Transform your Smartphone into a DSLR Camera: Learning the ISP in the Wild

Ardhendu Shekhar Tripathi[1], Martin Danelljan[1], Samarth Shukla[1], Radu Timofte[1,2], and Luc Van Gool[1,3]

ETH Zurich, Switzerland[1], University of Wurzburg, Germany[2], KU Leuven, Belgium[3]

Abstract. We propose a trainable Image Signal Processing (ISP) framework that produces DSLR quality images given RAW images captured by a smartphone. To address the color misalignments between training image pairs, we employ a color-conditional ISP network and optimize a novel parametric color mapping between each input RAW and reference DSLR image. During inference, we predict the target color image by designing a color prediction network with efficient Global Context Transformer modules. The latter effectively leverage global information to learn consistent color and tone mappings. We further propose a robust masked aligned loss to identify and discard regions with inaccurate motion estimation during training. Lastly, we introduce the ISP in the Wild (ISPW) dataset, consisting of weakly paired phone RAW and DSLR sRGB images. We extensively evaluate our method, setting a new state-of-the-art on two datasets. The code is available at https://github.com/4rdhendu/TransformPhone2DSLR.

1 Introduction

An Image Signal Processing (ISP) pipeline is characterized by a sequence of low-level vision operations that are performed to convert RAW data from the camera sensor to sRGB images. Each camera has an inherent ISP that is implemented on the device through hand-designed operations. With the advent of mobile photography, smartphones have become the primary source of photo capture due to their portability. However, their strict size constraints enforces small sensor sizes and compact lenses, which inevitably leads to higher sensor noise compared to DSLR cameras. In this work, we therefore strive towards mitigating the hardware constraints in mobile photography by designing a learnable alternative to the ISP pipeline, utilizing DSLR quality sRGB images as reference.

Compared to standard image enhancement/restoration tasks, learning the ISP mapping introduces new fundamental challenges, which require careful attention. In the paired learning setting, a primary issue is that the color mapping between the input RAW image and the DSLR sRGB image depends on partially unobserved factors, such as camera parameters and the environmental conditions. Further, the image pairs for training, each consisting of a smartphone RAW and a DSLR sRGB, inevitably contain substantial spatial misalignment that greatly complicate the learning. Despite recent efforts [8,4,24], these issues remain central in the strive towards a fully learning-based ISP solution.

(a) Phone RAW (b) LiteISP-Net [24]. (c) Ours (d) DSLR sRGB

Fig. 1. Our learnable ISP generates a DSLR quality sRGB image from RAW data captured by a smartphone camera. Our approach recovers rich details and produces colors that are more consistent with the DSLR sRGB ground-truth, compared to LiteISPNet (best performing competing method). Shown are the full resolution results on our ISP in the Wild (ISPW) dataset. Best viewed with zoom.

In this work, we propose a learnable ISP framework that can be effectively trained *in the wild*, using only weakly paired DSLR reference images with unknown and varying color and spatial misalignments. Our approach is composed of an ISP network that maps the input phone RAW to a DSLR quality output. Contrary to much previous works, we further condition the network on a target color image. This allows our ISP network to fully focus on the denoising and demosaicing tasks, without having to guess the unknown color transformation. To allow the target color image to be used during training, we propose a flexible and efficient parametric color mapping. Our color mapping between the input RAW and output DSLR sRGB image is individually optimized for every training image pair. The resulting mapping is then applied to the input RAW image to generate the target color image for conditioning. Importantly, this approach effectively mitigates information leakage from the target ground truth into the network, while achieving a faithful color transformation.

In order to achieve the target color image during inference, we further propose a dedicated target DSLR color prediction network, which solely takes the RAW phone image as input. To predict an accurate target color image, exploiting both local and global cues in an image is essential. While local information capture high-frequency details, global information is important in order to achieve a globally consistent and realistic color mapping across the entire image. We achieve the latter by designing an efficient Global Context Transformer block, which aggregates global color information into a compact latent array through cross-attention operations. This both alleviates the quadratic complexity of standard transformer modules, and importantly enables a variable input size. Finally, we address the problem of misaligned ground-truth by introducing a robust masked aligned objective for training our ISP framework.

To aid in extensive benchmarking and evaluation of RAW-to-sRGB mapping approaches for weakly paired data, we introduce the ISP in the Wild (ISPW) dataset. This dataset comprises of pairs of RAW sensor data from a recent

smartphone camera and sRGB images taken from a high-end DSLR camera. Our dataset consists of 197 captured 10+ MegaPixel image pairs, resulting in over 35,000 crops of size 320×320 for training, validation, and test. We perform extensive ablative and state-of-the-art experiments on the Zurich RAW-to-RGB (ZRR) dataset [8] and our ISPW dataset. Our approach outperforms all previous approaches by a significant margin, setting a new state-of-the-art on both datasets. Example visual results are provided in Fig. 1.**Contributions:** Our main contributions are summarized as: **(i)** We propose a color conditional trainable ISP in the wild. **(ii)** We propose a color prediction network that integrates a global-context transformer module for efficient and globally coherent prediction of the target colors. **(iii)** We condition on color information from the reference image during training by introducing a flexible parametric color mapping, which is efficiently optimized for a single RAW-sRGB training pair. **(iv)** We employ a loss masking strategy for robust learning under alignment errors. **(v)** We introduce the ISPW dataset for learning the camera ISP in the wild.

2 Related Work

Despite the successes of deep-learning for low-level vision tasks, its application to camera ISP in the wild has been much less explored. Among the existing methods, CycleISP [22] and Invertible-ISP [21] propose a full camera imaging pipeline in the forward and reverse directions. These methods learn the ISP in a well aligned setting, where the RAW-sRGB training pairs originate from the same device. For RAW-to-sRGB mapping in the wild, the goal of the AIM 2020 challenge [8] on learned image processing pipeline was to map the original low-quality RAW images captured by a phone to a DSLR sRGB image. In particular, the CNN approaches inspired by the Multi-level Wavelet CNNs (MWCNN) [13] obtained the best results. Among the MWCNN-based methods both, MW-ISPNet [8] and AWNet [4] employ different variations of a U-Net for generation of appealing sRGB images.

More recently, LiteISPNet [24] propose an aligned loss by explicitly calculating the optical flow between the predicted DSLR image and the ground truth. The idea of the aligned loss using optical flow in case of misaligned data was first used in DeepBurstSR [2] for burst super-resolution. Prior to DeepBurstSR, other efforts to handle misaligned data include a contextual bilateral loss (CoBi) [23] or primarily relying on a deep perceptual loss function, as in MW-ISPNet [8] and AWNet [4].

Another bottleneck for the field has been the dearth of datasets for camera ISP learning and benchmarking. The datasets MIT5K [3], DND [16], SIDD [1] and Zoom-to-Learn [23] capture several images from the same device under different settings. Moreover, [3,16,1] collect images in very controlled settings, where accurate alignment is possible. They are therefore unfit for designing approaches for ISP in the wild. Further, DPED [7] provides RGB images from different devices but does not contain RAW images and thus cannot be used for our task of designing and training the full ISP pipeline. In contrast, we aim to learn the ISP from a constrained device, i.e. smartphone, using high-quality DSLR images.

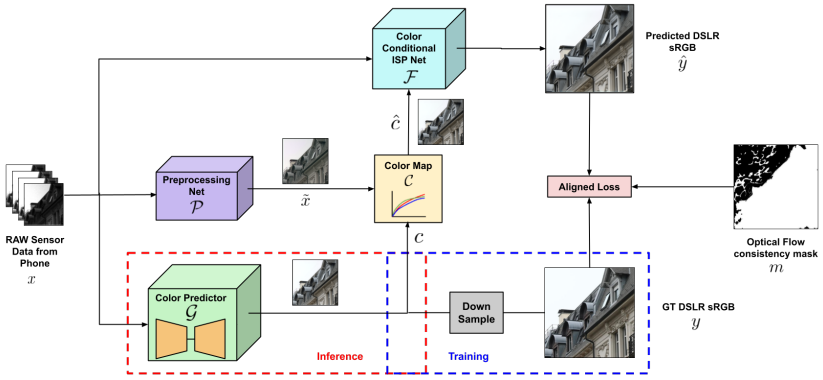


Fig. 2. An Overview of our learnable ISP framework: We learn a color conditional framework $\mathcal{F}(x, \hat{c})$ for RAW-to-sRGB mapping in the wild (Sec. 3.1). The estimated target color image $\hat{c}$ is achieved by our color mapping $\hat{c} = \mathcal{C}(x, c)$ (Sec. 3.3), which maps the raw input x to the color space of c. During training c is given by the downsampled ground truth. During inference, the DSLR-quality color content is predicted by the dedicated global attention based color prediction network $\mathcal{G}(x)$, using only the raw image x as input (Sec. 3.2). Finally, for robust learning of the ISP in the presence of even substantial misalignments (see Fig. 1), we propose a masked aligned loss (Sec. 3.4), which is robust to errors in the computed optical flow.

The BurstSR dataset [2] is designed for the burst super-resolution task. Most related is the ZRR dataset [8]. Our ISPW dataset contains RAW images collected via a more modern smartphone. Additionally, our ISPW dataset contains important meta information, such as the ISO and exposure settings, that can further be exploited by the community for controllable and conditional learning of the RAW-to-sRGB mapping for weakly paired data.

3 Method

In this work, we strive towards a fully deep learning based ISP module, which predicts a high-quality sRGB image $y \in \mathbb{R}^{3\times H\times W}$ given the RAW image $x \in \mathbb{R}^{4\times \frac{H}{2}\times \frac{W}{2}}$ captured by a mobile phone camera. Specifically, our aim is to learn such a module from a set of weakly paired training samples $\{(x^k, y^k)\}_k$. Our approach is illustrated in Fig. 2. It is comprised of a color conditional restoration network $\mathcal{F}(x, \hat{c})$ (Sec. 3.1). The color information $\hat{c}$ is provided by a dedicated color prediction network $\mathcal{G}(x)$ during inference (Sec. 3.2) and by the ground truth DSLR sRGB during training. To avoid the network from cheating during training, we propose a color mapping approach (Sec. 3.3) that maps the RAW sensor data to the target DSLR sRGB. During inference, our color mapping module works as a regularizer for our color predictor network in case of spurious inaccurate local colors predicted. Further, there also exists a spatial misalignment between the noisy mobile sensor data and the target DSLR sRGB image. To handle misalignment between the RAW-sRGB pairs, we propose a robust masked

aligned loss (Sec. 3.4) that also takes into account the inaccuracies that are introduced during the alignment operation.

3.1 ISP Network

As motivated in Sec. 1, there exists an unknown color mapping between the input x^k and the target y^k, which further varies between each capture (x^k, y^k) due to changes in the parameters and environment. Modelling the ISP pipeline in the wild as a single feed-forward network $y = \mathcal{F}(x)$ can therefore prove detrimental to the learning of an accurate RAW-to-sRGB mapping as no fixed global color mapping exists. In order to learn effectively the RAW-to-sRGB mapping in these conditions, we propose a network $y = \mathcal{F}(x, \hat{c})$ that is conditioned on the desired output color information $\hat{c}$. During training, the color information is extracted from the RAW-sRGB pair using a flexible parametric formulation, which is detailed in Sec. 3.3. This allows us to capture a rich color mapping model from a single training pair (x^k, y^k), while preventing the network $\mathcal{F}$ to cheat. Additionally, our dedicated RAW pre-processing network discussed in Sec. 3.3 mitigates the ill-effects that noise in the RAW sensor data has on our color mapping estimation module. During inference, the color information $\hat{c}$ is predicted by a dedicated color predictor network $\mathcal{G}(x)$ (Sec. 3.2) and the color mapping module (Sec. 3.3). Compared to a handcrafted ISP pipeline, demosaicing, denoising, and detail enhancement is performed by our ISP net, while color correction, gamma, and tone is handled by the color prediction network (Sec. 3.2).

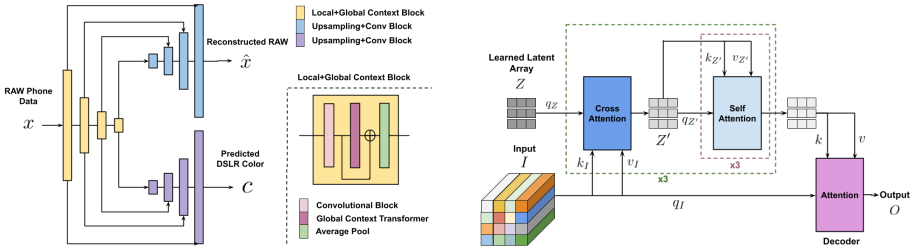
3.2 Color Prediction

In this section, we propose a low-resolution reference color prediction network $c = \mathcal{G}(x)$. This network aims to predict a low-resolution image c with the color content and dynamic range of the target DSLR camera. It is then the task of our ISP network $\mathcal{F}$, to predict a detailed high-resolution image, conditioned on this color information. The measured colors and intensities depend on the camera parameters during capture, along with various other environmental factors, such as the properties of the illuminants in the scene. These conditions vary on a capture to capture basis. Hence, a simple feedforward network fails to capture the DSLR sRGB color accurately.

Color prediction network: To circumvent this drawback of feed-forward nets, we design an encoder-decoder based color prediction network (Fig. 3a).

$$c = \mathcal{G}(x) = D_{\text{DSLR}}\big(E_{\text{phone}}(x)\big). \tag{1}$$

Here, D_{DSLR} is the DSLR decoding network that predicts a low resolution target sRGB color. Predicting the target sRGB colors in low resolution makes the learning easier and leads to a faster convergence. We employ a U-Net inspired architecture (Fig. 3a) for our encoder-decoder. This is because U-Net [17] effectively expands the receptive field by integrating pooling operations and exploiting contextual information at different scales using skip connections. Further, a U-Net is relatively insensitive to small misalignments in the image due to the



(a) Architecture overview. (b) Global Context Transformer.

Fig. 3. Illustration of the full Color Prediction Network (a) with its Global Context Transformer module (b).

low-resolution of core features, achieved by successive pooling operations. Our U-Net encoder E_{phone} exploits local and global cues by integrating a successive convolutional layer and an efficient global context transformer.

Global Context Transformer: For target color prediction, capturing a global context is pivotal since color in one patch of the image can be related to the color in a spatially distant patch of the same image. Hence, attending to different patches in the image may prove beneficial for predicting an accurate target color. Using standard transformers [19] for global attention is a viable option. However, its quadratic computational complexity w.r.t. the number of patches in the image/feature map makes it unsuitable for our color prediction network. Furthermore, our network needs to be able to process an image of arbitrary resolution, which brings further challenges to a standard transformer architecture.

We therefore design our Global Context Transformer block by taking inspiration from the Perceiver [10,9] architecture. Specifically, we perform cross attention operations between an auxiliary latent space $Z' \in \mathbb{R}^{K \times C}$ and the input feature map $I_l \in \mathbb{H}_l \times \mathbb{W}_l \times \mathbb{D}_l$, followed by self attention layers on Z'. Here, I_l is extracted from the U-Net encoder at level l. The latent space contains K tokens of dimension C, as is initialized by a learned constant array $Z \in \mathbb{R}^{K \times C}$. The majority of the computation thus happens on Z'. This reduces the complexity of the attention operations from quadratic to linear in the input size, and crucially enables a variable input image size.

Fig. 3b details the architecture of our global context block. It comprises multiple cross and self-attention layers on the fixed-size auxiliary latent array Z'. Cross-attention has a complexity of $O(NK) = O(N)$ (here $N = \mathbb{H}_l \times \mathbb{W}_l$) since $K \ll N$ is a small constant. Moreover, self-attention is only performed on Z, leading to a complexity of $O(K^2) = O(1)$. Hence, decoupling the network depth from the input size. Through the global attention operations, the learned latent arrays Z' can encode color transformations. The final decoder module then maps information encapsulated in Z' to the output array O through cross attention with the input query q_I. We integrate our Global Context Transformer block in the contracting path of our color prediction module after each convolutional block (Fig. 3a). This aids in exploiting local cues as well as global cues while remaining computationally efficient.

Reconstruction branch: In addition to the DSLR specific decoder, we also employ a decoder D_{phone} for reconstructing the RAW input x such that $\hat{x} = D_{\text{phone}}(E_{\text{phone}}(x))$ (Fig. 3a). Employing a RAW reconstruction decoder equips our color prediction framework to learn an optimal phone-specific embedding $E_{\text{phone}}(x)$ that encodes various meta-information that was not provided with the RAW data for reconstructing the RAW input x. Hence, intuitively our DSLR-specific decoder learns a mapping from the phone ISP to the DSLR ISP.

3.3 Color Mapping Module

In this section we introduce our approach for estimating the color transformation between the RAW input x and a target color sRGB image c. For this, we design a module $\hat{c} = \mathcal{C}(x, c)$ that estimates a color mapping between a single pair (x, c), and applies it to x. The result represents the RAW image x transformed according to the target color space in c. Our approach is particularly important during training, when c is derived from the ground-truth image y through downsampling and alignment. It supplies our ISP network, conditioned on $\hat{c}$, with the correct color transformation between the pair (x, y) while preventing information leakage from the ground-truth y. During inference, $\mathcal{C}$ works as a regularizer for our color predictor network (1) for spurious inaccurate local colors predicted.

Pre-processing network: Real world training image pairs, apart from being weakly paired in terms of alignment, pose many other challenges. In particular, the RAW sensor data from the phone is prone to noise due to the limited sensor size, along with other interference from the environment. The noise may be signal-dependent or signal-independent. A noisy source image x inhibits the performance of the color mapping significantly. Hence, removing noise from the RAW data is pivotal. In this direction, we design a pre-processing module for removing noise from the RAW data, thereby aiding our color mapping module.

Our RAW pre-processing network $\mathcal{P}$ aims to retrieve the clean source image $\tilde{x}$ given a noisy RAW x,

$$\mathcal{P}(x) \coloneqq \tilde{x} = x' - \eta(x'), \text{ where } x' = \Gamma(x). \tag{2}$$

Here, η is our noise estimation net and is implemented as a CNN with residual connections. For our framework, x' is a processed version of the mobile RAW sensor data x. We obtain x' by neglecting one of the green channels in x and normalizing the resulting 3-channel image between $[0, 1]$ uniformly. To further reduce the non-linearities in the color mapping, we apply a constant approximate gamma correction to obtain the final processed image x'. The processing operation $\Gamma(\cdot)$ is detailed in the supplementary.

Color mapping: Formulating our color mapping scheme, we define a set of $\mathcal{B}$ equally spaced bins between the range of values in each channel of the source image $\tilde{x}$ (Eq. 2). The b^{th} bin centroid for color channel j is denoted as k_b^j. The goal is to map the image $\tilde{x}$ to the target color image as,

$$\hat{c}_i^j = \sum_{b=1}^{\mathcal{B}} \hat{w}_{ib}^j (A_b^j \tilde{x}_i + B_b^j), \tag{3}$$

using a learned affine transformation $A_b^j \tilde{x}_i + B_b^j$ for each bin b. Here, $A_b^j \in \mathbb{R}^{1\times 3}$ and $B_b^j \in \mathbb{R}$ are the parameters of the affine map, while $\tilde{x}_i \in \mathbb{R}^3$ (Eq. 2) denotes the color values at pixel i after the pre-processing network. The result $\hat{c}_i^j$ is the mapped intensity at channel j and location i. The soft bin assignment weights in (3) are calculated as $\hat{w}_{ib}^j = \underset{b}{\text{SoftMax}}(-\|\tilde{x}_i^j - k_b^j\|^2/T)$, where, T is a temperature parameter. Hence, our color mapping (3) can be seen as an attention mechanism, with the source image attending to the learned values through the bin centroids. The motivation of learning an affine transformation instead of a fixed numeric value for each bin centroid is providing each bin more expressive power leading to better color mapping even with less number of bins.

In (3), the parameters (A_b^j, B_b^j) of the affine mapping are learned using only a single pair $(\tilde{x}, c)$. This is performed by minimizing the following squared error to the target color value c_i^j,

$$A_b^j, B_b^j = \underset{A,B}{\text{argmin}} \sum_i w_{ib}^j \|A\tilde{x}_i + B - c_i^j\|_2^2 \,. \tag{4}$$

Here, the weights $w_{ib}^j = \underset{i}{\text{SoftMax}}(-\|\tilde{x}_i^j - k_b^j\|^2/T)$. These set of weights signify how much each target intensity affects the affine transformation learned for each bin centroid. The objective (4) corresponds to a linear least squares problem, which can efficiently be solved in closed form as detailed in the supplementary.

3.4 Learning the Camera ISP

The RAW-sRGB pairs taken from two different devices are misaligned. The reasons are the different fields of view for both the cameras, parallax, and small motion of objects in the scene. Misalignment in the RAW-sRGB pair makes training the ISP pipeline difficult. Trying to learn in such a setting produces blurry results and significant color shift (Fig. 4). Hence, a robust loss applicable to the weakly paired setting is pivotal. In this section, we introduce an aligned masked loss for robust learning in a weakly paired setting. We then introduce the objectives for our main ISP network, pre-processing network, and the color prediction network. Lastly, we provide training strategies and details.

Alignment: We calculate aligned losses for learning our color conditional RAW-to-sRGB network in the wild. For alignment, we use the PWC-net [18] for computing optical flow. We denote by $c_{x'} = \mathcal{W}(c, f(c, x'))$ the color image c aligned with respect to the processed RAW x' (Sec. 3.3). Here, $f(c, x')$ is the optical flow from the color image c to the processed RAW x'. While we found PWC-Net to be robust to substantial color transformations between the input images, we use the processed RAW x' as input as it has a much smaller difference in color and intensity to the reference color image c. Further, the loss masking discussed next aids in a more robust loss calculation for inaccurately aligned regions.

Loss masking: Although, employing an aligned L_1-loss partially handles the misalignment problem for ISP learning in the wild, the flow estimation itself can introduce errors. In particular, optical flow is often inaccurate in the presence of repeating patterns, occlusions, and homogeneous regions. This leads to an

incorrect training signal which degrades the quality of the ISP network. We therefore propose a mask for our loss by identifying regions where the optical flow is inaccurate. Inspired by [15], we use the forward-backward consistency constraint to filter out regions with inaccurate flow. The optical-flow consistency mask m is set to 1 where the following condition holds true, and otherwise to 0:

$$\left|f(x', y^{\downarrow}) + f(x'_{y^{\downarrow}}, x')\right|^2 < \alpha_1 \left(\left|f(x', y^{\downarrow})\right|^2 + \left|f(x'_{y^{\downarrow}}, x')\right|^2 \right) + \alpha_2. \tag{5}$$

Here, x' is the processed RAW sensor data (Sec. 3.3). And, $y^{\downarrow}$ is the target sRGB image bilinearly downsampled by a factor of 2. And, $x'_{y^{\downarrow}}$ is x' aligned with $y^{\downarrow}$. Thus, the mask m aids in masking out inaccurately aligned regions.

ISP Network Loss: The masked target sRGB prediction loss is given by:

$$\begin{aligned} \hat{y} &= \mathcal{F}(x, \hat{c}), \text{ where } \hat{c} = \mathcal{C}(\tilde{x}, c_{\tilde{x}}) \\ L_{\text{pred}}(\hat{y}, y) &= \|m^{\uparrow} \odot \left(y_{\hat{y}} - \hat{y}\right)\|_1. \end{aligned} \tag{6}$$

Here, $y_{\hat{y}}$ is the target DSLR sRGB aligned w.r.t. the final predicted sRGB $\hat{y}$. We did not see a significant difference in performance when we align the predicted sRGB $\hat{y}$ w.r.t. the target DSLR sRGB for our loss calculation (see supplementary). This choice further circumvents the need of differentiating through the warping process. During training, the color image $c = y^{\downarrow}$ is the $2\times$ downsampled ground truth sRGB. Further, $c_{\tilde{x}}$ is the color image c aligned with $\tilde{x}$ (Eq. 2). Lastly, $m^{\uparrow}$ is the $2\times$ upsampled mask m via nearest neighbour interpolation.

Pre-processing Network Loss: The pre-processing net (Sec. 3.3) aims at providing a source image that aids our learned parametric color mapping scheme (Sec. 3.3) and denoising the processed RAW x' (Sec. 3.3). Motivated by this, we design loss for our pre-processing net $\mathcal{P}$ as,

$$\begin{aligned} L_{\text{map}}(\mathcal{C}(\tilde{x}, c_{x'}), c_{x'}) &= \|m \odot (\mathcal{C}(\tilde{x}, c_{x'}) - c_{x'})\|_1, \text{ and} \\ L_{\text{constraint}}(x', \tilde{x}) &= \|b * x' - b * \tilde{x}\|_1. \end{aligned} \tag{7}$$

Here, $\tilde{x}$ is the output of our Pre-processing Net (Eq. 2) and b is a predefined blurring kernel. The loss $L_{\text{constraint}}$ constrains $\mathcal{P}$ to keep the color of x'. The color image $c = y^{\downarrow}$ is the $2\times$ downsampled ground truth sRGB. And, $c_{x'}$ is the color image c aligned with x'. These losses aid the pre-processing network in not only denoising the RAW sensor data but also allows for the network to be flexible enough to learn a color space where the color mapping (Sec. 3.3) is optimal.

Color Prediction Network Loss: To train our target color prediction network (Sec. 3.2), we employ a color prediction loss on the predicted low resolution target color image $\hat{y}^{\text{clr}} = \mathcal{G}(x)$ and a reconstruction loss on the reconstructed RAW sensor data $\hat{x}$,

$$\begin{aligned} L^{\text{clr}}_{\text{pred}}(\hat{y}^{\text{clr}}, c_{x'}) &= \left\| m \odot \left(\hat{y}^{\text{clr}} - c_{x'}\right) \right\|_1 \\ L_{\text{reconstruct}}(\hat{x}, x) &= \|x - \hat{x}\|_1 \end{aligned} \tag{8}$$

Here, $c_{x'} = y^{\downarrow}_{x'}$ is the 2× downsampled ground truth sRGB aligned with x'. Hence, $c_{x'}$ serves as the target color image for training our color prediction network in the loss $L^{\text{clr}}_{\text{pred}}$. The reconstruction loss $L_{\text{reconstruct}}$ further encourages the encoder $E_{\text{phone}}(x)$ to preserve important image details.

Training: Thanks to the independent objectives, we can train our color conditional ISP network $\mathcal{F}$ and the color prediction network $\mathcal{G}$ separately. This allows use of larger batch sizes and reduced training times significantly. A comparative study with the joint fine-tuning of both the networks is provided in the supplementary. The final training loss for $\mathcal{F}$ is given by (6) and (7). The loss for the color prediction net $\mathcal{G}$ is given by (8). Each batch for training both, $\mathcal{F}$ and $\mathcal{G}$ comprises 16 pairs of randomly sampled RAW phone images $x \in \mathbb{R}^{4\times80\times80}$ and DSLR sRGB images $y \in \mathbb{R}^{3\times160\times160}$. During training, we augment the data by applying random flips and 90 deg rotations. To increase the robustness of our color conditional ISP network $\mathcal{F}$, we employ color augmentations on the ground truth DSLR sRGB during training. Specifically, we randomly jitter the hue, saturation, brightness and contrast in a range $[-0.2, 0.2]$.

The blurring kernel b in (7) is a 9×9 Gaussian with the standard deviation in each of the dimension set to 2. The constants α_1 and α_2 for computing m are set to 0.01 and 0.5, respectively. The number of bins $\mathcal{B}$ in our color mapping 3.3 is set to 15 and the temperature parameter $T = (1/\mathcal{B})^2$. Finally, to handle vignetting (dark corners) that occurs in RAW sensor data, we append the RAW data with a pixel-wise function of 2D coordinate map for the inputs to our pre-processing net $\mathcal{P}$ and the color prediction net $\mathcal{G}$. We use the ADAM algorithm [12] as optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate for training both our networks is set to $2e-4$ which is halved at 50%, 75%, 90% and 95% of the total number of epochs respectively. The networks are trained separately for 100 epochs on a Nvidia V100 GPU. The training time for our $\mathcal{F}$ and $\mathcal{G}$ nets was 27 hours and 22 hours, respectively.

4 Dataset

We propose the ISP in the Wild (ISPW) dataset for learning the camera ISP in the wild. The ISPW dataset consists of a set of 197 high-resolution captures from a Canon 5D Mark IV DSLR camera (with a lens of focal length 24mm) and a Huawei Mate 30 Pro mobile phone. Each capture comprises of the RAW sensor data from the mobile phone (4×1368×1824) and 3 sRGB DSLR images (3×4480×6720) of the same scene taken at different exposure settings (EV values: -1, 0 and 1). All DSLR images were captured with an ISO of 100 for more detail and less noise. Further a small aperture of F18 was used for a large depth of field. The dataset was collected over several weeks in a variety of places and in various illumination and weather conditions to ensure diversity of samples. During the capture, both the devices were mounted on a tripod using a custom made rig to ensure no blur due to camera motion. Collection was focused on predominately static scenes in order to ease the alignment between the two cameras. However, small motion is inevitable in most settings, and thus need to be handled by our data processing and robust learning objectives. We split

the ISPW dataset into 160, 17, and 20 high-resolution captures for training, validation, and test, respectively. We believe that it can serve as an important benchmarking and training set for RAW-to-sRGB mapping in the wild.

Data processing: We describe the pre-processing pipeline for our ISPW data here. We consider the DSLR image taken at EV value 0 as the target DSLR sRGB in this work. We first crop out the matching field of view from the phone and the DSLR high-resolution captures using SIFT [14] and RANSAC [5]. Crops of size 320×320 are then extracted in a sliding manner (stride of 160) from both, the DSLR sRGB and the phone sRGB. Local alignment is performed by estimating the homography between two crops. The corresponding 4-channel RAW crop from the phone of size 160×160 is extracted using the coordinates of the 320×320 phone sRGB crop and paired with the DSLR sRGB crop. To filter out crops with extreme scene mismatch, we discard the pairs which have a normalized cross-correlation of less than 0.5 between them.

5 Experiments

Here, we perform extensive experiments to validate our approach. We evaluate our approach on the test sets of the ZRR dataset [8] and ISPW datasets (Sec. 4). The methods are compared in terms of the widely used PSNR and SSIM [20] metrics. For a fair comparison, we align the ground truth DSLR sRGB with the phone RAW for the computation of PSNR and SSIM metrics. See supplementary for more qualitative and quantitative results,

5.1 Ablative Analysis of the Color Mapping

Here, we study the effectiveness of our color mapping scheme (Sec. 3.3) compared to other alternatives. The results on the ZRR dataset are reported in Tab. 1.

NoColorPred: As a baseline for evaluating our color mapping scheme, we train $\mathcal{F}(x, \hat{c})$ with the color information $\hat{c}$ set to 0. This implies a simple feed-forward network setting. We do not include the color mapping module $\mathcal{C}$ in this version. NoColorPred achieves a PSNR of 21.27 dB and a SSIM of 0.844. This variation learns average average and dull colors and is not able to account for various factors on which the color in an image depends. **ColorBlur:** Next, as in CycleISP [22], we train $\mathcal{F}(x, \hat{c})$ where the target color $\hat{c} = z * y^{\downarrow}_{x'}$ is achieved by blurring the 2x downsampled target DSLR sRGB (aligned with x') with a Gaussian kernel z during training. At inference, we apply the same blurring to our predicted target color $\hat{c} = z * \mathcal{G}(x)$. As in NoColorPred, we do not include the color mapping module $\mathcal{C}$ in this version. ColorBlur achieves a gain of 2.16 dB in PSNR over NoColorPred. Although being better than NoColorPred, ColorBlur fails to capture the sudden changes of color in the image contour.

We further evaluate different versions of the color mapping scheme $\mathcal{C}$. **LinearMap:** First, we consider learning a 3×3 global color correction matrix between the processed RAW x' and the color c for each training pair, as in [2]. LinearMap produces inaccurately colored images specially in terms of the contrast, since it cannot represent more complex color transformations and tone curves.

Table 1. Ablative study of our color mapping scheme (Sec. 3.3) on the ZRR dataset.

	NoColorPred	ColorBlur	LinearMap	ConstValMap	AffineMapIndep	AffineMapDep	+Preprocess
PSNR↑	21.27	23.43	21.89	22.65	23.78	24.41	25.24
SSIM↑	0.844	0.857	0.832	0.859	0.861	0.873	0.879

Table 2. Ablative study of our loss (Sec. 3.4) on the ZRR dataset.

	NoAlign	+AlignedLoss	+Mask
PSNR↑	20.56	24.62	25.24
SSIM↑	0.785	0.867	0.879

Table 3. Ablative study of our color prediction network (Sec. 3.2) on the ZRR dataset

	NoColorPred	+U-Net	+Reconstruct	+GlobalContext
PSNR↑	21.27	24.09	24.43	25.24
SSIM↑	0.844	0.865	0.871	0.879

ConstValMap: Here, we use a simplified version of our approach (Sec. 3.3) as $\mathcal{C}$ by using fixed values for each bin instead of the affine mapping learned in Sec. 3.3. Channel dependence is not exploited in this version for calculating the values. This achieves a substantial improvement of 0.76 dB in PSNR over LinearMap. Thus, proving the utility of using a more flexible color mapping formulation. **AffineMapIndep:** Setting $\mathcal{C}$ to our color mapping scheme (Sec. 3.3) but without any channel dependence boosts the PSNR by a further 1.13 dB over ConstValMap. Increasing the expressive power of each bin by predicting an affine transform instead of a constant is thus pivotal for better performance of our color conditional RAW-to-sRGB mapping. **AffineMapDep:** Here, $\mathcal{C}$ is set to our full formulation discussed in Sec. 3.3. Thus, exploiting channel dependence in $\mathcal{C}$ is beneficial as quantified by the PSNR increase of 0.63 dB w.r.t.AffineMapIndep. **+Preprocess:** Finally, we add our pre-processing network $\mathcal{P}$ (Sec. 3.3) to the AffineMapDep version. This gives an impressive boost of 0.83 dB in PSNR over AffineMapDep hence, validating the need to remove noise and pre-process the phone RAW before color mapping.

5.2 Ablative Study of the Training Loss

Here, we study the effect of our masked aligned loss (Sec. 3.4). The results on the ZRR dataset are reported in Tab. 2.

NoAlign: As a baseline for ablating our loss, we employ an unaligned L_1-loss for all our objectives (Eq. (6), (7) and (8)). The mask m is set to 1 at all locations. **+AlignedLoss** Further, employing alignment before the loss calculation leads to more crisp predictions, giving a large improvement of 4.06 dB in PSNR and a relative gain of 10.4% in SSIM. Although improving the results, the prediction lacks detail and is characterized by a noticeable color shift. This is due to the inaccuracies in optical flow computations that may occur due to occlusions and homogenous regions. **+Mask** Finally, our masking strategy using Eq. (5) leads to a significant gain of 0.62 dB in PSNR. (+Mask) produces a more detailed output with colors consistent with the target DSLR sRGB. This shows that accurate supervision using our masked loss during training is beneficial to our DSLR sRGB restoration network.

5.3 Ablative Study of the Color Prediction Network

Next, we study the effect of our color prediction module (Sec. 3.2). The results on the ZRR dataset are reported in Tab. 3.

(a) MWISP-Net (b) AWNet (c) LiteISP-Net (d) Ours (e) OursFast (f) GT

Fig. 4. Visual results for state-of-the-art comparison on our ISPW dataset (first row) and the ZRR dataset (second row). Best viewed with zoom.

NoColorPred: This is the same baseline as in Sec. 5.1, which employs no explicit color prediction or conditioning. **U-Net:** Integrating a low resolution U-Net based color predictor without the reconstruction branch or global context transformer leads to an impressive gain of 2.82 dB over NoColorPred. This demonstrates the effectiveness of conditioning $\mathcal{F}$ on the color image for robust ISP learning and prediction. **+Reconstruct:** Further, integrating a reconstruction branch in our color predictor helps $\mathcal{G}(x)$ in learning a more informative encoding $E_{\text{phone}}(x)$, leading to a 0.34 dB increase in PSNR. Thus, +Reconstruct facilitates our encoder in the color predictor module to encapsulate all the information into the encoding that is necessary for accurate color prediction. **+GlobalContext:** Finally, integrating the global context transformer (Sec. 3.2) in our U-Net color predictor $\mathcal{G}(x)$ provides our color conditional ISP net $\mathcal{F}(x, \hat{c})$ with a substantial gain of 0.81 dB. This clearly demonstrates the importance of exploiting global information in predicting coherent colors.

5.4 State-of-the-Art Comparison

In this section, we compare our color conditional ISP network with state-of-the-art methods for RAW-to-sRGB mapping, namely PyNet [6], MW-ISPNet [8], AWNet [4] and LiteISPNet [24]. We evaluate on the test splits of the ZRR dataset [8] and our ISPW dataset (Sec. 4). Among these methods, MW-ISPNet, AWNet and LiteISPNet employ discrete wavelet transforms for incorporating global context. To deal with misalignments, MW-ISPNet, AWNet and PyNet incorporate the VGG perceptual loss [11], while LiteISPNet employs an aligned loss using optical flow computation [18].

Table 4 lists the quantitative results on the test split of the ZRR dataset that contains 1203 RAW-sRGB crop pairs of size 448×448. Our method outperforms all previous approaches by a significant margin, achieving a gain of 1.43 dB PSNR compared to the second best method: the very recent LiteISPNet. We then run the best performing methods on the test split of our ISPW dataset, that contains 3023 RAW-sRGB crop pairs of size 320×320. For a fair comparison, all the methods were retrained on our dataset using apt train settings. The

Table 4. State-of-the-Art comparison on the ZRR [8] and our ISPW datasets.

		ZRR Dataset			ISPW Dataset		
	#Params(M)↓	PSNR↑	SSIM↑	Time(ms)↓	PSNR↑	SSIM↑	Time(ms)↓
PyNet [6]	47.6	22.73	0.845	62.7	-	-	-
MW-ISPNet [8]	29.2	23.13	0.849	111.3	22.43	0.746	99.4
AWNet [4]	52.2	23.52	0.855	63.4	23.10	0.787	50.8
LiteISPNet [24]	**11.9**	23.81	0.873	**23.3**	23.51	0.809	**17.2**
Ours	35.2	**25.24**	**0.879**	67.6	**25.05**	**0.821**	55.7
OursFast	**13.4**	**24.70**	**0.876**	**18.2**	**24.57**	**0.815**	**13.8**

performance gap between our color conditional ISP network and other methods is more stark for the ISPW dataset, with our approach achieving a PSNR 1.54 dB higher than the second best LiteISPNet. Efficiency is crucial for deploying the model on a smartphone. We therefore evaluate a *faster and lighter* version of our approach. In OursFast, we omit the color mapping regularization $\mathcal{C}$ after the color prediction network. We further reduce the number of parameters by $\sim 3\times$, by reducing the depth of all three networks and dimensionality of the Global Context Block. OursFast outperforms LiteISPNet by 1.06dB while being 20.2% faster on the ISPW dataset. Further details on the model complexity and execution times are given in the supplementary.

Figure 4 shows the visual results for our color conditional ISP compared to the top three performing methods. Compared to our approach, all the other three methods fail to capture the accurate color of the target DSLR sRGB. Moreover, the results for MW-ISPNet and AWNet are blurry due to their inability to handle misalignment well. On the other hand, although LiteISPNet employs an aligned loss, it fails to account for inconsistent flow computations hence leading to significant color shift and loss of detail. Conversely, our approach produces crisp DSLR-like sRGB predictions with accurate colors, thus proving the utility of our global attention based color predictor paired with our masked aligned loss. The blur and color shift effect is more intense for all other methods on our dataset that contains misaligned RAW-sRGB pairs. Finally, we calculate the average inference time per image for our method on both the datasets. We achieve an average per image inference times of 67.6 ms and 55.7 ms, respectively on the sRGB images of sizes 448×448 (ZRR dataset) and 320×320 (ISPW dataset).

6 Conclusion

We address the problem of mapping RAW sensor data from a phone to a high quality DSLR image by modelling it as a conditional ISP framework on the target color. To aid our color conditional ISP net during inference, we propose a novel encoder-decoder based color predictor that encapsulates an efficient global attention module. A flexible parametric color mapping scheme from RAW to the target color is integrated for a robust training and inference. Finally, we propose a masked aligned loss for filtering out regions with inconsistent optical flow during aligned loss calculations. We perform experiments on the ZRR dataset and our ISPW dataset, setting a new state-of-the-art on both the datasets.

Acknowledgement: This work was supported by the ETH Zürich Fund (OK), Huawei Technologies Oy (Finland) and Alexander von Humboldt Foundation.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 3
2. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Deep burst super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 9209–9218. Computer Vision Foundation / IEEE (2021) 3, 4, 11
3. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input / output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (2011) 3
4. Dai, L., Liu, X., Li, C., Chen, J.: Awnet: Attentive wavelet network for image ISP. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12537, pp. 185–201. Springer (2020). https://doi.org/10.1007/978-3-030-67070-2_11, https://doi.org/10.1007/978-3-030-67070-2_11 1, 3, 13, 14
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). https://doi.org/10.1145/358669.358692, http://doi.acm.org/10.1145/358669.358692 11
6. Ignatov, A., Gool, L.V., Timofte, R.: Replacing mobile camera ISP with a single deep learning model. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020. pp. 2275–2285. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPRW50498.2020.00276 13, 14
7. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Dslr-quality photos on mobile devices with deep convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3277–3285 (2017) 3
8. Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., Dai, L., Liu, X., Li, C., Chen, J., Ito, Y., Vasudeva, B., Deora, P., Pal, U., Guo, Z., Zhu, Y., Liang, T., Li, C., Leng, C., Pan, Z., Li, B., Kim, B., Song, J., Ye, J.C., Baek, J., Zhussip, M., Koishekenov, Y., Ye, H.C., Liu, X., Hu, X., Jiang, J., Gu, J., Li, K., Tang, P., Hou, B.: AIM 2020 challenge on learned image signal processing pipeline. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12537, pp. 152–170. Springer (2020). https://doi.org/10.1007/978-3-030-67070-2_9, https://doi.org/10.1007/978-3-030-67070-2_9 1, 3, 4, 11, 13, 14
9. Jaegle, A., Borgeaud, S., Alayrac, J., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Hénaff, O.J., Botvinick, M.M., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver IO: A general architecture for structured inputs & outputs. CoRR **abs/2107.14795** (2021), https://arxiv.org/abs/2107.14795 6
10. Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver: General perception with iterative attention. CoRR **abs/2103.03206** (2021), https://arxiv.org/abs/2103.03206 6
11. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands,

October 11-14, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9906, pp. 694–711. Springer (2016). https://doi.org/10.1007/978-3-319-46475-6_43 13
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980 10
13. Liu, P., Zhang, H., Lian, W., Zuo, W.: Multi-level wavelet convolutional neural networks. IEEE Access **7**, 74973–74985 (2019) 3
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999. pp. 1150–1157. IEEE Computer Society (1999). https://doi.org/10.1109/ICCV.1999.790410, https://doi.org/10.1109/ICCV.1999.790410 11
15. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 7251–7259. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16502 9
16. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1586–1595 (2017) 3
17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., III, W.M.W., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer (2015). https://doi.org/10.1007/978-3-319-24574-4_28, https://doi.org/10.1007/978-3-319-24574-4_28 5
18. Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. CoRR **abs/1709.02371** (2017), http://arxiv.org/abs/1709.02371 8, 13
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017) 6
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861, https://doi.org/10.1109/TIP.2003.819861 11
21. Xing, Y., Qian, Z., Chen, Q.: Invertible image signal processing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 6287–6296. Computer Vision Foundation / IEEE (2021) 3
22. Zamir, S.W., Arora, A., Khan, S.H., Hayat, M., Khan, F.S., Yang, M., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle,

WA, USA, June 13-19, 2020. pp. 2693–2702. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00277 3, 11
23. Zhang, X., Chen, Q., Ng, R., Koltun, V.: Zoom to learn, learn to zoom. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3762–3770. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00388 3
24. Zhang, Z., Wang, H., Liu, M., Wang, R., Zhang, J., Zuo, W.: Learning raw-to-srgb mappings with inaccurately aligned supervision. CoRR **abs/2108.08119** (2021), https://arxiv.org/abs/2108.08119 1, 2, 3, 13, 14