A ISP revisited

Here we give a brief review of the digital camera image formation process, from camera sensor RAW data to output sRGB, please refer to Fig. 1(a) for an illustrative diagram, the ISP steps mainly include:

(a). Pre-processing involves some pre-process operations such as BlackLevel adjustment, WhiteLevel adjustment, and lens shading correction.

(b). Noise reduction eliminates noise and keeps the visual quality of image, this step is closely related to exposure time and camera ISO settings [47,68].

(c). Demosaicing is used to reconstruct a 3-channel color image from a single-channel RAW, executed through interpolation of the absent values in the Bayer pattern, relying on neighboring values in the CFA.

(d). White Balance simulates the color constancy of human visual system (HVS). An auto white balance (AWB) algorithm estimates the sensor's response to illumination of the scene and corrects RAW data.

(e). Color Space Transformation mainly includes two steps, first is mapping white balanced pixel to un-render color space (*i.e.* CIEXYZ), and the second is mapping un-render color space to the display-referred color space (*i.e.* sRGB), typically each use a 3×3 matrix based on specific camera [16].

(f). Color and Tone Correction are often implemented using 3D and 1D lookup tables (LUTs), while tone mapping also compresses pixel values.

(g). Sharpening enhances image details by unsharp masking or deconvolution.

We refer other detailed steps such as digital zoom and gamma correction to previous works [16,35,56]. Meanwhile, in the ISP pipeline, many other operations prioritize the quality of the generated image rather than its performance in machine vision tasks. Therefore, for specific adapter designs, we selectively omit certain steps and focus on including the steps mentioned above. We provide detailed explanations in the Sec. 3.1 and Sec. C.

B Impact of Different Blocks

We conducted ablation experiments to assess the effectiveness of different stages in RAW-Adapter. The experiments were designed on the PASCAL dataset with RetinaNet [43] (ResNet-50 backbone), covering normal, dark, and over-exposed conditions. The results are presented in Table. B5, we can find that the kernel predictor $\mathbb{P}_{\mathbb{K}}$ exhibits significant improvements in dark scenarios (+2.4), attributable to the gain ratio g and denoising processes, but it doesn't seem to be of much help in both overexposed and normal scenes (+0.0), this might be due to the current kernel-based denoising methods being too simplistic and eliminating some detail information. Meanwhile the implicit LUT $\mathbb{L}$ does not show improvement under over-exposed and low-light conditions but proves effective in normal light condition. Finally, the model-level adapters $\mathbb{M}$ and matrix predictor $\mathbb{P}_{\mathbb{M}}$ yield performance improvements across all scenarios.

Table B5: Ablation analyze on RAW-Adapter's model structure.

	base	$\mathbb{P}_\mathbb{K}$	$\mathbb{P}_\mathbb{M}$	$\mathbb{L}$	$\mathbb{M}$	mAP (normal)	mAP (over-exp)	mAP (dark)
blocks	✓					89.2	88.8	82.6
	✓	✓				89.2 (+0.0)	88.8 (+0.0)	85.0 (+2.4)
	✓	✓	✓			89.3 (+0.1)	89.0 (+0.2)	86.2 (+3.6)
	✓	✓	✓	✓		89.4 (+0.2)	89.0 (+0.2)	86.3 (+3.7)
	✓	✓	✓	✓	✓	89.6 (+0.4)	89.4 (+0.6)	86.6 (+4.0)

C Detailed Design of Input-level Adapters

In the main text of our paper, we outlined that the input level adapters of RAW-Adapter comprise three components: the kernel predictor $\mathbb{P}_\mathbb{K}$, the matrix predictor $\mathbb{P}_\mathbb{M}$, and the neural implicit 3D LUT $\mathbb{L}$. In this section, we will provide a detailed explanation of how to set the parameter ranges for input-level adapters, along with conducting some results analysis.

The kernel predictor $\mathbb{P}_\mathbb{K}$ is responsible for predicting five ISP-related parameters, including the ① gain ratio g, the Gaussian kernel ② k's major axis radius r_1, ③ k's minor axis radius r_2, and the ④ sharpness filter parameter σ.

① The gain ratio g is used to adjust the overall intensity of the image $\mathbf{I}_1$. Here g initialized to 1 under normal light and over-exposure conditions. In low-light scenarios, g is initialized to 5.

② The major axis radius r_1 is initialized as 3, and we predict the bias of the variation of r_1, then add it to r_1.

③ The minor axis radius r_2 is initialized as 2, and we predict the bias of the variation of r_2, then add it to r_2.

④ The sharpness filter parameter σ is constrained by a Sigmoid activation function to ensure its range is within (0, 1).

The matrix predictor $\mathbb{P}_\mathbb{M}$ is responsible for predicting ⑤ a white balance related parameter ρ and ⑥ white balance matrix $\mathbf{E}_{ccm}$ (9 parameters). In total, 10 parameters need to be predicted.

⑤ ρ is a hyperparameter of the Minkowski distance in SOG [20] white balance algorithm. We set its minimum value to 1 and then use a ReLU activation function followed by adding 1 to restrict its range to (1, $+\infty$).

⑥ The matrix $\mathbf{E}_{ccm}$ consists of the 9 parameters predicted by $\mathbb{P}_\mathbb{M}$ and forms a 3x3 matrix. No activation function needs to be added, it would directly added to the identity matrix $\mathbf{E}_3$ to form the final $\mathbf{E}_{ccm}$.

For the neural implicit 3D LUT (NILUT) [11] $\mathbb{L}$, in the main text, we set the MLP dimension of the neural implicit 3D LUT $\mathbb{L}$ to 32 to save FLOPs, here we test the effects of different dims of $\mathbb{L}$ on the final results, as shown in Table C6. Compared to the MLP dimension of 32 in the main text, we observed that setting the MLP dimension of NILUT [11] to 16 leads to a decrease in performance. Increasing the LUT dimension to 64 results in a slight improvement in performance, but further increasing it to 128 does not lead to performance

Table C6: Ablation analyze on neural implicit 3D LUT $\mathbb{L}$'s dims, we shoe the efficiency comparison (# Para and Flops), along with mAP comparison on ADE 20K RAW dataset. Flops are calculated from a tensor of size (1, 3, 512, 512).

$\mathbb{L}$	# Para ↓	Flops ↓	mIOU (normal)	mIOU (over-exp)	mIOU (dark)
dim=16	0.93K	0.207G	47.97 (−0.41)	46.95 (−0.11)	41.02 (−0.80)
dim=32	1.97K	0.784G	48.38 (+0.00)	47.06 (+0.00)	41.82 (+0.00)
dim=64	12.9K	3.041G	48.44 (+0.06)	47.26 (+0.20)	41.82 (+0.00)
dim=128	50.4K	11.98G	48.40 (+0.02)	47.05 (−0.01)	41.75 (−0.05)

enhancement. Additionally, as the MLP dimension increases, both the parameter number and FLOPs of $\mathbb{L}$ increase substantially. Therefore, in the experiments of RAW-Adapter, choosing a dimension of 32 for $\mathbb{L}$ is a more reasonable option.

D Segmentation with Swin-Transformer

In the main text, we evaluated the segmentation performance using the Segformer [70]. Here, we extend the evaluation to include the Swin-Transformer [45] backbone. In the default setting of `mmsegmentation`, the segmentation head comprises UperNet with an auxiliary segmentation head FCN. We compare different sizes of the Swin-Transformer backbone, including Swin-tiny (Swin-T), Swin-small (Swin-S), and Swin-normal (Swin-N). The training settings and comparison metrics remain the same as Table 4 in the main text. Additionally, we include various ISP methods [6,34,35,71,79] and Dirty-Pixel [18] for comparison.

The comparison results are shown in Table. F7, we can find that RAW-Adapter still achieves the best results on the Swin-Transformer [45] architecture. Our segmentation performance remains optimal across all lighting conditions, at the same time, our approach also holds advantages in terms of parameter count and inference speed. Furthermore, it achieves superior performance even on lighter weight Swin backbones compared to other methods on heavier weight Swin backbones. This also demonstrates that RAW-Adapter can serve as a general approach for extension across different network architectures.

E Segmentation on Real-World Dataset [40]

Additionally, we made the experiments on real-world RAW semantic segmentation dataset iPhone XSmax [40], iPhone XSmax consist of 1153 RAW images with their corresponding semnatic labels, where 806 images are set as training set and the other 347 images are set as evaluation set. We adopt Segformer [70] framework with MIT-B5 backbone, training iters are set to 20000 and other settings are same as ADE 20K RAW's setting. The experimental results are shown in Fig. E7. RAW-Adapter method could also achieve satisfactory results.

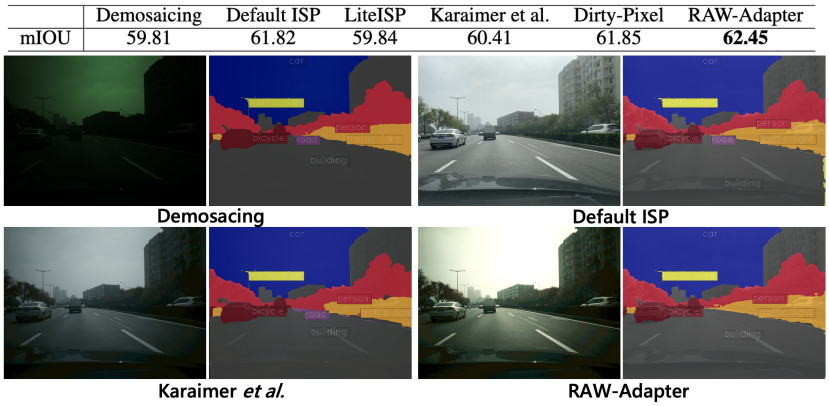

Fig. E7: Semantic segmentation results on iPhone XSmax [40] dataset.

F Limitation of Current Design

Input-level Adapter still adopts simple kernel-based denoising and sharpening methods, this approach is considered for saving computational costs and for simplicity in design, however, we believe that perhaps more advanced denoising methods could bring about better improvements. Another part is that the implicit 3D LUT [11] is not designed to be image-adaptive, instead, it is a fixed LUT learned from the same dataset, we believe that perhaps an image-adaptive LUT could lead to better improvements, as different images within the same dataset can still have significant variations in information and lighting conditions.

Model-level Adapter's integration method is still relatively simple. We have extracted intermediate images from the ISP process (I1, I2, I3, I4) to serve as guidance information for designing the model-level adapter. We use the convolution process to simply fuse I1, I2, I3, I4 together to assist the network backbone. We believe that perhaps more effective feature fusion method [19] could better help improve the performance of downstream tasks.

G More Visualization Results

We show more visualization results in this section. The detection results are shown in Fig. G8, where line $1 \sim 6$ are the detection results on LOD [27] dataset and line $7 \sim 8$ are the detection results on PASCAL RAW [52] dataset, we show the comparison with ISP methods Karaimer *et al.* [35] and InvISP [71], along with joint-training method Dirty-Pixel [18]. The segmentation results are shown in Fig. G9, with comparison of various methods [6, 18, 35, 71, 79].

Table F7: Comparison on Swin-Transformer [45] in **ADE 20K RAW** (normal/over-exp/dark). **Bold** denotes the best result while underline denotes second best result. The same background color in the table indicates same backbone weights.

	backbone	params(M) ↓	inference time(s) ↓	mIOU ↑ (normal)	mIOU ↑ (over-exp)	mIOU ↑ (dark)
Demosacing	Swin-N	121.27	0.168	43.71	40.86	31.22
Karaimer *et al.* [35]			0.588	44.97	41.21	34.84
InvISP [71]			0.266	43.02	41.87	6.01
LiteISP [79]			0.324	44.52	40.61	4.81
DNF [34]			0.249	-	-	30.77
SID [6]			0.375	-	-	24.92
Dirty-Pixel [18]		125.47	0.239	44.81	43.48	38.32
	Swin-S	85.31	0.138	36.29	35.01	27.66
	Swin-T	63.92	0.057	35.21	32.47	25.91
RAW-Adapter (w/o $\mathbb{M}$)	Swin-N	121.35	0.209	45.01	43.72	38.44
	Swin-S	81.29	0.106	37.11	34.94	28.63
	Swin-T	**59.98**	**0.049**	36.30	32.38	26.45
RAW-Adapter	Swin-N	121.81	0.238	**45.97**	**44.86**	**39.78**
	Swin-S	81.65	0.112	38.02	37.01	28.99
	Swin-T	60.29	0.063	36.85	33.00	26.77

Fig. G8: Object detection results on LOD [27] (line 1 $\sim$ 6) and PASCAL RAW [52] (line 7 and line 8), please zoom in to see details.

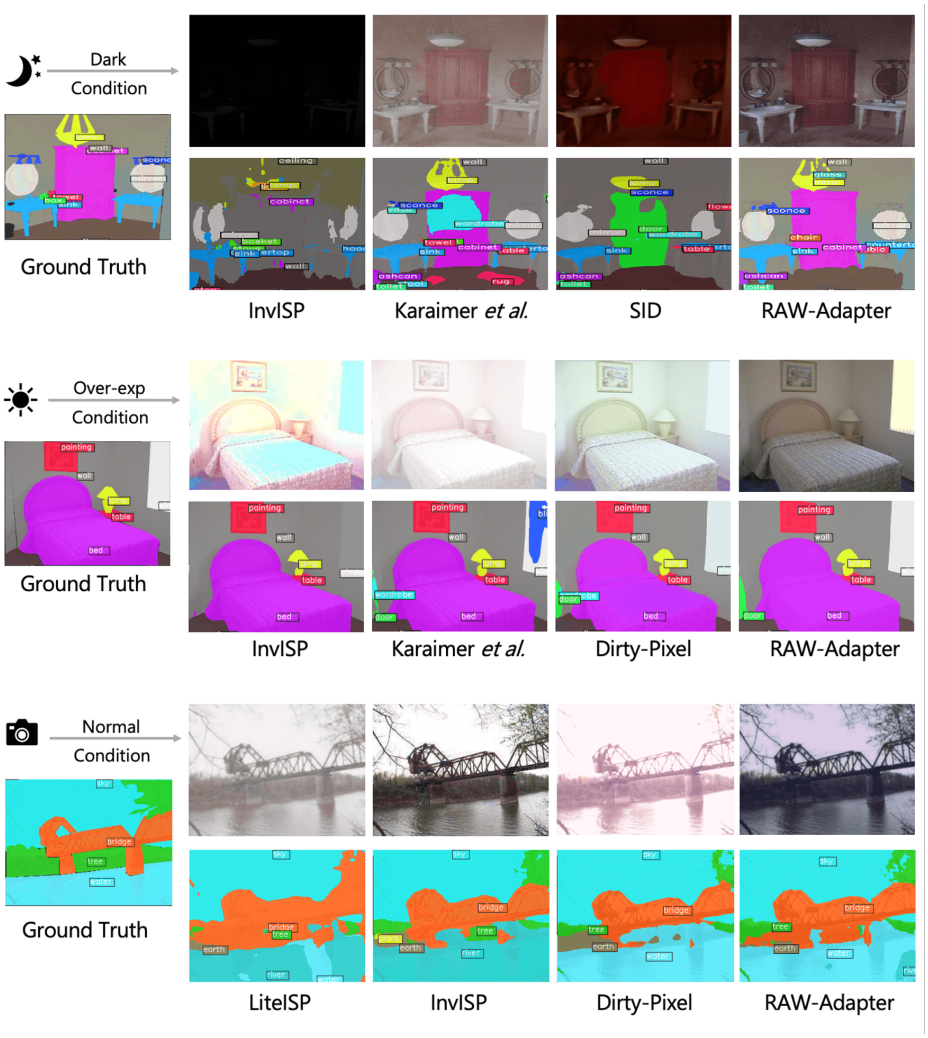

Fig. G9: Segmentation results on ADE20K RAW dataset, including dark scene, over-exposure scene and normal scene.