

One-Shot Flow, Any-Time Frame: A Bidirectional Warping Framework for Event-Based Video Frame Interpolation

Supplementary Material

Due to space constraints in the main paper, we present additional model details and experimental results in the Supplementary Material. The organization of the supplementary material is as follows:

Sec. 1 will present the reproduction details of our method, including the structure of some models and training specifics.

Sec. 2 will further distinguish our any-time frame interpolation method from multi-frame interpolation approaches (TLX-Net and TimeTracker) and analyze the advantages of our method.

Sec. 3 compares our approach with SOTA flow-free (Synthesis) methods to further validate the effectiveness of our proposed method.

Sec. 4 provides a demonstrative video, which includes a complete comparison of video frame interpolation results achieved by our method against other SOTA methods.

Sec. 5 addresses the inherent limitations and defects within our proposed methodology.

Sec. 6 provides additional visualization results comparing our method with other SOTA methods across various datasets,

1. Implementation Details

1.1. Backward Flow Estimator

The function of the Backward Flow Estimator (BFE) module is outlined within the Bidirectional Flow Estimation Block (BiFEB) description, while its detailed architecture is illustrated in Fig. 1. The BFE leverages the input context

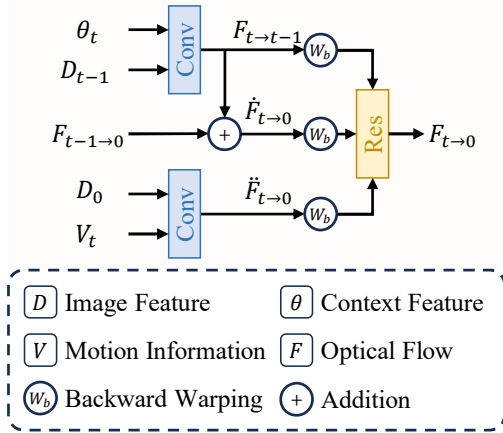


Figure 1. Detailed presentation of the Backward Flow Estimator

feature θ_t , the previous output $F_{t-1 \rightarrow 0}$, and motion information V_t to predict three distinct optical flows: $F_{t \rightarrow t-1}$, $\dot{F}_{t \rightarrow 0}$, and $\ddot{F}_{t \rightarrow 0}$. Subsequently, the corresponding feature D_t is processed via backward warping. Finally, this warped result, along with D_0 , is fed into the Residual layer to yield the ultimate flow representation $F_{t \rightarrow 0}$.

1.2. Model Details

In our model, all feature representations possess 128 channels and are maintained at $1/4$ scale resolution. For individual convolution layers, we do not apply any activation function. Our residual layer first concatenates all input features. This concatenated tensor is then processed by a single convolution layer to generate a 128-channel main output. The secondary output is obtained using multiple convolution layers with LeakyReLU, and then added element-wise to the main element.

1.3. Train Details

During the training phase, we set the batch size to 1. However, each batch comprises 17 images (I_0 , 15 ground-truth frames, and I_1). We simultaneously interpolate all 15 intermediate frames in a single forward pass to compute the loss. We employ a composite loss function consisting of the L1 loss and the LPIPS[8] loss. To balance their magnitudes, the LPIPS term is weighted by $w = 0.1$. The Adam[2] optimizer is used with its default parameters. All models are trained on a single NVIDIA RTX 3090 GPU. The training process runs for 20 epochs, which takes approximately 80 hours.

2. Any-Time Frame Interpolation VS Multi-Frame Interpolation

Although multi-frame methods (such as TLX-Net and TimeTracker) have achieved certain results by predicting multiple optical flows simultaneously to reduce computational costs, they still fundamentally struggle to achieve any-time frame interpolation with low computational cost. We will further illustrate the differences between our approach and theirs, explaining why they face challenges in any-time frame interpolation and how our method resolves these issues.

2.1. Core Differences

As shown in Figure 1, multi-frame methods still essentially predict one optical flow for each fixed time step. This strategy fundamentally determines their inability to interpolate

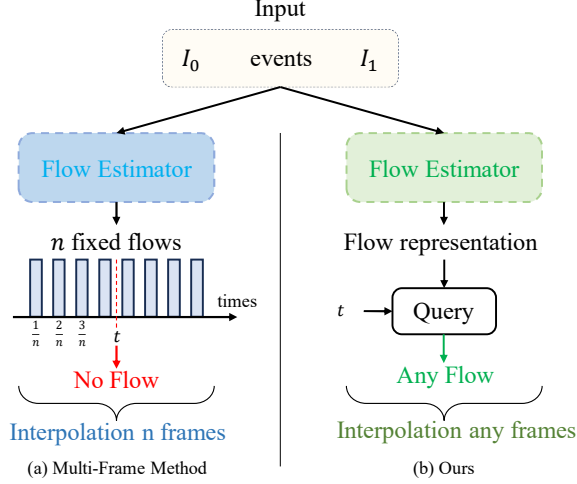


Figure 2. Core differences between any-time frame interpolation and multi-frame interpolation

frames at arbitrary timestamps. In contrast, as illustrated in Figure 2, our approach predicts the object’s motion between two frames in a single pass, then obtains optical flows for any arbitrary time through flow querying at minimal cost, thereby enabling frame interpolation at any temporal position.

2.2. Challenges in Multi-Frame Methods Achieving Any-Time Frame Interpolation

A critical challenge that multi-frame interpolation must overcome to achieve any-time frame interpolation is how to obtain the optical flow at any given time t without repeatedly executing optical flow estimation. A straightforward approach involves using blending—sampling from pre-computed optical flows at multiple fixed timestamps—to derive the optical flow at time t with low computational cost, as illustrated in Fig. 3(c). However, this introduces a new issue: since the goal of backward warping is to locate pixels in the keyframe that correspond to the target pixels, the sampled pixels may originate from other objects, leading to inaccuracies in the blended backward flows. This problem is particularly difficult for multi-frame interpolation methods to resolve, as they lack the capability to predict such problematic regions.

2.3. Our Solution

Due to the strict physical constraints of forward warping, which aims to displace objects to target positions, blending forward flows still ensures the movement of the same object, as shown in Fig. 3(b). We observe that such rigorous physical constraints enable forward warping to guide the network in rectifying erroneous regions caused by blending backward flows. However, forward warping suffers from the issue of holes, which can in turn be mitigated through

backward warping. To address this, we designed the BiW for guided refinement, as illustrated in the “Guide Refine” of Fig. 3.

Nevertheless, when dealing with nonlinear motion, traditional VFI methods can only predict linear forward optical flows, leading to inaccuracies in forward flows and consequently erroneous guidance in our bidirectional warping. To resolve this, we introduce event information. Our proposed BiFEB decomposes object motion into multiple short-term segments, leveraging event information to predict linear optical flows. Even in the presence of nonlinear motion, this strategy ensures accurate forward optical flows, thereby enabling our bidirectional warping to produce correct guidance.

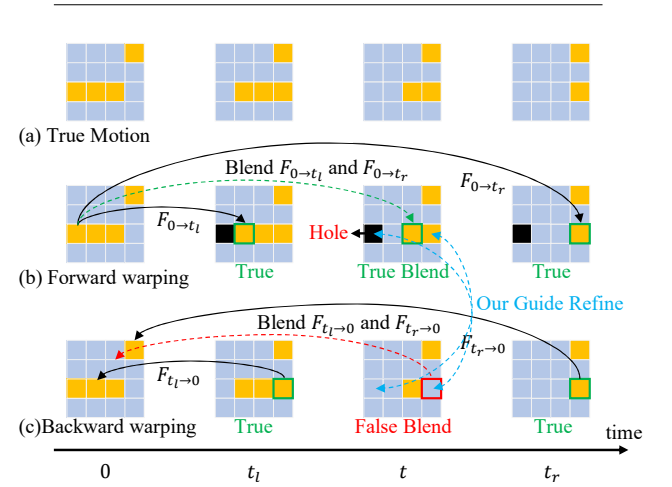


Figure 3. Drawbacks of Forward warping and Backward warping

2.4. Additional Comparison

We conducted a further comparative analysis against Multi-Frame Methods, specifically including TLX-Net and TimeTracker. It is noteworthy that we utilized a reproduced implementation of TimeTracker. However, due to the omission of several critical implementation details in their original paper, our implementation may not be perfectly accurate. This is the underlying reason why we refrained from extensively employing our reproduced TimeTracker results in the main text of this paper.

Evaluation on Real Dataset. As shown in Tab. 2 and under identical training strategies, our method comprehensively outperforms the Multi-Frame Methods on both the HSERGB and BSERGB datasets. This superior performance demonstrates the effectiveness of our strategy, which involves achieving accurate optical flow estimation via BiFEB and subsequently utilizing the BiW module to guide the network in repairing the intermediate frames.

Computational Cost Analysis. As shown in Tab. 1, our proposed method is uniquely capable of interpolating a sub-

Table 1. Analysis of computational cost on the GOPRO Dataset. The best performance in each column is marked in **bold**, and the second best is marked with underlines. The measured costs include Memory (peak GPU memory usage), MACs/f (average MACs per interpolated frame), and Time/f (average time per interpolated frame). OOM means Out of GPU Memory (>24GB)

Method	31 frames			63 frames			127 frames		
	Memory	MACs/f	Time/f	Memory	MACs/f	Time/f	Memory	MACs/f	Time/f
TLX-Net[6]	<u>11.70GB</u>	729.45G	0.079s	OOM	-	-	OOM	-	-
TimeTracker[3]	17.74GB	1852.28G	0.343s	OOM	-	-	OOM	-	-
Ours	5.29GB	<u>887.09G</u>	<u>0.137s</u>	5.95GB	738.09G	0.117s	7.27GB	665.35G	0.108s

stantial number of frames at a significantly low computational cost and with exceptional speed. Conversely, Multi-Frame Methods incur a drastically higher computational overhead due to the large volume of intermediate calculations required, as they still necessitate predicting a reverse optical flow for every individual interpolated frame.

Table 2. Regarding the evaluation on the BS-ERGB and HS-ERGB datasets. The best results are marked in **bold**, while the second are marked with underlines.

Method	BS-ERGB				HS-ERGB			
	Skip 1		Skip 3		Skip 5		Skip 7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
TLXNet[6]	29.30	<u>0.813</u>	28.72	<u>0.807</u>	-	-	31.58	0.827
TimeTracker[3]	<u>29.41</u>	0.808	<u>28.79</u>	0.803	<u>33.59</u>	<u>0.872</u>	<u>32.68</u>	<u>0.861</u>
Ours	29.76	0.823	29.03	0.815	34.63	0.889	34.19	0.881

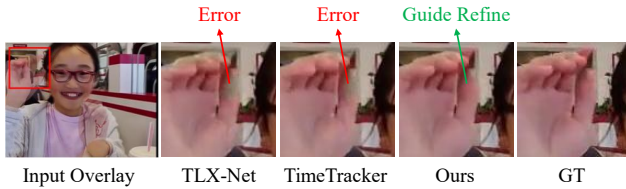


Figure 4. Visual results of our method compared with Multi-Frame methods

3. Comparison with Synthesis Methods

As the main paper only presented comparisons with flow-based methods, we now compare with SOTA synthetic methods—including VFIT-B[7], SuperFast[1], DSER[4], and EPA[5]—to further substantiate the advantages of our method.

3.1. Evaluation on Real Dataset

As shown in Tab. 3 and Fig. 5, our proposed method comprehensively outperforms synthetic approaches on both the

HS-ERGB and BS-ERGB datasets. This result further validates that our designed BiFEB and BiW modules are effective in handling various complex motions.

Table 3. Regarding the evaluation on the BS-ERGB and HS-ERGB datasets. The best results are marked in **bold**, while the second are marked with underlines.

Method	BS-ERGB				HS-ERGB			
	Skip 1		Skip 3		Skip 5		Skip 7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VFIT-B[7]	23.04	0.763	21.56	0.733	27.53	0.786	26.88	0.777
SuperFast[1]	-	-	-	-	-	-	31.18	0.862
DSER[4]	<u>29.09</u>	<u>0.810</u>	<u>28.38</u>	<u>0.802</u>	<u>34.48</u>	<u>0.883</u>	<u>34.11</u>	<u>0.879</u>
EPA[5]	27.94	0.791	27.22	0.782	33.84	0.872	33.40	0.867
Ours	29.76	0.823	29.03	0.815	34.63	0.889	34.19	0.881

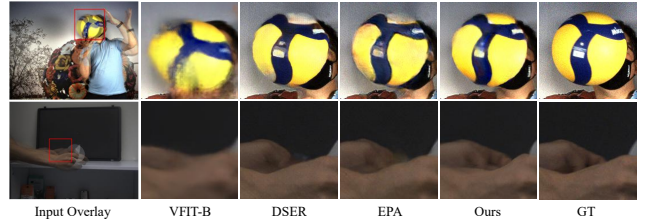


Figure 5. Visual comparison among different synthesis methods on real datasets

3.2. Computational Cost Analysis

As shown in Tab. 4, although our method does not achieve the absolute best performance in terms of GPU memory usage, it still maintains a relatively low footprint. Crucially, regarding the cost of other computational resources, our method is significantly lower than the compared SOTA synthesis methods.

4. Video Results

To more concisely illustrate the distinctions between our approach and existing methods, and thereby validate the superiority of our technique, we have produced a supplementary

Table 4. Analysis of computational cost on the GOPRO Dataset. The best performance in each column is marked in **bold**, and the second best is marked with underlines. The measured costs include Memory (peak GPU memory usage), MACs/f (average MACs per interpolated frame), and Time/f (average time per interpolated frame). OOM means Out of GPU Memory (>24GB)

Method	31 frames			63 frames			127 frames		
	Memory	MACs/f	Time/f	Memory	MACs/f	Time/f	Memory	MACs/f	Time/f
VFIT-B[7]	<u>4.49GB</u>	<u>1071.39G</u>	0.595s	<u>4.49GB</u>	<u>1071.39G</u>	0.580s	<u>4.49GB</u>	<u>1071.39G</u>	0.574s
SuperFast[1]	7.43GB	2712.19G	0.662s	7.43GB	2762.19G	0.644s	7.43GB	2762.19G	0.637s
DSER[4]	8.91GB	3459.31G	0.897s	8.91GB	3459.31G	0.903s	8.91GB	3459.31G	0.907s
EPA[5]	1.68GB	1402.88G	<u>0.306s</u>	1.68GB	1402.88G	<u>0.305s</u>	1.68GB	1402.88G	<u>0.302s</u>
Ours	5.29GB	887.09G	0.137s	5.95GB	738.09G	0.117s	7.27GB	665.35G	0.108s

video, Video.mp4. This video provides a direct visual comparison of our frame interpolation results against those of other SOTA approaches.

5. Limitations

Although our work demonstrates excellent performance on both synthetic and real datasets, it still possesses certain limitations. For instance, when confronted with objects undergoing purely linear motion, the proposed BiFEB mechanism tends to decompose this motion temporally into multiple linear segments, resulting in a redundancy of computational resources. Furthermore, given that our method fundamentally relies on event information, it struggles to achieve high-quality interpolation when keyframes lack sufficient event data. Therefore, we believe this work still has areas worthy of improvement.

6. Additional Visual Results

We further present the visualization results comparing our method with other SOTA methods across several datasets, including the GOPRO, SNU-FILM, BS-ERGB, and HS-ERGB datasets.

6.1. Visual Results on GOPRO Dataset

Fig. 6 illustrates the visualization results obtained on the GOPRO dataset, where our proposed method clearly demonstrates superior performance.

6.2. Visual Results on SNU-FILM Dataset

Fig. 7 illustrates the visualization results on the SNU-FILM dataset, where our method likewise demonstrates superior performance.

6.3. Visual Results on BS-ERGB Dataset

We also conducted extensive visualization experiments on real-world datasets to further validate the superiority of our method. The visualization results obtained on the BS-ERGB dataset are illustrates in Fig. 8.

6.4. Visual Results on HS-ERGB Dataset

Fig. 9 illustrates our visualization results on the HS-ERGB dataset, including both two close scenes and two far scenes. Our method consistently maintains the best performance across these varying depth scenarios.

References

- [1] Yue Gao, Siqu Li, Yipeng Li, Yandong Guo, and Qionghai Dai. Superfast: 200× video frame interpolation via event camera. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):7764–7780, 2022. 3, 4
- [2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1
- [3] Haoyue Liu, Jinghan Xu, Yi Chang, Hanyu Zhou, Haozhi Zhao, Lin Wang, and Luxin Yan. Timetracker: Event-based continuous point tracking for video frame interpolation with non-linear motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17649–17659, 2025. 3
- [4] Yuhao Liu, Yongjian Deng, Hao Chen, and Zhen Yang. Video frame interpolation via direct synthesis with the event-based reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8477–8487, 2024. 3, 4
- [5] Yuhao Liu, Linghui Fu, Zhen Yang, Hao Chen, Youfu Li, and Yongjian Deng. Epa: Boosting event-based video frame interpolation with perceptually aligned learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3, 4
- [6] Yongrui Ma, Shi Guo, Yutian Chen, Tianfan Xue, and Jinwei Gu. Timelens-xl: Real-time event-based video frame interpolation with large motion. In *European Conference on Computer Vision*, pages 178–194. Springer, 2024. 3
- [7] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17482–17491, 2022. 3, 4
- [8] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE*

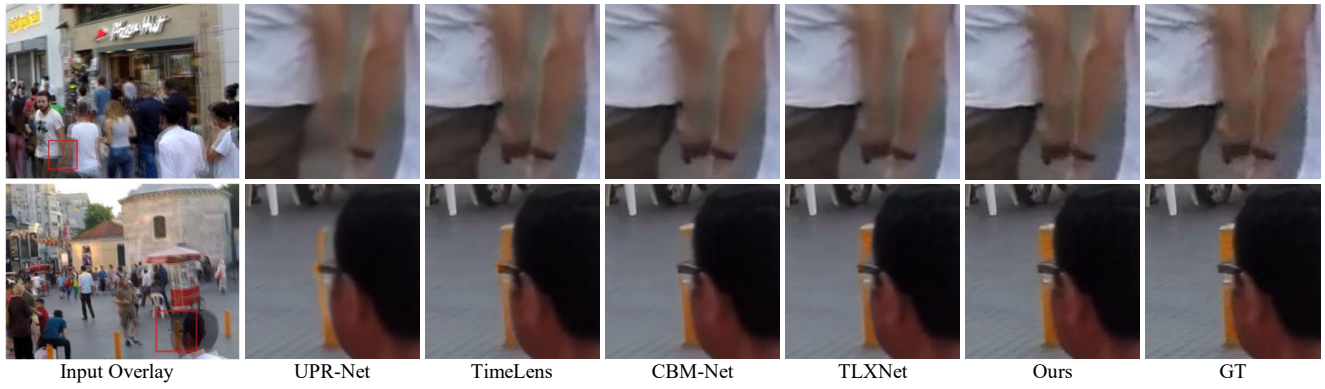


Figure 6. Additional visual comparison among different methods on GOPRO datasets.

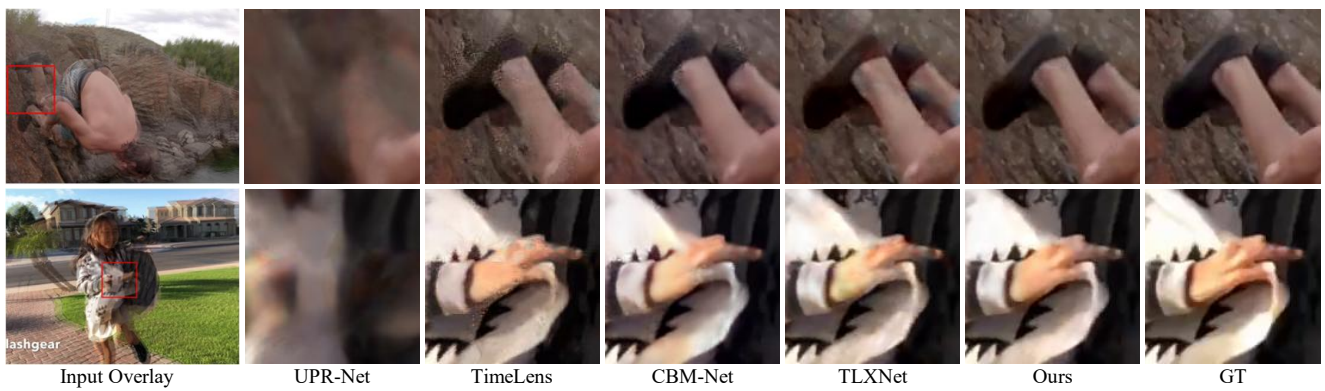


Figure 7. Additional visual comparison among different methods on SNU-FILM datasets.

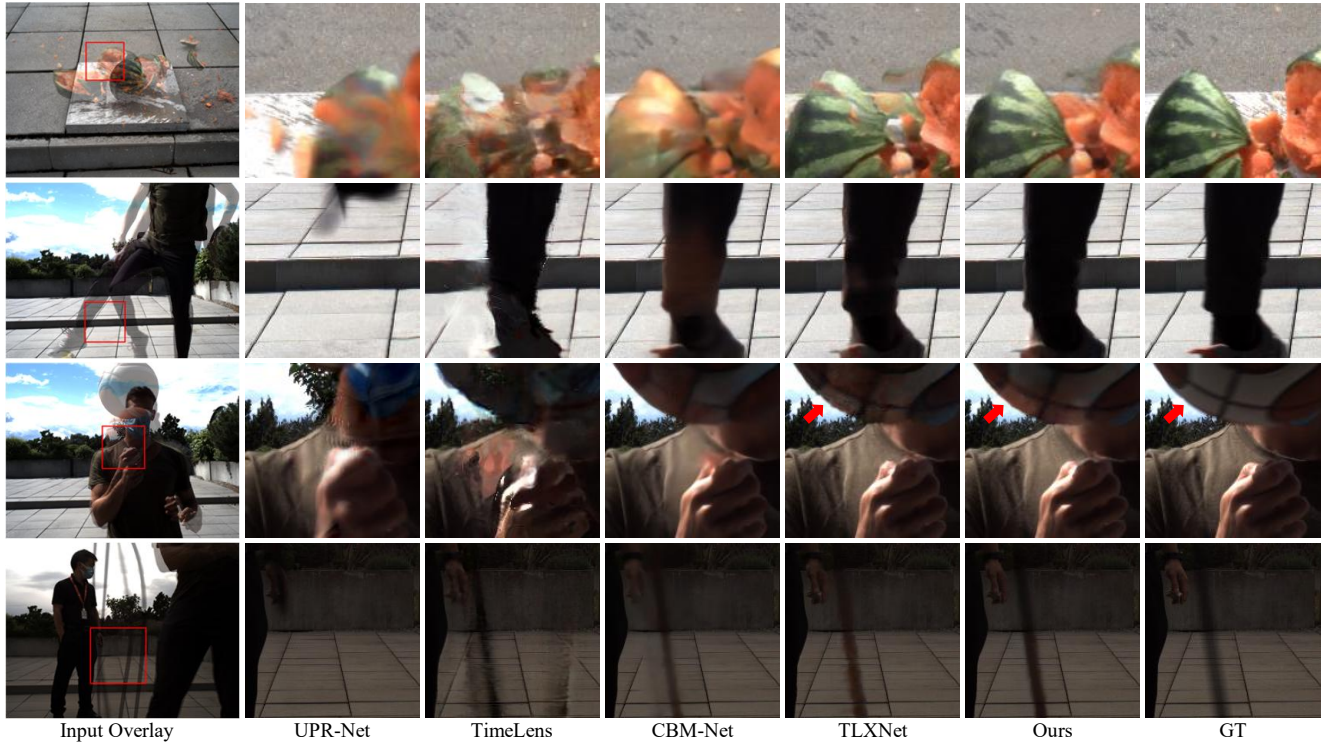


Figure 8. Additional visual comparison among different methods on BS-ERGB datasets.

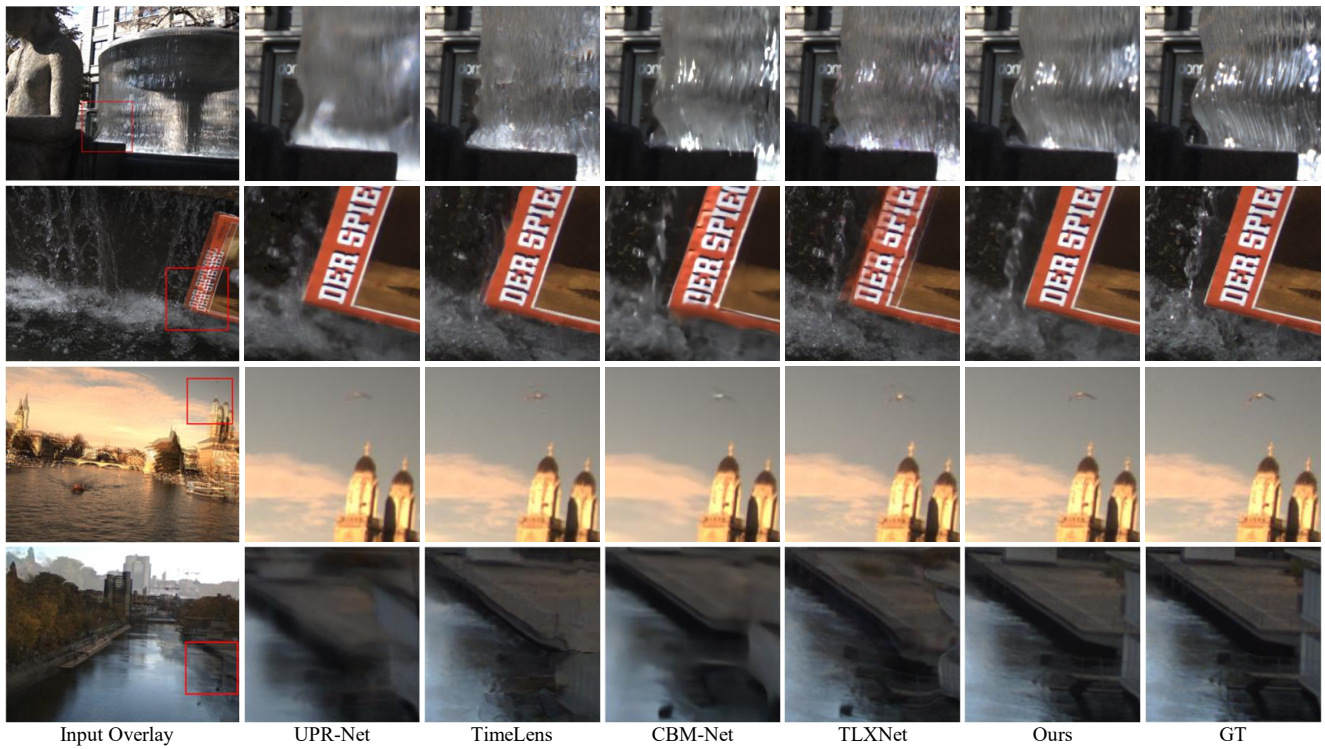


Figure 9. Additional visual comparison among different methods on HS-ERGB datasets.