BiEvLight: Bi-level Learning of Task-Aware Event Refinement for Low-Light Image Enhancement

Supplementary Material

In this supplementary material, we provide additional details and experimental results. First, we outline the framework details of BiEvLight in Sec. 1. Next, Sec. 3 offers descriptions of the training settings. In Sec. 4, we provides more qualitative examples to illustrate the model's performance.

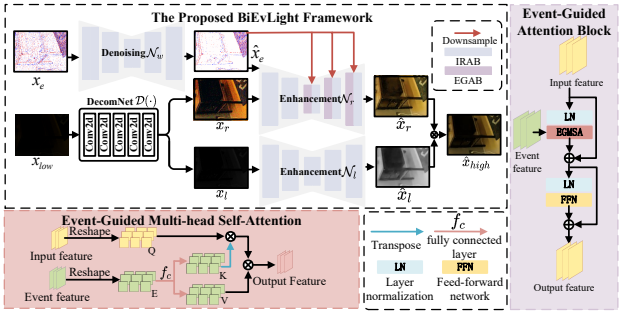

Figure 1. Qualitative analysis of denoising hyperparameters q and window size W.

1. Framework Details

The training strategy of the proposed BiEvLight has been elaborated in the main paper. Here, we provide a more detailed description of the network composition of BiEvLight and the processing of input images x_{low} and denoising event $\hat{x}_e$.

Specifically, as shown in Fig. 1, BiEvLight consists of an event denoising network $\mathcal{N}_w(\cdot)$ and an enhancement network $\mathcal{N}_\theta(\cdot)$ (comprising three subnetworks: a pre-trained decomposition network $\mathcal{D}(\cdot)$, a reflectance enhancement network $\mathcal{N}_r(\cdot)$, and an illumination enhancement network $\mathcal{N}_l(\cdot)$, where $\mathcal{D}(\cdot)$ is frozen and excluded from the training process).

Regarding the detailed network configuration, excluding the pre-trained decomposition network $\mathcal{D}(\cdot)$, the remaining subnetworks are built upon a unified 3-level encoder-decoder architecture. Specifically, in the reflectance enhancement network $\mathcal{N}_r(\cdot)$, the denoised event stream $\hat{x}_e$ is first encoded via convolution and downsampling operations to generate multi-scale features F_{ev} that align with the spatial resolution of the image features F_{img}. Subsequently, these event features are hierarchically injected into the network starting from the bottleneck layer via the Event-Image Feature Fusion Block, thereby incorporating multi-scale structure-guided information.

Specifically, the Event-Image Attention Block is designed to efficiently incorporate structural priors from the event modality into image features via a cross-attention mechanism. In the core attention computation, we project the input image features as Queries (Q) and the event features as Keys (K) and Values (V). To reduce computational complexity while preserving a global receptive field, we adopt a Transposed Attention strategy, calculating the feature cross-covariance matrix A in the channel dimension rather than the traditional spatial dimension. Furthermore, an implicit positional encoding based on depth-wise separable convolutions is introduced in parallel to supplement local spatial context, followed by a residual connection for effective feature aggregation:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_q \mathbf{F}_{img}, \quad \mathbf{K} = \mathbf{W}_k \mathbf{F}_{ev}, \quad \mathbf{V} = \mathbf{W}_v \mathbf{F}_{ev}, \\ \mathbf{A} &= \text{Softmax}\left(\alpha \cdot \bar{\mathbf{K}} \bar{\mathbf{Q}}^\top\right), \\ \mathbf{F}_{out} &= \mathbf{W}_p \left(\mathbf{A}\mathbf{V}\right) + \text{CPE}(\mathbf{V}) + \mathbf{F}_{img}, \end{aligned} \tag{1}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_p$ denote the linear projections, and $\bar{\mathbf{Q}}, \bar{\mathbf{K}}$ represent the L_2-normalized query and key features, respectively. α is a learnable scaling factor used to modulate the attention map $\mathbf{A}$, which is calculated along the channel dimension. Additionally, CPE$(\cdot)$ refers to the conditional positional encoding implemented by depth-wise separable convolutions to capture local spatial context.

Given the input image x_{low} and the denoised events $\hat{x}_e$, we first employ shallow projection layers to process the distinct modal information independently. Subsequently, the shallow features from both modalities are fused and propagated forward. The merged features pass through three encoder stages, where they undergo interactive fusion with the aforementioned event features, followed by three decoder stages and an output projection layer to generate the enhanced reflectance $\hat{x}_r$. Finally, based on Retinex theory , the enhanced image is obtained as $\hat{x}_{high} = \hat{x}_r \odot \hat{x}_l$.

2. Complexity and hyperparameters Analysis

As shown in Tab. 2, BiEvLight introduces no extra inference latency. Despite a 23% rise in training time over the alternating strategy, it yields a notable **0.64 dB** gain in PSNR*. Notably, the proposed framework is compatible with single-loop bilevel algorithms, which can be leveraged for scaling to larger networks.

Similar to common practice, the **step size** in BiEvLight follows a cosine annealing schedule with restarts. It is initialized at **2e-4** and decays to **1e-6** over **150k** iterations. We observed no instability with this setup. Regard-

Table 1. Quantitative comparison with advanced methods of low-light image enhancement on SDE and SDSD dataset. The red and blue represent the best and second-best values.

Datasets		Image Only						Image + Event			BiEvLight
		SNRNet	FourLLIE	Reformer	NeRCo	SCI++	URWKV	eSLNet	ELIE	EvLight	
Reference		*CVPR' 22*	*ACM MM'23*	*ICCV' 23*	*ICCV' 23*	*TPAMI'25*	*CVPR' 25*	*ECCV' 20*	*TMM' 23*	*CVPR' 24*	
SDE-in	PSNR↑	20.3482	19.7856	21.3169	19.4266	12.4683	21.5831	21.3622	22.0414	**22.1880**	**22.8680**
	PSNR*↑	23.6785	23.2709	23.6502	23.2052	22.3026	22.6191	23.0825	23.6395	**23.6940**	**26.0023**
	SSIM↑	0.6267	0.6176	0.6873	0.5626	0.4928	0.7168	0.7023	0.7083	**0.7189**	**0.7750**
SDE-out	PSNR↑	21.5542	20.6323	22.3075	18.4125	12.0277	22.3732	20.6244	22.2167	**22.4372**	**24.3599**
	PSNR*↑	23.3224	22.1990	23.9459	22.2225	23.2990	24.0964	22.3612	23.6772	**24.4223**	**26.1617**
	SSIM↑	0.6508	0.6410	0.6981	0.5490	0.3033	0.6880	0.6295	0.7002	**0.7070**	**0.7451**
SDSD-in	PSNR↑	25.9892	24.2775	27.7175	19.9693	9.9750	28.7034	25.8946	28.3385	**29.3563**	**30.7576**
	PSNR*↑	26.4070	28.0437	28.2613	23.4650	29.3317	28.9094	26.3361	29.6009	**30.4038**	**30.9992**
	SSIM↑	0.8730	0.8668	0.9125	0.7821	0.3108	0.9158	0.8961	0.9187	**0.9250**	**0.9473**
SDSD-out	PSNR↑	23.6656	23.4002	26.3934	21.7971	18.9373	22.0251	23.3921	25.0251	**26.7407**	**27.4108**
	PSNR*↑	27.0967	27.7418	28.6221	27.6637	28.8727	27.2018	26.3331	29.1202	**30.3066**	**30.4579**
	SSIM↑	0.8310	0.7944	0.8485	0.7482	0.5776	0.8176	0.8031	0.8626	**0.8673**	**0.8873**

Table 2. Comparison of model parameters (M), FLOPs (G) and FPS (Frames) on 256×256 images, training time (Hours), and PSNR* on SED_in Dataset.

Methods	**SSIM**	**PSNR***	**#Params**	**FLOPs**	**FPS**	**Train Cost**
URWKV	0.716	22.61	2.25	36.67	11	56
eSLENet	0.702	23.08	0.19	471.81	11	28
ELIE	0.708	23.63	220.01	201.57	4	93
EvLight	0.718	23.69	22.73	194.24	18	34
Joint	0.735	24.23	2.471	61.58	24	19
Alternating	0.769	25.36	2.471	61.59	24	39
BiEvLight	0.775	26.00	2.471	61.59	24	48

ing the finite-difference scale ϵ, it is **adaptively scaled** as $\epsilon = m/||\nabla_\theta \varphi(w_k, \theta')||_2$ to ensure numerical stability during optimization. Robustness tests (Tab. 3) show stable performance across different *m* (we used **0.01**).

Table 3. Sensitivity analysis of the scaling factor m on the SED_in dataset.

m	**SSIM**	**PSNR***
0.005	0.772	25.85
0.01	0.775	26.00
0.05	0.771	25.92
0.1	0.772	25.76

For the sensitivity of threshold q and denoising window W (Fig. 2), in BiEvLight, we set $\boldsymbol{q = 0.01}$ and $\boldsymbol{W = 5}$. As shown, high q erases details, while low q retains noise. We chose $q = 0.01$ as a trade-off. Although slight texture loss or noise may persist, this is specifically resolved by our interaction strategy, which uses feedback to recover textures and suppress noise. The second row illustrates the impact of varying W with $q = 0.01$. The denoising results remain consistent across different window sizes, demonstrating the robustness of our strategy. The results without the local window strategy (w/o W) are also displayed. Furthermore, the denoising pseudo-labels are guided by the gradient of the ground truth reflectance, thereby avoiding inaccurate gradient estimations caused by low-light degradation.

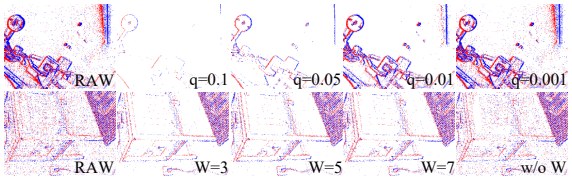

Figure 2. The network architecture of the proposed BiEvLight.

3. Implementation Detail

Our models were trained using the Adam optimizer with an initial learning rate set to 2e-4. All experiments were conducted on a single NVIDIA RTX 3090 GPU, utilizing a batch size of 8. To enhance model generalization and robustness, we applied several data augmentation techniques during training. These included random cropping to a patch size of 128 × 128 pixels, and random rotations at angles of 90, 180, and 270 degrees.

4. More Qualitative Results

In this section, we present additional visual examples for various state-of-the-art models.

Comparison on SDSD Dataset: Similarly, we conducted comparative experiments on the SDSD dataset , with quantitative results detailed in Tab. 1. BiEvLight achieved optimal performance in both indoor and outdoor scenarios. Specifically, for SDSD-in and SDSD-out tasks, our method surpasses Evlight with PSNR improvements of 1.41 dB and 0.67 dB, respectively. For the PSNR* , BiEvLight demonstrates competitive advantages with improvements of 0.60 dB for SDSD-in and 0.15 dB for SDSD-out. These results comprehensively demonstrate the effectiveness of the proposed BiEvLight method.

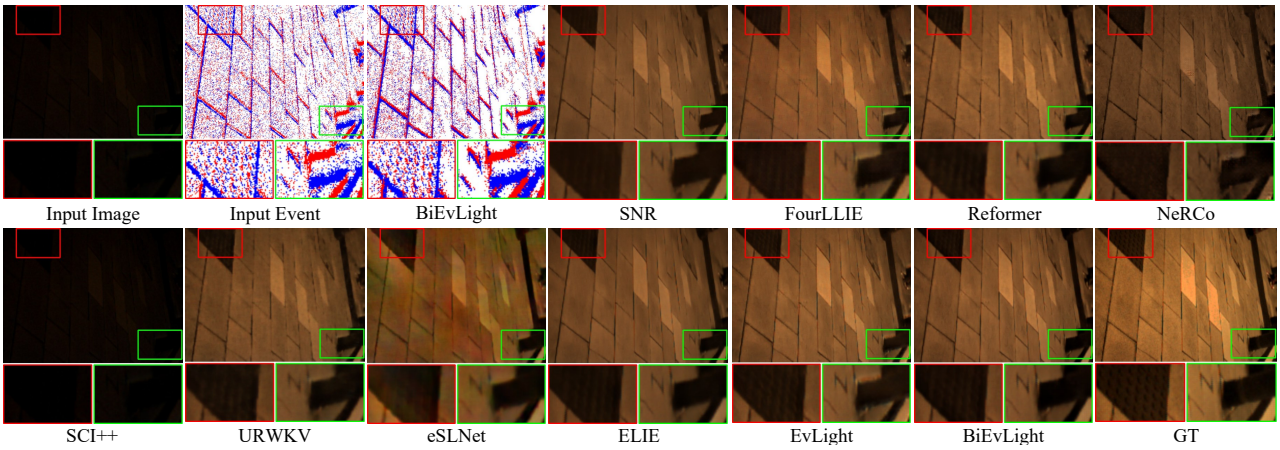

Figure 3. Qualitative results on SDE-out dataset.

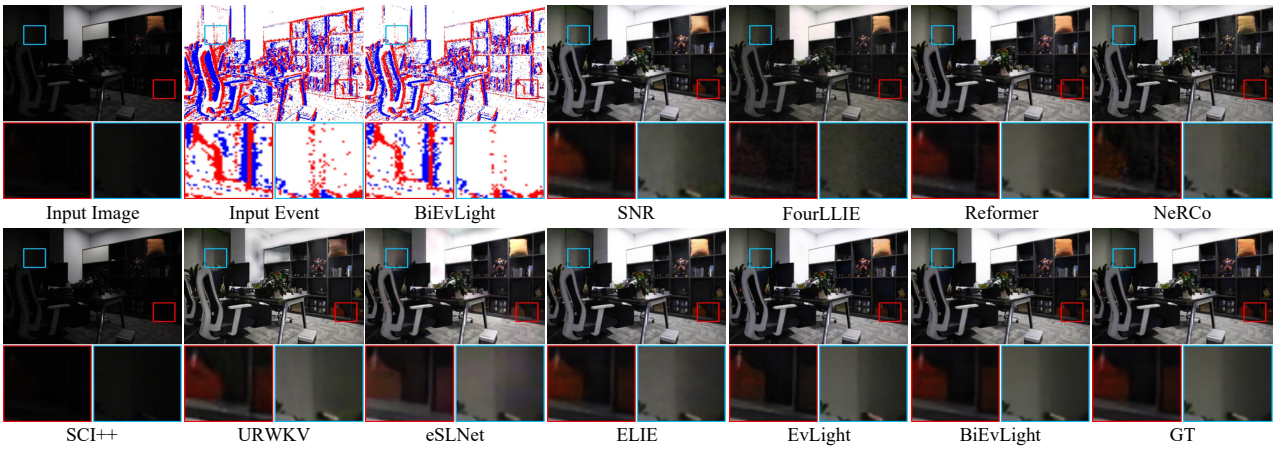

Figure 4. Qualitative results on SDSD dataset.

Qualitatively, as shown in Fig. 4, even though the event streams generated by the simulator contain only a small amount of noise, it can still degrade task performance if not addressed. As illustrated in the magnified regions, our method effectively preserves valid events while precisely removing redundant noise, and successfully recovers under-exposed image details, further highlighting its efficacy.