Learnable Feature Patches and Vectors for Boosting Low-light Image Enhancement without External Knowledge–Supplementary Material

Supplementary Material

1. Implementation Details

The baselines selected for our experiments all have corresponding official code releases. We use the released code and run each model on our GPUs, following the official training scripts and settings. Additionally, we use the more challenging dynamic SDSD dataset for the SNR method, while other methods utilize the corresponding static versions. In the ablation study, the training settings are maintained as "Original" across all ablation configurations.

In different experiments, we apply loss weights to balance the contributions of the two terms in Eq. 9.

The structure of SU and MU are both a stack of convolution layers. If we adopt seven layers, the structure of SU can be summarized in Tables. 1 and 2, and the architecture of MU can be presented in Tables. 3 and 4. The corresponding graphical diagram of the model architecture is shown in Fig. 1. Moreover, the identity embedding in LFPVs is implemented by nn.Embedding in Pytorch.

The training procedure is summarized in Alg. 1.

Layer Type	Norm	Activation	Kernel	Stride	Padding	Output Size
Input Feature	-	-	-	-	-	$1 \times 1 \times 3c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times c$
Convolution	-	-	1	1	0	$1 \times 1 \times c$

Table 1. Architecture of SU with seven layers for the vector part, where "c" is the channel number.

Layer Type	Norm	Activation	Kernel	Stride	Padding	Output Size
Input Feature	-	-	-	-	-	$k \times k \times 3c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times c$
Convolution	-	-	3	1	1	$k \times k \times c$

Table 2. Architecture of SU with seven layers for the patch part, where "c" is the channel number.

2. More Ablation Studies

Comparison with other strategies that prioritize hard samples. Some previous works in other tasks, such as image classification, have also adopted techniques that prioritize hard samples, including online hard example mining (OHEM) and focal loss [1]. In this section, we conduct experiments to compare our strategy with these methods. For OHEM, we use the distance to the ground truth as a metric to measure sample difficulty. The results are pre-

Layer Type	Norm	Activation	Kernel	Stride	Padding	Output Size
Input Feature	-	-	-	-	-	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times 4c$
Convolution	BN	LeakyReLU	1	1	0	$1 \times 1 \times c$
Convolution	-	-	1	1	0	$1 \times 1 \times c$

Table 3. Architecture of MU with seven layers for the vector part, where "c" is the channel number.

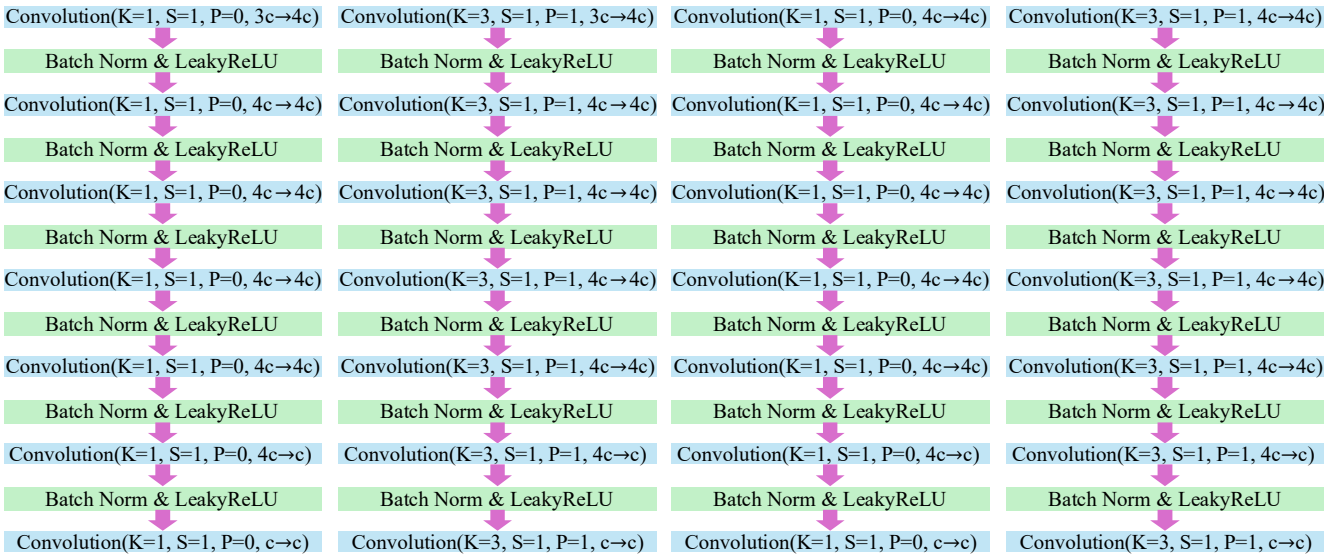
Layer Type	Norm	Activation	Kernel	Stride	Padding	Output Size
Input Feature	-	-	-	-	-	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times 4c$
Convolution	BN	LeakyReLU	3	1	1	$k \times k \times c$
Convolution	-	-	3	1	1	$k \times k \times c$

Table 4. Architecture of MU with seven layers for the patch part, where "c" is the channel number.

Algorithm 1 Our training strategy to improve the performance of the target network F with LFPVs

Parameter: Training data $(\boldsymbol{I}_d, \boldsymbol{I}_n)$, initialized F, maximum number of iteration T_{max}, number of iteration $T \leftarrow 0$

while $T \neq T_{max}$ **do**
  Read batch $\boldsymbol{D}_b = \{\boldsymbol{I}_d^1, ..., \boldsymbol{I}_d^b\}$, $\boldsymbol{Y}_b = \{\boldsymbol{I}_n^1, ..., \boldsymbol{I}_n^b\}$.
  Extract the feature of $\boldsymbol{D}_b$ as $\{\boldsymbol{f}_d^1, ..., \boldsymbol{f}_d^b\}$.
  Randomly sampling feature vectors and patches as $\boldsymbol{f}_d(x_i, y_i)$ and $\boldsymbol{f}_{p_d}(x_i, y_i)$ from $\{\boldsymbol{f}_d^1, ..., \boldsymbol{f}_d^b\}$, and update $\boldsymbol{C}_v$ and $\boldsymbol{C}_p$ with Eq. 2 and Eq. 3.
  Use the mutual-updater to update the value of $\boldsymbol{C}_v$ and $\boldsymbol{C}_p$ with Eq. 4.
  Update the original features $\{\boldsymbol{f}_d^1, ..., \boldsymbol{f}_d^b\}$ to $\{\boldsymbol{f'}_d^1, ..., \boldsymbol{f'}_d^b\}$ with Eqs. 6, 7, and 8.
  Forward $\{\boldsymbol{f}_d^1, ..., \boldsymbol{f}_d^b\}$ and $\{\boldsymbol{f'}_d^1, ..., \boldsymbol{f'}_d^b\}$ to the decoder as the output images as $\{\hat{\boldsymbol{I}}_n^1, ..., \hat{\boldsymbol{I}}_n^b\}$ and $\{\bar{\boldsymbol{I}}_n^1, ..., \bar{\boldsymbol{I}}_n^b\}$, respectively
  Compute $\mathcal{L}$ with Eq. 9 to update target network F.
  $T \leftarrow T + 1$.
end while



(a) graphical diagram of SU for the vector part (b) graphical diagram of SU for the patch part (c) graphical diagram of MU for the vector part (d) graphical diagram of MU for the patch part

Figure 1. The graphical diagrams of different models in our framework are illustrated, including SU and MU modules for both the vector and patch parts. The notation "Convolution(K=k, S=s, P=p, $i \rightarrow j$)" denotes a convolutional layer with kernel size k, stride s, padding p, and channel dimensions changing from i (input) to j (output).

Methods	SID		SMID		SDSD-Indoor		SDSD-Outdoor	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
R.M.	22.27	0.649	26.97	0.758	25.67	0.827	24.79	0.802
R.M.+OHEM	22.41	0.652	27.25	0.760	25.98	0.831	25.06	0.807
R.M.+Focal	22.63	0.657	27.58	0.764	26.19	0.835	25.70	0.801
R.M.+Ours	**23.60**	**0.664**	**28.63**	**0.775**	**27.06**	**0.846**	**26.77**	**0.814**

Table 5. Ablation study results on different datasets using various methods that prioritize hard samples. "R.M." denotes Restormer.

sented in Table 5, where "OHEM" refers to online hard example mining, and "Focal" denotes the use of focal loss. We train multiple hyperparameter configurations and report the best results. As shown in Table 5, our method consistently outperforms both baselines. This may be because OHEM and focal loss mainly focus on suppressing hard samples, whereas our approach goes a step further by leveraging LFPVs to actively assist in recovering remaining hard samples.

Cross-dataset validation. We conduct experiments to validate the generalization ability and adaptability of LFPVs. To this, we conducted cross-dataset experiments by applying LFPVs learned on SID to LOL. As shown in Table 6, we observe a degradation, since LFPVs are correlated with the ground truth features of D_n, which differs between SID and LOL. However, it's still better than the baseline, as both datasets' ground truths consist of normal-light images from diverse scenes. Thus, although LFPVs are static during inference, they demonstrate adaptability and generalization since principles shared across datasets when collecting normal-light images. We recommend employing LFPVs with datasets containing varied scenarios in practice.

Methods	SNR	+LFPVs	+LFPVs (C.D.)	+LFPVs (O.L.E.P)
PSNR	21.48	24.38	22.55	23.04
SSIM	0.849	0.864	0.852	0.857

Table 6. Ablation study results. "C.D.": cross-dataset setting; "O.L.E.P.": training with only the loss on LFPVs enhanced path.

Train with $\mathcal{L}(\bar{\boldsymbol{I}}_n, \boldsymbol{I}_n)$ only. In the original implementation, the loss function is defined as a combination of two terms: (1) $\mathcal{L}(\hat{\boldsymbol{I}}_n, \boldsymbol{I}_n)$ and (2) $\mathcal{L}(\bar{\boldsymbol{I}}_n, \boldsymbol{I}_n)$. To validate the necessity of combining both terms, we conduct an ablation study by training models using only the second loss term. The results are lower than the original LFPVs implementation but higher than the baseline (as shown in Table 6). This may be because removing the first loss term prevents full fitting of some D_f samples, making them resemble D_n and thus learned by LFPVs. As a result, LFPVs may need to learn more samples besides D_n, increasing its learning difficulty.

More ablation study results for Sec. 4.3. In Sec. 4.3 of the main paper, we have conducted ablation studies. We provide more experimental results with anther LLIE baseline in this section. The results are shown in Table 7 which also support the conclusion in the main paper.

3. Limitations

In this work, we introduce a novel strategy for formulating LFPVs, which serve as effective references to enhance LLIE methods. While this approach adds computational overhead during inference, we plan to further optimize for

Method	SID		SMID		SDSD-Indoor		SDSD-Outdoor	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
[2]	22.27	0.649	26.97	0.758	25.67	0.827	24.79	0.802
[2]+wo SU	22.88	0.655	27.18	0.760	26.04	0.830	25.52	0.791
[2]+wo MU	22.85	0.658	27.23	0.769	26.63	0.836	26.08	0.804
[2]+wo Pos	23.04	0.657	28.02	0.765	26.88	0.833	26.63	0.797
[2]+Hard	23.15	0.661	27.79	0.769	27.29	0.852	26.70	0.816
[2]+LFPV-D.	22.76	0.650	27.91	0.767	26.57	0.826	25.89	0.800
[2]+8LFPV	22.73	0.662	28.14	0.772	26.90	0.842	26.03	0.809
[2]+32LFPV	23.76	0.667	28.82	0.783	27.31	0.859	26.82	0.820
[2]+LSUMU	24.00	0.675	28.93	0.787	27.46	0.855	26.95	0.824
[2]+Patch8	23.37	0.656	28.86	0.789	27.11	0.851	26.86	0.819
[2]+Patch16	23.16	0.653	28.39	0.767	27.48	0.854	26.88	0.822
[2]+Order	24.14	0.678	28.95	0.787	27.13	0.856	26.90	0.823
[2]+LSTM	24.36	0.681	29.15	0.790	27.91	0.868	27.08	0.826
[2]+Original	23.60	0.664	28.63	0.775	27.06	0.846	26.77	0.814

Table 7. Results of ablation studies on different datasets with distinct networks. The best and second best scores are highlighted. R.M. denotes Restormer.

efficiency. Additionally, our LFPVs currently rely solely on the information of datasets, without incorporating domain knowledge. Future improvements could include automatically tuning LFPVs parameters for specific scenarios or integrating external knowledge with the aid of large models.

References

[1] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

[2] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 3