Zero-Reference Low-Light Enhancement via Physical Quadruple Priors

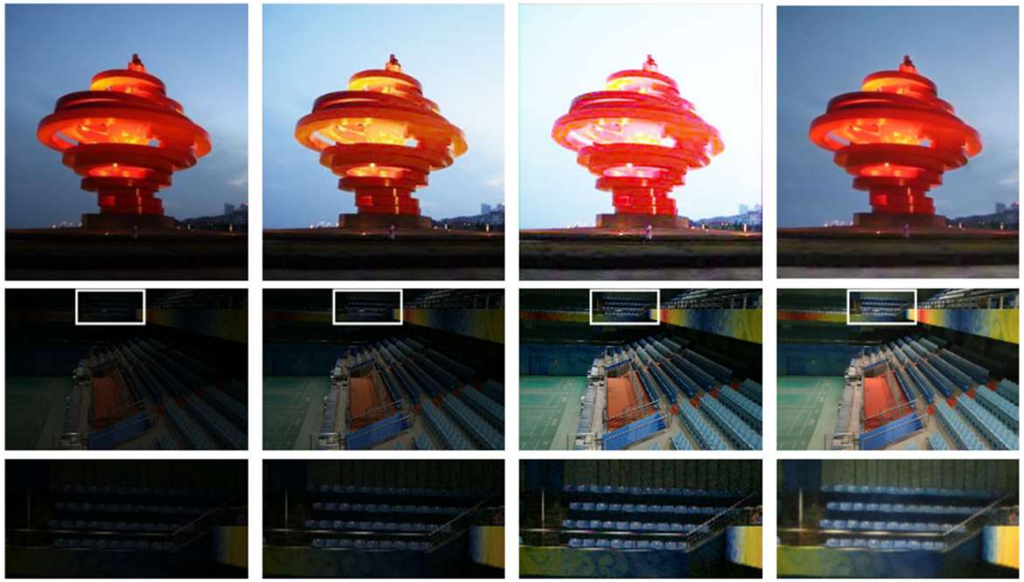
Wenjing Wang
Peking University

Huan Yang
01.AI

Jianlong Fu
Microsoft Research Asia

Jiaying Liu *
Peking University

CVPR 2024



(a) Input (b) SCI-MIT (c) SCI-LOL (d) Ours

Figure 1. Comparison with a SOTA zero-reference method: SCI [34]. The SCI model, trained on varied datasets like LOL [48] and MIT [2], yields diverse enhancement results. Nevertheless, none effectively maintains a consistent lighting effect across both dark and moderately dark images. In contrast, our model demonstrates greater robustness across various scenarios.

Overall Methodology

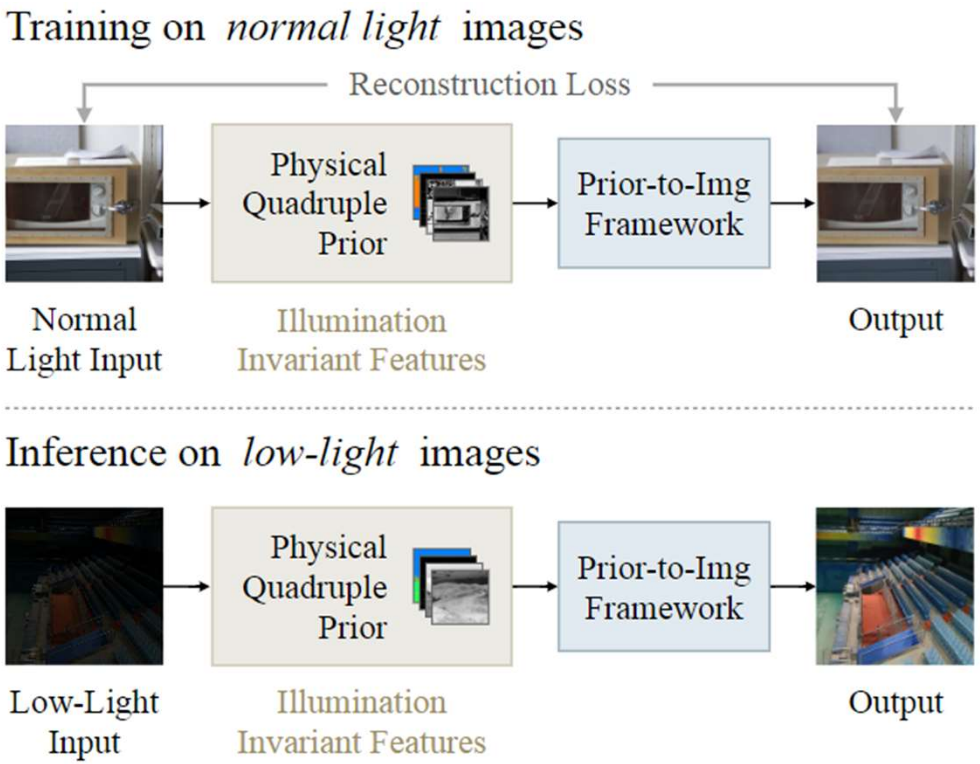


Figure 2. The overall methodology of our zero-reference low-light enhancement approach. Our model is trained to reconstruct images from an illumination-invariant prior (the physical quadruple prior) in the normal light domain. During testing, the model extracts illumination-invariant priors from low-light images and reconstructs them into normal light images.

Learnable Illumination-Invariant Prior

KubelkaMunk theory:

$$E(\lambda, \mathbf{x}) = e(\lambda, \mathbf{x}) \left((1 - i(\mathbf{x}))^2 R_\infty(\lambda, \mathbf{x}) + i(\mathbf{x}) \right), \quad (1)$$

$$i(\mathbf{x}) \approx 0$$

$$E(\lambda, \mathbf{x}) = e(\lambda, \mathbf{x}) R_\infty(\lambda, \mathbf{x}), \quad (2)$$

Retinex theory: $X = I \odot R$

$$E^\lambda = \frac{\partial E(\lambda, \mathbf{x})}{\partial \lambda}, \; R_\infty^\lambda = \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda}, \quad (3)$$

$$E^{\lambda\lambda} = \frac{\partial^2 E(\lambda, \mathbf{x})}{\partial \lambda^2}, \; R_\infty^{\lambda\lambda} = \frac{\partial^2 R_\infty(\lambda, \mathbf{x})}{\partial \lambda^2}. \quad (4)$$

Learnable Illumination-Invariant Prior

- Assuming equal energy illumination：

$$E(\lambda,\mathbf{x}) = e(\lambda,\mathbf{x})\left((1-i(\mathbf{x}))^2 R_\infty(\lambda,\mathbf{x}) + i(\mathbf{x})\right), \quad (1)$$

$$E(\lambda,\mathbf{x}) = \tilde{e}(\mathbf{x})\left((1-i(\mathbf{x}))^2 R_\infty(\lambda,\mathbf{x}) + i(\mathbf{x})\right), \quad (5)$$

$$\frac{E^\lambda}{E^{\lambda\lambda}} = \frac{\tilde{e}(\mathbf{x})(1-i(\mathbf{x}))^2 R_\infty^\lambda}{\tilde{e}(\mathbf{x})(1-i(\mathbf{x}))^2 R_\infty^{\lambda\lambda}} = \frac{R_\infty^\lambda}{R_\infty^{\lambda\lambda}}, \qquad (6)$$

$$H = \arctan\left(E^\lambda / E^{\lambda\lambda}\right). \qquad (7)$$

Learnable Illumination-Invariant Prior

$$E(\lambda, \mathbf{x}) = \tilde{e}(\mathbf{x}) \left((1 - i(\mathbf{x}))^2 R_\infty(\lambda, \mathbf{x}) + i(\mathbf{x}) \right), \quad (5)$$

- Further assuming that the surface is matte, i.e. i(x) $\approx$ 0：

$$E(\lambda, \mathbf{x}) = \tilde{e}(\mathbf{x}) R_\infty(\lambda, \mathbf{x}), \quad (8)$$

$$\begin{aligned} C &= \log\left(\frac{(E^\lambda)^2 + (E^{\lambda\lambda})^2}{E(\lambda, \mathbf{x})^2}\right) \\ &= \log\left(\frac{(R_\infty^\lambda)^2 + (R_\infty^{\lambda\lambda})^2}{R_\infty(\lambda, \mathbf{x})^2}\right). \end{aligned} \quad (9)$$

- Further assuming uniform illumination：

$$E(\lambda, \mathbf{x}) = \bar{e} R_\infty(\lambda, \mathbf{x}), \quad (10)$$

$$\begin{aligned} W &= \tan\left(\left|\frac{\partial E(\lambda, \mathbf{x})}{\partial \mathbf{x}} \frac{1}{E(\lambda, \mathbf{x})}\right|\right) \\ &= \tan\left(\left|\frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \mathbf{x}} \frac{1}{R_\infty(\lambda, \mathbf{x})}\right|\right). \end{aligned} \quad (11)$$

$$O(x, y) = [O_R(x, y), O_G(x, y), O_B(x, y)], \quad (13)$$

Learnable Illumination-Invariant Prior

$$H = \arctan\left(E^{\lambda}/E^{\lambda\lambda}\right). \tag{7}$$

$$\begin{aligned} W &= \tan\left(\left|\frac{\partial E(\lambda,\mathbf{x})}{\partial \mathbf{x}}\frac{1}{E(\lambda,\mathbf{x})}\right|\right) \\ &= \tan\left(\left|\frac{\partial R_{\infty}(\lambda,\mathbf{x})}{\partial \mathbf{x}}\frac{1}{R_{\infty}(\lambda,\mathbf{x})}\right|\right). \end{aligned} \tag{11}$$

$$\begin{aligned} C &= \log\left(\frac{(E^{\lambda})^2+(E^{\lambda\lambda})^2}{E(\lambda,\mathbf{x})^2}\right) \\ &= \log\left(\frac{(R_{\infty}^{\lambda})^2+(R_{\infty}^{\lambda\lambda})^2}{R_{\infty}(\lambda,\mathbf{x})^2}\right). \end{aligned} \tag{9}$$

$$O(x,y) = \left[O_R(x,y), O_G(x,y), O_B(x,y)\right], \tag{13}$$

Method--Framework

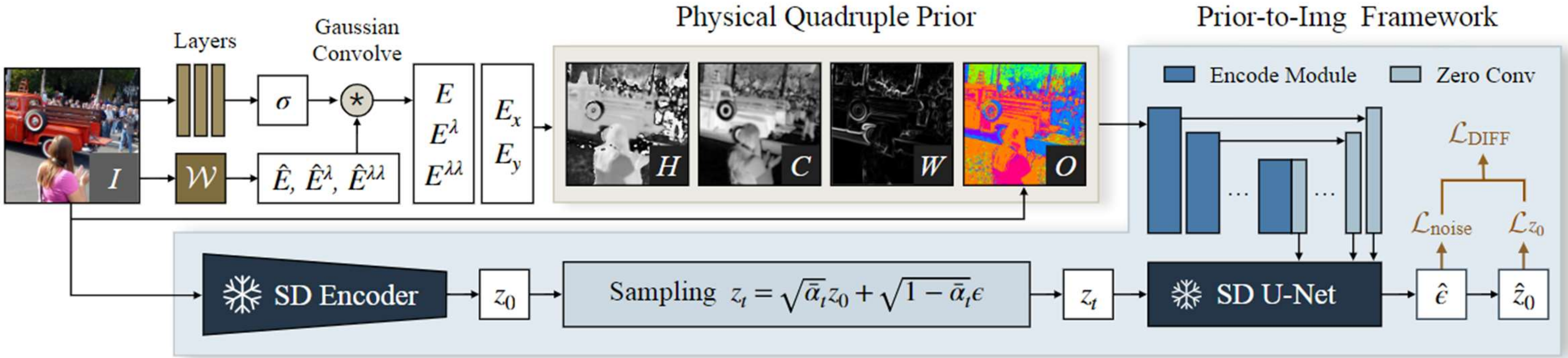


Figure 3. Our illumination-invariant prior and the training process for our prior-to-image model framework. We start by predicting the physical quadruple prior from the input image I. During the training phase, the model dynamically learns the linear mapping $\mathcal{W}$ and the layers for predicting the scale σ. In the process of reconstructing priors into images, a static SD encoder extracts the latent representation z_0 from the input image I. Following this, we sample noisy latent z_t based on z_0. Finally, the physical quadruple prior is encoded by convolutional and transformer modules, and is then merged with a frozen SD U-net to predict both noise ϵ and z_0.

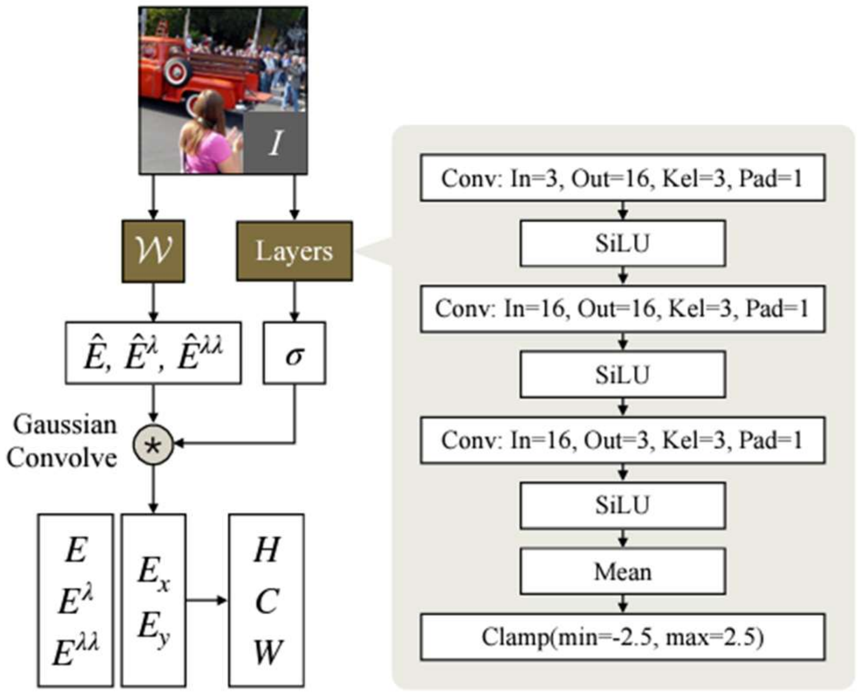
$$\begin{bmatrix} \hat{E}(x,y) \\ \hat{E}^{\lambda}(x,y) \\ \hat{E}^{\lambda\lambda}(x,y) \end{bmatrix} = \mathcal{W} \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

$$\mathcal{L}_{\text{noise}} = ||\epsilon - \hat{\epsilon}||_2^2. \tag{15}$$

$$\mathcal{L}_{z_0} = ||z_0 - \hat{z}_0||_2^2 = ||z_0 - \frac{z_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}||_2^2. \tag{16}$$

$$\mathcal{L}_{\text{DIFF}} = \mathcal{L}_{z_0} + \mathcal{L}_{\text{noise}}. \tag{17}$$

Method



(a) Detailed architecture in Physical Quadruple Prior

```python
def O(batch):
    max_RGB = torch.argmax(batch, dim=1)
    min_RGB = torch.argmin(batch, dim=1)

    batch_ = torch.flip(batch, dims=(1,))

    max_RGB_ = 2 - torch.argmax(batch_, dim=1)
    min_RGB_ = 2 - torch.argmin(batch_, dim=1)

    RGB_order = torch.zeros(batch.shape, device=batch.device, dtype=batch.dtype)
    RGB_order = RGB_order.scatter_(1, max_RGB.unsqueeze(1), 0.5, reduce='add')
    RGB_order = RGB_order.scatter_(1, max_RGB_.unsqueeze(1), 0.5, reduce='add')
    RGB_order = RGB_order.scatter_(1, min_RGB.unsqueeze(1), -0.5, reduce='add')
    RGB_order = RGB_order.scatter_(1, min_RGB_.unsqueeze(1), -0.5, reduce='add')
    return RGB_order
```

(b) Code for computing O

Figure 10. The detailed network for computing H, C, and W, as well as the PyTorch [16] code for computing O.

Method

(a) I (b) SD Decoded (c) $\tilde{I}$ (d) Ours

Figure 4. Image restoration effect of the SD decoder and ours. (a) Input image I, from which we extract latent z_0. (b) z_0 decoded by the SD decoder. (c) The distorted version of I. (d) z_0 decoded by our decoder using the encoder features from $\tilde{I}$.

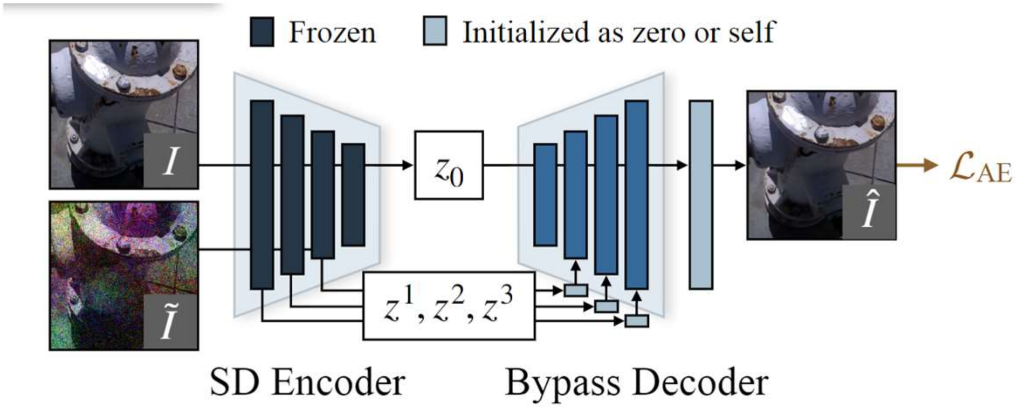


Figure 5. The training strategy of our bypass decoder. We distort the input image I into $\tilde{I}$, and allow the decoder to reconstruct I using encoder features from the distorted $\tilde{I}$.

Method

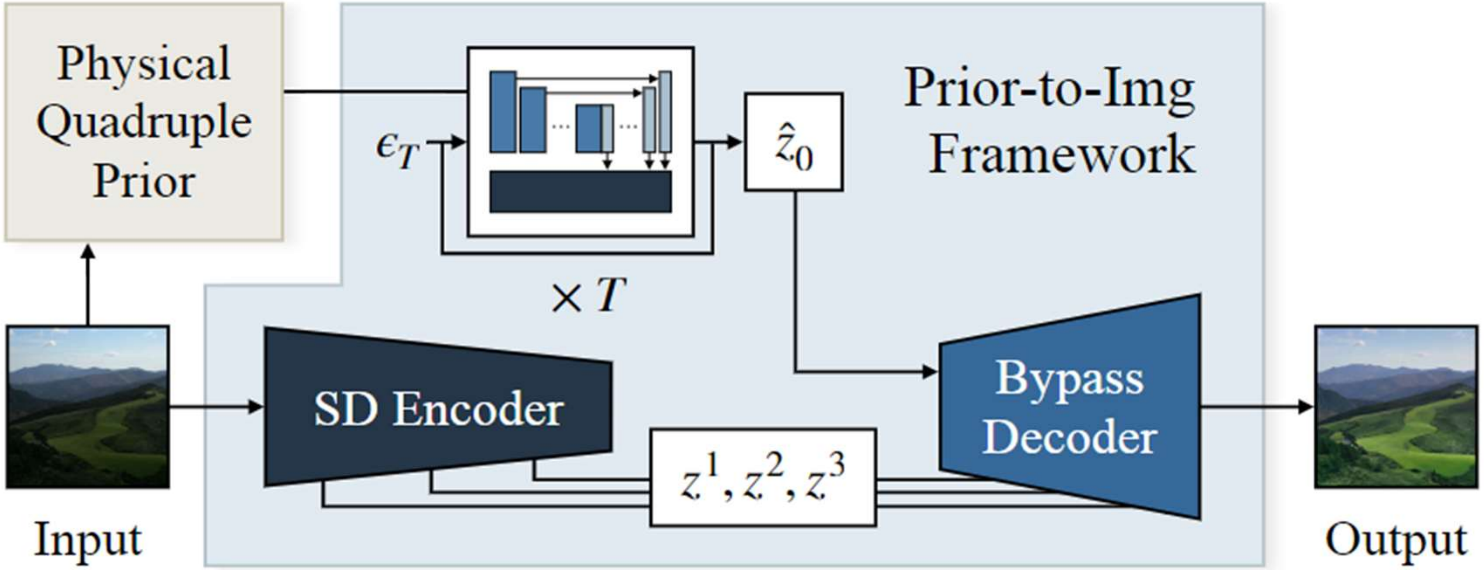


Figure 6. The inference pipeline of our overall framework. Given a low-light image, we extract its physical quadruple prior. Then, this prior serves as the condition for predicting the latent representation $\hat{z}_0$ from pure noise ϵ_T. Lastly, the bypass decoder utilizes features extracted by the encoder from the low-light image to map the predicted $\hat{z}_0$ back into images.

Experiments

Table 1. Benchmarking results for low-light enhancement. Among unsupervised methods, we highlight the top-ranking scores in red and the second in blue. Additionally, we denote the training set used by each model. "LOL+" indicates a fusion of LOL and other datasets.

Datasets		Train Set	LOL [48, 55]				MIT-Adobe FiveK [2]				Unpaired Sets	
Metrics			PSNR↑	SSIM↑	LPIPS↓	LOE↓	PSNR↑	SSIM↑	LPIPS↓	LOE↓	BRISQUE↓	NL↓
Supervised	Retinex-Net [48]	LOL	16.19	0.403	0.534	0.346	12.30	0.687	0.258	0.244	27.10	3.254
	KinD [63]	LOL	20.21	0.814	0.147	0.245	14.71	0.756	0.176	0.174	26.89	0.700
	KinD++ [64]	LOL	16.64	0.662	0.410	0.288	15.76	0.650	0.319	0.176	26.16	0.431
	URetinex-Net [49]	LOL	20.93	0.854	0.104	0.245	14.10	0.734	0.182	0.187	23.80	1.319
	Retinexformer [3]	LOL	28.48	0.877	0.117	0.256	13.87	0.692	0.222	0.224	14.77	1.064
	Retinexformer [3]	MIT	13.02	0.426	0.365	0.280	24.93	0.907	0.063	0.162	24.13	0.684
	DiffLL [18]	LOL+	28.54	0.870	0.102	0.253	15.81	0.719	0.244	0.213	14.96	0.888
Unsupervised	ExCNet [61]	test images	16.29	0.455	0.380	0.295	14.21	0.719	0.197	0.197	19.03	1.563
	EnlightenGAN [19]	own data	18.57	0.700	0.302	0.291	13.28	0.738	0.203	0.199	20.65	0.779
	PairLIE [7]	LOL+	19.70	0.774	0.235	0.278	10.55	0.642	0.273	0.225	29.84	1.471
	NeRCo [54]	LSRW [12]	19.67	0.720	0.266	0.310	17.33	0.767	0.208	0.213	22.81	0.603
	CLIP-LIT [27]	own data	14.82	0.524	0.371	0.320	17.00	0.781	0.159	0.194	23.44	1.962
	ZeroDCE [10]	own data	17.64	0.572	0.316	0.296	13.53	0.725	0.201	0.191	21.76	1.569
	ZeroDCE++ [23]	own data	17.03	0.445	0.314	0.391	12.33	0.408	0.280	0.417	19.34	1.150
	RUAS [30]	MIT	13.62	0.462	0.346	0.292	9.53	0.610	0.301	0.272	29.91	2.091
	RUAS [30]	LOL	15.47	0.490	0.305	0.330	5.15	0.373	0.669	0.399	44.70	3.312
	RUAS [30]	FACE [56]	15.05	0.456	0.371	0.292	5.00	0.366	0.685	0.398	46.21	3.633
	SCI [34]	MIT	11.67	0.395	0.361	0.286	16.29	0.795	0.143	0.165	16.73	0.853
	SCI [34]	LOL+	16.97	0.532	0.312	0.289	7.83	0.573	0.360	0.187	24.46	1.893
	SCI [34]	FACE [56]	16.80	0.543	0.322	0.297	10.95	0.684	0.272	0.205	18.33	1.335
	Ours	COCO [28]	20.31	0.808	0.202	0.281	18.51	0.785	0.163	0.188	14.64	0.423

Experiments

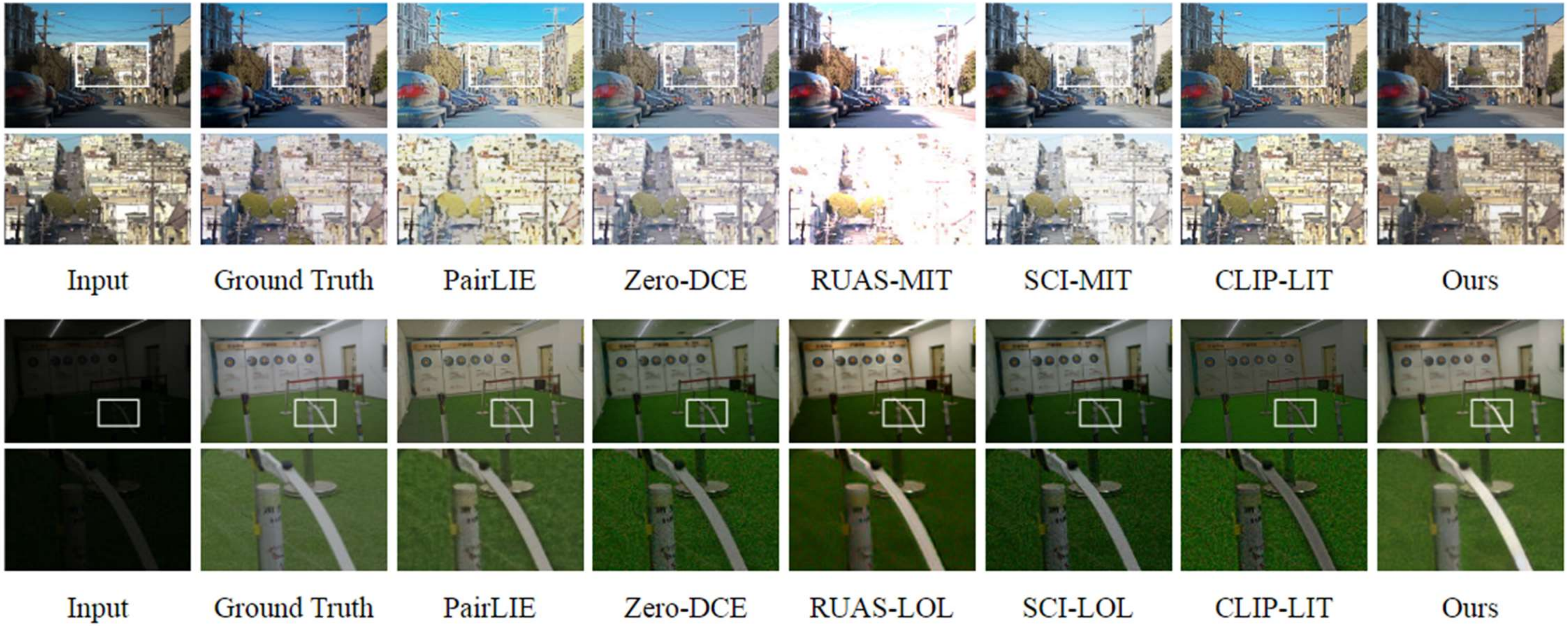


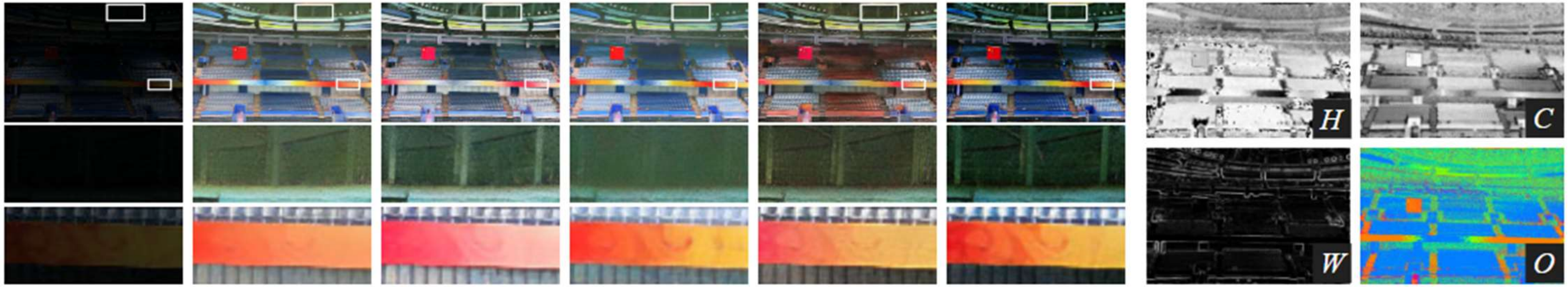
Figure 7. Example low-light enhancement results on the MIT-Adobe FiveK (top row) and LOL datasets (bottom row).

Experiments

Table 2. Ablation studies on the effect of our method designs.

Datasets		LOL [48]			
Metrics		PSNR↑	SSIM↑	LPIPS↓	LOE↓
Prior	Ours w/o H	17.60	0.756	0.262	0.314
	Ours w/o C	17.60	0.762	0.262	0.313
	Ours w/o W	17.77	0.749	0.291	0.313
	Ours w/o O	18.63	0.764	0.285	0.315
	HS channels in HSV	18.04	0.562	0.498	0.410
	CIConv	17.02	0.455	0.551	0.421
	Reflectance by PairLIE [7]	20.16	0.790	0.287	0.296
AE	SD Decoder [42]	19.26	0.665	0.243	0.353
	Consistency Decoder [1]	19.35	0.686	0.235	0.350
Ours Final Version		20.25	0.807	0.199	0.278

Experiments



Test Image w/o H w/o C w/o W w/o O Full Version Prior Visualization

Figure 8. Low-light enhancement effects for different prior designs (left), and the visualization of our physical quadruple prior (right).

Test image SR3 Our Prior-to-Img

Figure 9. Effects of using different prior-to-image frameworks.

Test Image SD Consistency Ours

Figure 10. Effects of different decoders in our framework.

Thanks!