Supplementary Material
TTA-EVF: Test-Time Adaptation for Event-based Video Frame Interpolation via Reliable Pixel and Sample Estimation

Hoonhee Cho, Taewoo Kim, Yuhwan Jeong, and Kuk-Jin Yoon
KAIST
{gnsgnsgml,intelpro,jeongyh98,kjyoon}@kaist.ac.kr

In this supplementary material, we provide more details of our Test-time Adaptation for Event-based Video Frame Interpolation (TTA-EVF). Specifically, we provide

- Detailed specifications for event-based video frame interpolation networks in Section 1;
- More implementation details in Section 2;
- Details about the proposed ERDS dataset in Section 3;
- Additional analysis of the proposed modules in Section 4 and Section 5;
- Additional qualitative results and video demo in Section 6 and Section 7;

1. Specifications of Event-based Video Frame Interpolation Models

Our framework is a model-agnostic approach, and to demonstrate this, we adopt recent event-based video frame interpolation models [2, 4, 7]. Instead of updating all parameters of the network, we replace some of them with the proposed norm-residual (NR) blocks, updating only those parameters. Recent networks are more complex than a simple encoder-decoder structure, and each network differs slightly in how NR blocks are integrated. We provide details on this below.

TimeReplayer [2]. As described in [2], TimeReplayer consists of three modules as flow estimation, flow refinement, and frame synthesis. Each module is designed as a U-Net [5] with skip connections, following the previous work [3]. We replaced the downsample blocks in the encoder of each U-Net with NR block. As the code for TimeReplayer is not publicly available, we reimplemented the method based on the public code of [3], inserting events to replicate the methodology as closely as possible.

TimeLens [7]. TimeLens consists of four modules: warping-based interpolation, synthesis, warping refinement, and attention-based averaging. Each module is designed as a U-Net, and we replaced only the downsample blocks of the U-Net encoder with NR Blocks. As the performance reproduction from the official code of Timelens was not achieved accurately, we opted to reimplement it from scratch to ensure that the performance closely matches the original.

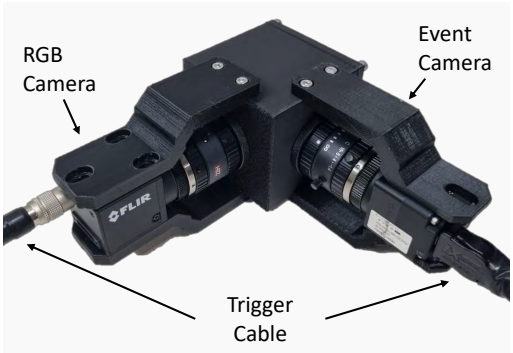

Figure 1. Camera setup for ERDS dataset.

CBMNet [4]. The CBMNet consists of two main modules: FlowNet and Synthesis. Each of these modules has its own set of encoders, and we replaced all the resblocks in these encoders with NR Blocks.

2. More Implementation Details

We implemented the NR Block based on the method described in [1], and this block is effective in terms of inference time despite having a high number of parameters due to its low computational complexity. In Section 3.3.1 of the main paper, it is noted that the performance improves with random sampling of t_0 and t_1 due to augmentation. However, to avoid CPU overload in the dataloader, we set t_0 and t_1 to 0.5 and 1.5, respectively, for all experiments, pre-processing the event voxel data for training, instead of generating new voxels each time.

3. Event-RGB Distribution Shift Dataset

Event cameras need to adapt their bias settings manually or automatically, depending on the surrounding lighting conditions, to acquire suitable data. For instance, in low-light en-

vironments, lowering the bias is necessary to increase sensitivity and capture sufficient events, while in high-light environments, increasing the bias is essential to capture critical information such as strucutres. From this perspective, we introduce a new dataset called the Event-RGB Distribution Shift (ERDS) dataset, where we adapt camera parameters for RGB and event cameras based on the surrounding lighting conditions, continuously shifting the distribution to acquire data. Test-time adaptation ultimately needs to be robust in the face of continuously changing distributions. Therefore, we believe that our dataset is highly suitable for TTA settings of event-based VFI.

3.1. Camera Configuration

To capture a high-resolution, high dynamic range scene even in the presence of continuous lighting changes, we use a hybrid sensor that combines separate RGB and event sensors using a beam splitter, rather than relying on a DAVIS sensor. As shown in Fig. 1, we adopt FLIR Blackfly S 1440×1080 RGB camera and a Prophesee IMX636 (HD) 1280×720 event camera. Despite achieving geometric alignment using a beam splitter to align the two cameras, there still exists a baseline of less than 1 mm. To address this, we employ a homography matrix to perform additional geometric alignment. Both cameras are linked to the microcontroller through a trigger cable, and the signals generated by the micro-controller are simultaneously transmitted to both the event and RGB cameras. Each camera detects the falling and rising edges of the trigger signals and synchronizes their operations accordingly. Through this external trigger, we can precisely control the frame rate and exposure time of the RGB camera using synchronized signals.

3.2. Camera Parameter Settings

Gain of RGB camera. If the lighting conditions are too low or too high, the data provided by the RGB camera may be insufficient for video frame interpolation. Therefore, we adjust the Gain value of the Blackfly RGB camera, which amplifies the pixel values, to modify brightness of the image and ensure that adequate information is captured. Please refer to the following for more details on the gain value: http://softwareservices.flir.com/BFS-U3-123S6/latest/Model/public/AnalogControl.html

On and Off biases of event camera. We manage the contrast sensitivity threshold biases of events by modifying the values of 'bias_diff_on' and 'bias_diff_off'. 'bias_diff_on' and 'bias_diff_off' adjust the contrast threshold for positive and negative events, respectively. Both having lower values make it more sensitive, resulting in more events, while higher values have the opposite effect. For a detailed explanation of this parameter, please refer to the following: https://docs.prophesee.ai/stable/hw/manuals/biases.html

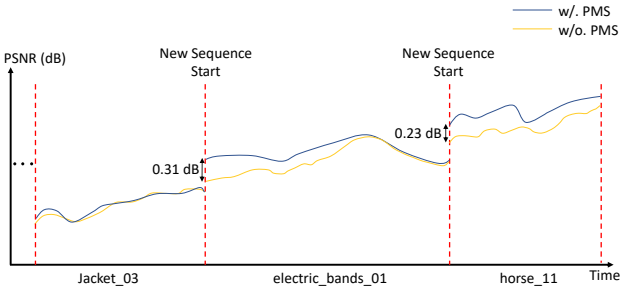

Figure 2. Visualization of PSNR over time during test-time adaptation (from HighREV [6] to BS-ERGB [8]) with and without the Patch-Mixed Sampling (PMS). The time axis is labeled with the names of each sequence. The red dotted line indicates the point where a new sequence is fed into the network.

3.3. Dataset Details

We captured data from a total of 7 sequences, depending on the environment and settings. Among these, we utilized the sequence acquired in the typical lighting conditions with the largest amount of data as the source dataset for pre-training. The remaining sequences were used for evaluation of test-time adaptation. Each sequence contains various scenes with diverse motion and objects. When capturing the scenes in the same sequence, the camera and the surrounding environmental settings were kept consistent. Details about the sequences are provided in Table 1, while specifics about each scene within the sequences are presented in Table 2. Samples from each sequence are provided in Fig. 4.

4. Analysis of Patch-Mixed Sampling

Figure 2 illustrates the PSNR values during the test-time adaptation process to verify the effectiveness of PMS. In VFI, online test-time adaptation may lead to issues like overfitting since the network continues to receive data from the same scene until a new sequence arrives. Therefore, the knowledge previously well-learned from the source dataset may fade, resulting in sub-optimal performance. However, observing the 'electric_bands_01' sequence in Fig. 2, the proposed PMS allows for flexible learning as data from multiple scenes blended during training, leading to a performance improvement of 0.31 dB at the initial state. If a sequence is long enough, networks with and without PMS can achieve relatively similar performance over time. However, in cases where the sequence length is short, such as 'horse_11', the performance difference throughout the sequence persists, highlighting the pronounced effect of PMS.

5. Visual Exhibitions for the Reliable Pixels

We present the reliable map, R_{t_0} generated during the Reliable Pixel Sampling (RPS) process in Fig. 3. Figure 3

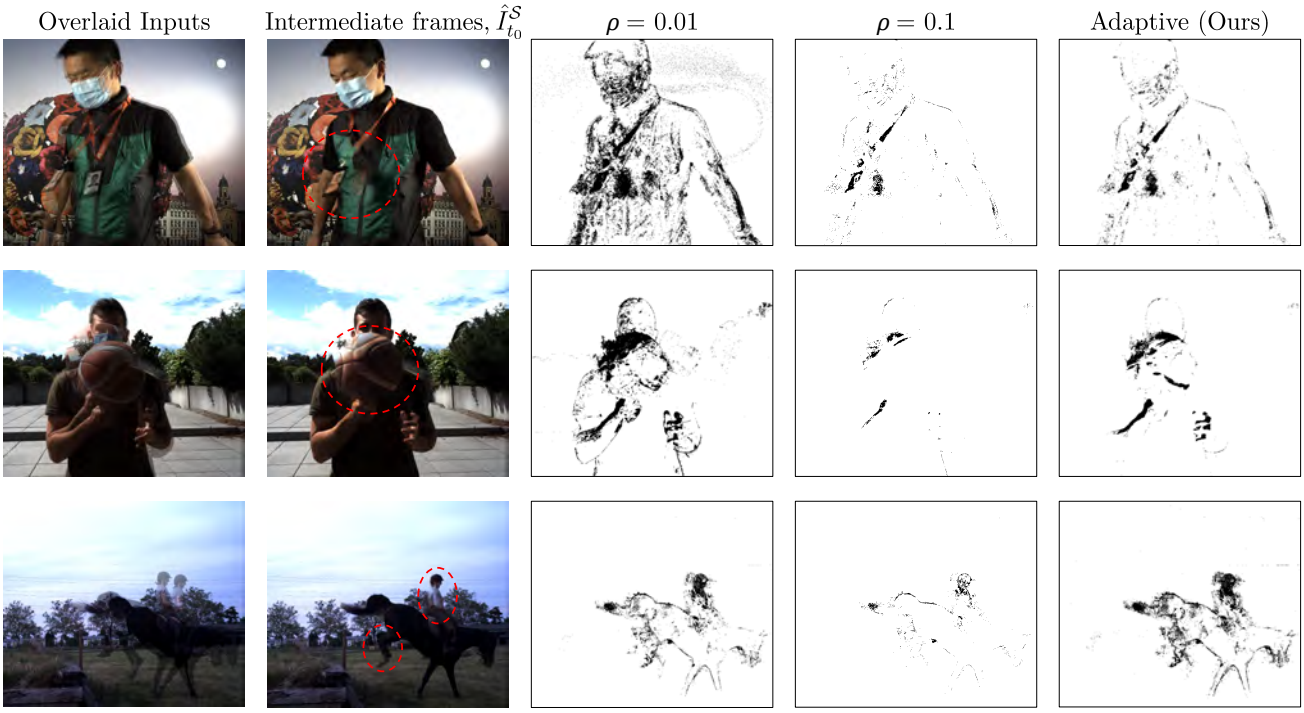

Figure 3. Visualization of the reliable map, R_{t_0}, on the cross-domain experiments. We visualize both the fixed threshold, ρ, approach and our adaptive threshold approach. The black points indicate unreliability, while the white points indicate reliability. $\hat{I}^{\mathcal{S}}_{t_0}$ is the intermediate frame generated during the self-training process, where ground truth is not available. Red dotted circles indicate areas where artifacts are predominantly present.

also includes the intermediate frame, $\hat{I}^{\mathcal{S}}_{t_0}$, generated by the student network, $\mathcal{F}_{\mathcal{S}}$. Firstly, it can be observed that the regions marked as unreliability by all threshold methods closely match the areas where artifacts are present in the intermediate generated frame, $\hat{I}^{\mathcal{S}}_{t_0}$.

However, the fixed threshold approach exhibits issues for different scenes. For example, in the 1st row, where dynamic motion is highly present, setting a small ρ, such as 0.01, labels all moving points as unreliable. Conversely, a higher ρ, as 0.1, aligns well with the actual artifact areas. In contrast, in the 3rd row, where the scene is less complex and the generated image has fewer artifacts, a small ρ aligns well with the error areas, enhancing the stability of the self-training process. On the other hand, a high ρ accumulates errors and uses error area for training. The 2nd row represents a scenario level between the 1st and 3rd rows. In contrast to these fixed threshold methods, our approach, being adaptive to the scene and particularly based on motion magnitude, demonstrates effective matching with actual artifact regions, regardless of scene complexity. Therefore, this leads to effective and stable self-training and superior performance.

6. Additional Qualitative Results

Due to space constraints in the main paper, we provide additional qualitative results in the supplementary material.

Cross-domain Datasets: Please refer to Fig. 5 and Fig. 6.
Continuous Domain Shifts: Please refer to Fig. 7.

7. Video Demos

To better demonstrate the effectiveness of our method, we provide a video demo file, '3095_supp.mp4'. Pausing at intervals for inspection allows for a more thorough examination of the improved results.

References

[1] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. 1

[2] Weihua He, Kaichao You, Zhendong Qiao, Xu Jia, Ziyang Zhang, Wenhui Wang, Huchuan Lu, Yaoyuan Wang, and Jianxing Liao. Timereplayer: Unlocking the potential of event cameras for video interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17804–17813, 2022. 1

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018. 1

[4] Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18032–18042, 2023. 1

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[6] Lei Sun, Christos Sakaridis, Jingyun Liang, Peng Sun, Jiezhang Cao, Kai Zhang, Qi Jiang, Kaiwei Wang, and Luc Van Gool. Event-based frame interpolation with ad-hoc deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18043–18052, 2023. 2, 7, 8

[7] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16155–16164, 2021. 1

[8] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. *arXiv preprint arXiv:2203.17191*, 2022. 2, 7, 8

Table 1. Overview of the proposed ERDS dataset.

Sequence Name	**No. Frames**	**Illuminance** (lux)	**Gain**	**Off Bias**	**On Bias**	**Description**
Source Sequence (for pre-train)						
General Scene	2,392	50	10	0	0	Typical indoor lighting conditions and corresponding camera settings
Target Sequences (for test-time adaptation)						
Sequence 1	1,196	50	10	-20	-20	Typical indoor lighting conditions and increased sensitivity of event cameras
Sequence 2	1,495	10	25	-20	-20	Low-light conditions and increased sensitivity of both RGB and event cameras
Sequecne 3	1,495	5	30	-35	-35	Extreme low-light conditions and increased sensitivity of both RGB and event cameras
Sequence 4	1,495	10	25	-35	-35	Extreme low-light conditions and increased sensitivity of both RGB and event cameras
Sequence 5	1,495	30	15	-35	-35	Typical indoor lighting conditions and highly increased sensitivity of event cameras
Sequence 6	897	150	10	60	60	High-light conditions and reduction in the sensitivity of both RGB and event cameras

Table 2. Details about the scene for each sequence in the ERDS dataset.

Seq. Name	Scene Name	FPS	Description
General Scene	Soccer Ball 1	75	Moving the soccer ball slowly up and down in the xy plane
	Outerwear 1	75	Shaking a flexible and non-rigid outer garment in the xy plane
	Umbrella 1	75	Spinning the umbrella around the z-axis
	Checkerboard 1	75	Moving the checkerboard dynamically in the xy plane
	Rod 1	85	Rotating the long rod-like object around the z-axis
	Checkerboard 2	75	Rotating the checkerboard slowly along the x-axis
	Checkerboard 3	108	Rotating the checkerboard rapidly along the x-axis
	Umbrella 2	84	Swinging the umbrella up and down
Target Sequences 1	Soccer Ball 2	108	Moving the soccer ball rapidly up and down in the xy plane
	Umbrella 3	108	Moving the umbrella in the z-axis direction
	Rod 2	78	Repetitively moving the rod-like object in a rowing motion in the xy plane
	Outerwear 2	75	Shaking a flexible and non-rigid outer garment in the xy plane
Target Sequences 2	Umbrella 4	75	Spinning the umbrella around the z-axis
	Checkerboard 4	75	Moving the checkerboard dynamically in the xy plane
	Soccer Ball 3	75	Moving the soccer ball rapidly up and down in the xy plane
	Rod 3	75	Rotating the long rod-like object around the z-axis
	Outerwear 3	75	Shaking a flexible and non-rigid outer garment in the xy plane
Target Sequences 3	Umbrella 5	75	Spinning the umbrella around the z-axis
	Checkerboard 5	75	Moving the checkerboard dynamically in the xy plane
	Soccer Ball 4	75	Moving the soccer ball slowly up and down in the xy plane
	Rod 4	75	Rotating the long rod-like object around the z-axis
	Outerwear 4	75	Shaking a flexible and non-rigid outer garment in the xy plane
Target Sequences 4	Umbrella 6	75	Spinning the umbrella around the z-axis
	Checkerboard 6	75	Moving the checkerboard dynamically in the xy plane
	Soccer Ball 5	75	Moving the soccer ball dynamically in the xy plane
	Rod 5	75	Repetitively moving the rod-like object in a rowing motion in the xy plane
	Outerwear 5	75	Shaking a flexible and non-rigid outer garment in the xy plane
Target Sequences 5	Soccer Ball 6	75	Moving the soccer ball rapidly up and down in the xy plane
	Umbrella 7	75	Spinning the umbrella rapidly around the z-axis
	Checkerboard 7	75	Rotating the checkerboard dynamically in the xy plane
	Rod 6	75	Rotating the long rod-like object rapidly around the z-axis
	Outerwear 6	75	Shaking a flexible and non-rigid outer garment in the xy plane
Target Sequences 6	Soccer Ball 7	75	Moving the soccer ball along the x-axis direction
	Umbrella 8	75	Spinning the umbrella around the z-axis
	Checkerboard 8	75	Dynamic movements in the xy plane while rotating the checkerboard around the z-axis

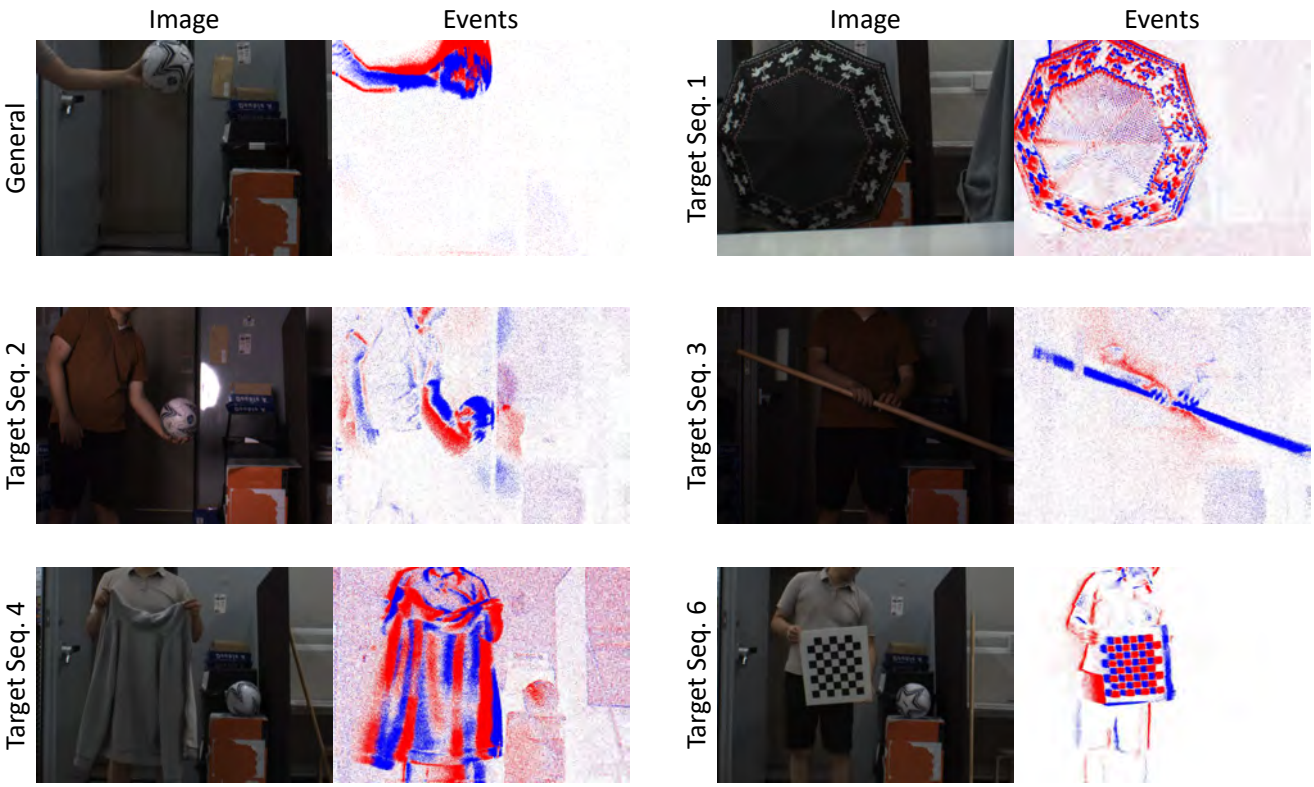

Figure 4. The samples of ERDS dataset. The ERDS dataset includes RGB and event data adjusted with light conditions. Therefore, each sequence contains data with different distributions.

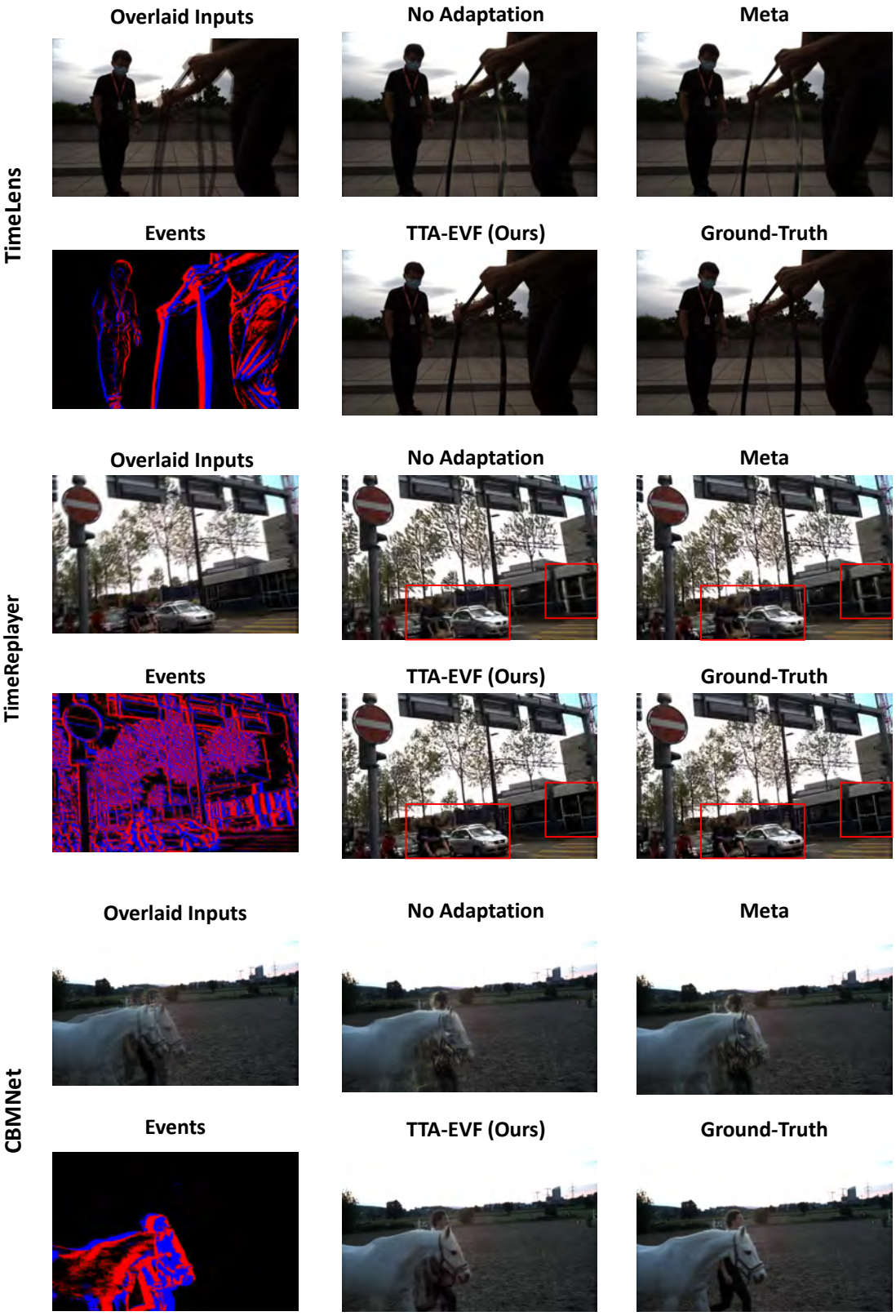

Figure 5. Visual results on the cross-domain datasets (from HighREV [6] to BS-ERGB [8]).

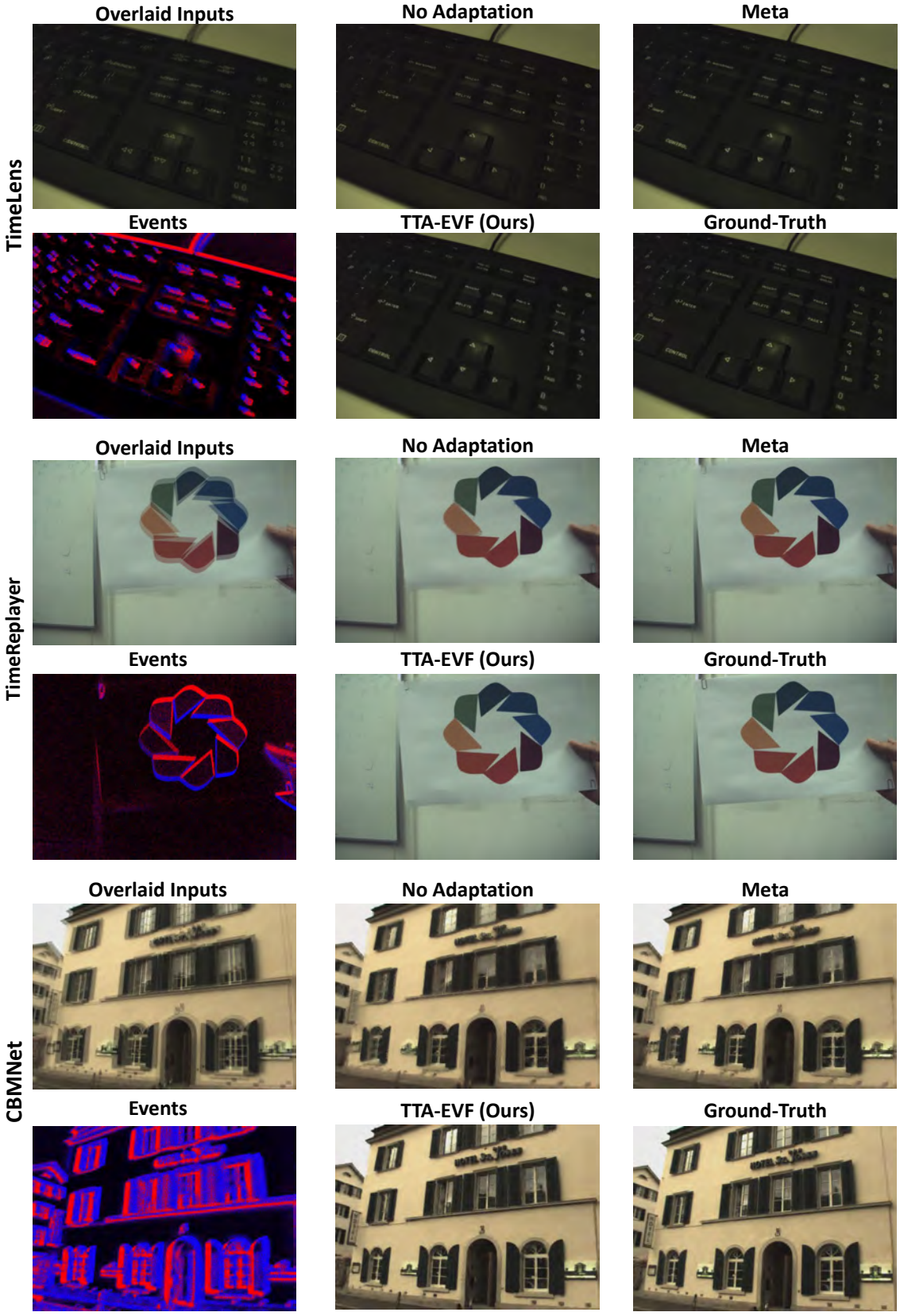

Figure 6. Visual results on the cross-domain datasets (from BS-ERGB [8]) to HighREV [6].

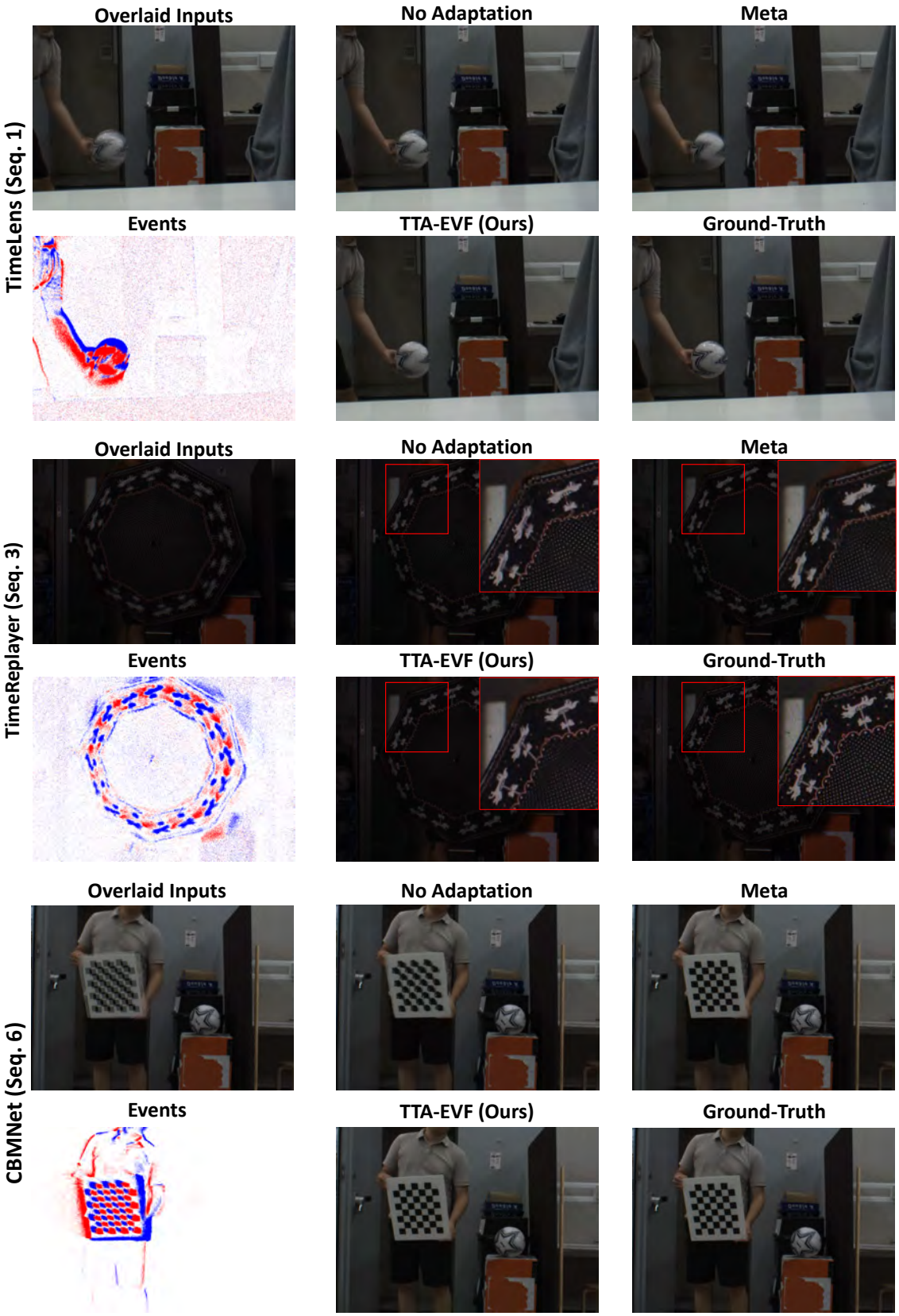

Figure 7. Visual results on the ERDS dataset with continuous domain shifting settings. For Seq. 3, the lighting was too low, so brightness of cropped image is post-processed for better visualization.