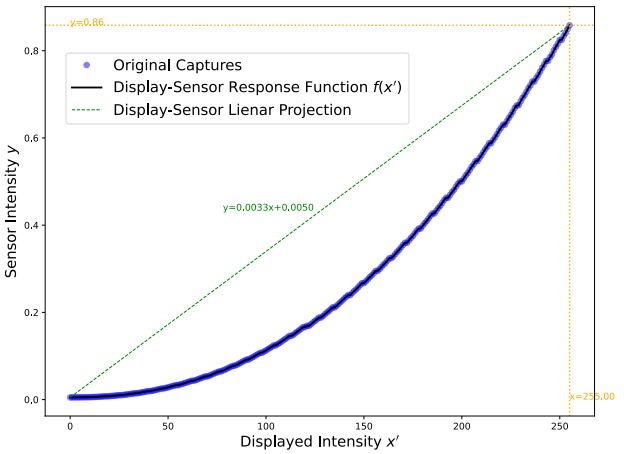

Figure S2. Display-to-sensor linearization function obtained for our display prototype and Pixel 9 Pro Main camera.

Android MotionCam Pro[2], which meets the above requirements. The capture process is automated through Android Debug Bridge (ADB) commands, enabling remote control of cameras.

S2. Radiometric Alignment between HR-LR

We employ a display prototype to generate HR-LR image pairs in both SR kernel modeling (Sec. 3.1) and SR model evaluation using reference-based metrics (Sec. 4.2). The displayed image undergoes a nonlinear transformation, such as tone adjustment and/or gamma correction, depending on the color profile and the monitor's specifications, before being captured by the target camera sensor.

To determine the nonlinear mapping from the displayed image to the sensor space, we first display gray patches at 255 different steps (gray-scale 8-bit format images, from 0 to 255), *i.e.* $\max_{\acute{\mathbf{x}}} = 255$ and capture a RAW image for each displayed patch. The intensity of the captured gray patches and their corresponding displayed values are used to fit a nonlinear curve to the measurements, denoted as $f(\acute{\mathbf{x}})$. The display-to-sensor response curve obtained through this process for the Pixel 9 Pro Main camera is shown in Fig. S2. We then display 140 color patches corresponding to the color values found in ColorChecker Digital SG [3]. Using the values obtained for the color patches in the senor domain and their linearized displayed sRGB values *i.e.* $f^{-1}(\acute{\mathbf{x}})$, we compute a 3×3 color correction matrix (CCM). This CCM, denoted by $\mathbf{C}$, together with the inverse of the monitor-to-sensor response curve, forms our color mapping function from the display domain to the sensor color space as:

$$\mathcal{D}^{-1}(\acute{\mathbf{x}}) = \mathbf{C} f^{-1}(\acute{\mathbf{x}}). \tag{S1}$$

[2] https://www.motioncamapp.com

[3] https://www.xrite.com/categories/calibration-profiling/colorchecker-digital-sg

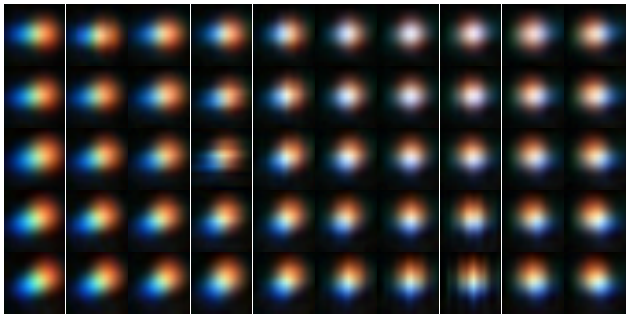

Figure S3. Examples of camera-specific SR kernels we calibrate and add to the pool of SR degradations; Calibrated $4\times$ SR kernels for the center of FOV of S23+ Tele camera. Each kernel's bounding box is $17\times$ 17.

S3. Kernel Modeling More Details

In Fig. S3, we show examples of kernels for $4\times$ SR that we calibrate for S23+ Tele camera. Below, we describe additional details regarding the kernel calibration process.

Sensor-to-display alignment. The camera and target display are geometrically aligned to approximate a perspective homography, which is used to initialize $\mathbf{H}$ in Eq. (3). For geometric alignment, we use gray-code structured patterns (Fig. 2 in the main paper) to establish the mapping between pixels in the display and their corresponding pixels in the sensor [36]. We employ a sequence of 23 vertical gray-code stripes and 23 horizontal gray-code stripes to encode the pixel locations on the screen and identify the corresponding positions on the sensor. These dense pairs of corresponding locations are then used to fit a homography matrix $\mathbf{H}$, which facilitates mapping of any displayed pattern on the monitor to the sensor domain as $\mathcal{W}(\acute{\mathbf{x}}, \mathbf{H})$ in Eq. (2).

We follow the steps outlined in Sec. S2 to calibrate the display's nonlinear color space and the color transformation from the display domain to the sensor. Note that when the displayed patterns consist only of black-and-white pixel values, the linearization and color mapping can be performed more efficiently.

Let $\mathbf{y}_1$ and $\mathbf{y}_2$ denote RAW captures of a pair of white ($\max_{\acute{\mathbf{x}}}$) and black (0) patches $\mathbf{x}_1$ and $\mathbf{x}_2$ displayed on the monitor, respectively (Fig. 2)). Then, the linearization and color mapping can be simplified to:

$$\mathcal{D}^{-1}(\mathbf{x}) = \frac{\left(\frac{\mathbf{x}(\max_{\mathbf{y}} - \min_{\mathbf{y}})}{\max_{\acute{\mathbf{x}}}} + \min_{\mathbf{y}}\right) - b}{w - b}, \tag{S2}$$

where $\max_{\mathbf{y}} = \text{Mean}(\mathbf{y}_1)$ and $\min_{\mathbf{y}} = \text{Mean}(\mathbf{y}_2)$ are mean intensities of RAW captures of a pair of white and black patches $\mathbf{x}_1$ and $\mathbf{x}_2$ displayed on the monitor, and w and b denote the white level and the black level of the sensor, respectively. Here, Mean$(\cdot)$ denotes spatial averaging.

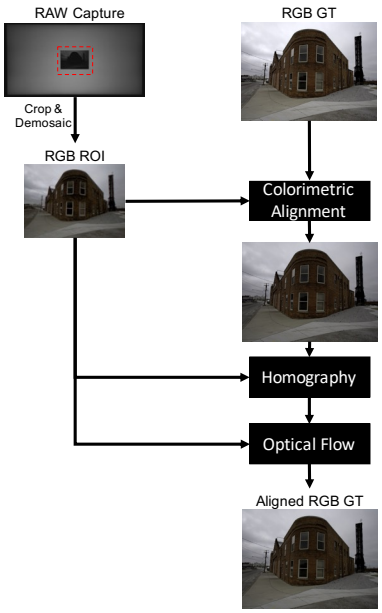

Figure S4. Our alignment method for producing paired evaluation data. After aligning the color and brightness of the ground-truth image with a demosaicked crop of the monitor capture, we perform spatial alignment using homography and optical flow. The resulting aligned ground-truth image is well-suited for computing reference-based quantitative metrics, with the RAW capture crop serving as the input.

Using the white and black field captures $\mathbf{y}_1$ and $\mathbf{y}_2$, we can model the lens vignette effect combined with the non-uniformity of the brightness on the monitor. It is straightforward to show that the combination of $\mathbf{v}$ and $\mathcal{D}^{-1}(\cdot)$ can be modeled as:

$$\mathbf{v} \odot \mathcal{D}^{-1}(\mathbf{x}) = \frac{\left(\frac{\mathbf{x} \odot (\mathbf{y}_1 - \mathbf{y}_2)}{\max_{\hat{\mathbf{x}}}} + \mathbf{y}_2\right) - b}{w - b}, \qquad \text{(S3)}$$

where $\odot$ denotes element-wise multiplication. As seen here, using only two pixel intensities–0 and $\max_{\hat{\mathbf{x}}}$–to design kernel modeling patterns makes the model less sensitive to the non-linear color space of the display.

Capture details. Our monitor-to-sensor image formation model in Eq. (2) assumes that the observations are noise-free. Thus, we capture a burst of 100 images of the displayed pattern at the lowest ISO level available in the RAW capture application and average them to obtain $\mathbf{y}$, a noise-free RAW image for Eq. (3).

The exposure time is adjusted so that the white pixels displayed on the monitor map to approximately 80% of sensor' white-level. This ensures good contrast in the captures while avoiding saturation of the pixels.

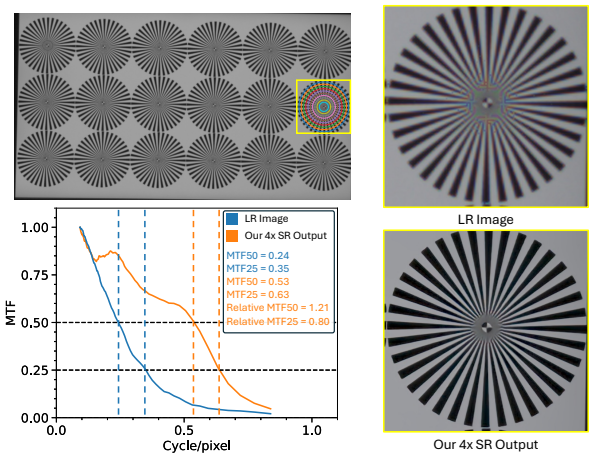

Figure S5. Our MTF evaluation. The MTF measured on the LR image is used to calculate the relative MTF on the output of SR models. Here is the actual measurement for our model output in the Pixel 9 Pro Main experiment (Tab. 1 of the main paper). The average of all values across all the Siemens star patterns in the FOV is reported per model in our experiments in Sec. 4.

S4. Paired Evaluation Data Collection

To use a pair of ground-truth and display-capture images for reference-based evaluation of our models, the ground-truth image must first be spatially aligned with the display-capture image. An overview of our alignment pipeline is depicted in Fig. S4. The alignment process begins by linearizing the ground-truth image from the display domain to the sensor domain, as described in Sec. S2. Since the display capture is in mosaicked RAW format, we first apply bilinear demosaicking for alignment. Then, we apply the white balance gains obtained from the metadata of the RAW display capture to both the demosaicked display image and the ground-truth linearized image.

After cropping the ROI from the demosaicked display capture containing the ground-truth image, we perform bicubic downsampling on the ground-truth image such that it becomes exactly $4\times$ larger than the ROI. To correct for the residual brightness difference between the two images, we match the histogram of the ground-truth image and the ROI. The calibrated homography matrix for the sensor-display is then applied to spatially align the ground-truth image with the RAW ROI at a global scale. Finally, to improve local alignment, we use a pre-trained optical flow estimation network [38] to further warp the ground-truth image to better match the ROI.

S5. Details Regarding MTF Metric

In our experiments, we use MTF50 and MTF25 to quantify the details recovered in the model outputs. Fig. S5 shows the pattern we use to measure MTF. The RAW capture of

Camera	Method	Ref. Metrics	
		PSNR	SSIM
Pixel 6 Main	BSRAW [15]	28.75	0.877
	Ours (Cross-camera)	**29.14**	**0.913**
Mi11 Main	BSRAW [15]	33.17	0.897
	Ours (Cross-camera)	**39.72**	**0.956**

Table S2. Our 4× SR model is trained on synthetically generated data using our degradation models obtained from seven different cameras. The performance is evaluated on RAW images captured by the Pixel 6 Main and Mi 11 Main cameras, and compared with other baselines. The degradations of these two cameras are not included in the training data for any of the models.

the displayed pattern is processed using the SR model, and MTF50 and MTF25 are obtained from the MTF plot for each Siemens star in the image. The MTF plot represents contrast as a function of spatial frequency, derived from contrast modulations along a sinusoidal wave fitted to intensities measured at pixels located at different radii from the pattern center. Determining the MTF curve this way can be sensitive to the distance between the camera and the display. Therefore, in each camera-specific evaluation, we measure the MTF on the LR image and use it to calculate the relative MTF as a metric for the enhancement from the LR image to the super-resolved image. Fig. S5 shows MTF curves for both LR and HR images, along with the obtained relative MT50 and relative MT25 for one of the Siemens star patterns in the FOV.

S6. Additional Results

We present additional qualitative results in Figs. S6 and S7 for Pixel 6 Main and Mi 11 cameras, respectively, obtained in our cross-camera SR experiments (Sec. 4.5 of the main paper).

Since BSRAW [15] is one of our comparison baselines, which performs RAW-to-RAW super-resolution, we convert all RGB outputs from our model into RAW format by simulating Bayer mosaicking. We then compute the PSNR and SSIM on these mosaicked outputs and compare with metrics computed on the RAW outputs from BSRAW. This approach eliminates any potential detrimental effects a downstream demosaicking method might introduce into the outputs of the BSRAW baseline. The metrics for all outputs are calculated on the four-channel stack of Bayer channels and reported in Tab. S2.

Figure S6. Additional qualitative results for 4× SR on **Pixel 6 Main** from the experiments in Sec. 4.5 of the main paper. Note that Pixel 6 Main degradations are not explicitly included in the training data of any of the models. Model outputs are white-balanced and gamma-corrected for better visualization.

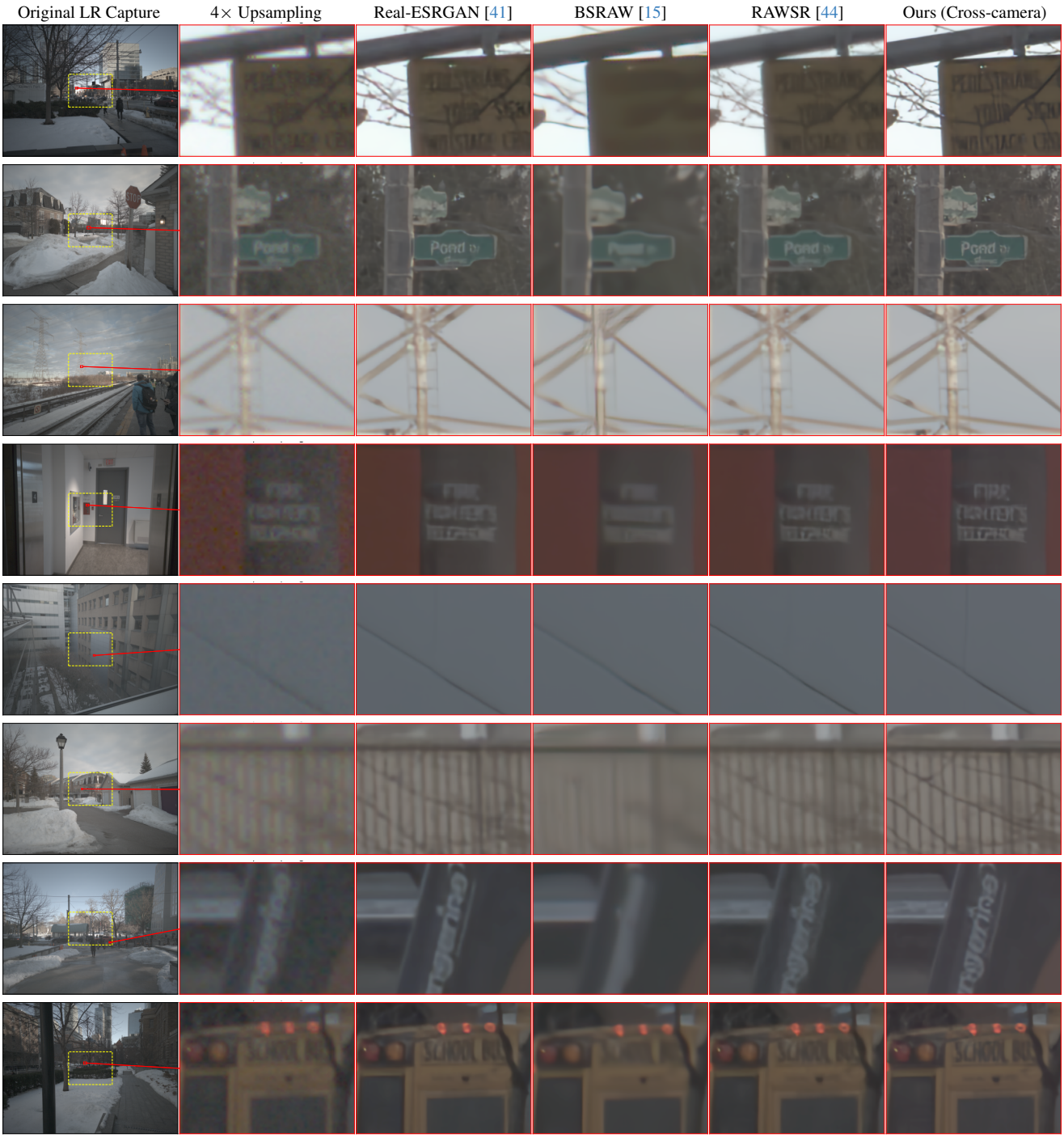

Figure S7. Additional qualitative results for **Mi 11 Main** 4× SR results regarding the experiments presented in Sec. 4.5 of the paper. Note that Mi 11 Main degradations are not explicitly seen by any of the models during training. Model outputs are white-balanced and gamma-corrected similarly for better visualization.