Diffusion-Based Extreme High-speed Scenes Reconstruction with the Complementary Vision Sensor —— Supplementary Material

1. Details of our network architecture

We provide a detailed breakdown of the network parameters for our two-stage cascaded diffusion model. In the first stage, a bi-directional recurrent Network performs coarse reconstruction for all intermediate frames between the two input RGB frames. The detailed network parameters are listed in Tab. 1. Due to memory constraints, the second stage applies frame-by-frame super-resolution on the low-resolution frames reconstructed in the first stage, with detailed network parameters provided in Tab. 2.

2. Details of our dataset spliting

2.1. GoPro

The original data was recorded at 240 Hz with a resolution of 1280×720, with the trainset containing 22 video clips and the testset containing 11 video clips. The transformation process maintains the original train-test split while dividing each video clip into sub-clips of 250 frames. From each sub-clip, a 1280×640 center crop is extracted, which is then further divided into four equal parts (top-left, bottom-left, top-right, bottom-right), each with a resolution of 640×320, effectively increasing the number of sub-clips by a factor of four. After this transformation, the final trainset consists of 312 video clips, while the testset contains 175 video clips.

2.2. X4K1000FPS

The original dataset is divided into X-TRAIN and X-TEST. X-TEST consists of 15 video clips, each containing 33 frames at 4K resolution and 1000 FPS. However, since our data conversion method requires at least 50 consecutive frames, we did not use X-TEST in our experiments and only use X-TRAIN as a part of our training dataset. X-TRAIN comprises 4,408 clips collected from 110 different scenes, with each clip containing 65 frames at 1000 FPS. We extract the first 50 consecutive frames from each clip and, based on our camera's resolution, split each 768×768 frame into two equal parts (each 384×768), effectively doubling the number of clips. These frames are then resized to 640×320 using bilinear interpolation. After our transformation process, the remaining 6,126 clips are used for training.

2.3. SportsSlomo

The original dataset consists of 8,498 slow-motion sports videos scraped from YouTube, with varying lengths. The raw videos are segmented into clips with frame counts that are integer multiples of 25 (ranging from 50 to 250 frames). Each segment is center-cropped to 1280×640 and then proportionally resized to 640×320. Since online videos often contain abrupt temporal transitions due to scene cuts or stitched footage, an automated script is used to remove discontinuous segments. After processing, the final dataset consists of 5,689 training clips, 189 validation clips, and 838 test clips.

3. Details of our comparison method settings

For the RGB video interpolation comparison methods XVFI [7], AMT [3], and LDMVFI [1], we retrain them using the officially provided code. The first and last RGB frames are taken from the camera-captured images obtained after DMD conversion of the RGB video, which includes motion blur caused by multi-frame integration during the exposure period. The intermediate ground truth (GT) frame is taken from the sharp frame in the original video. In the "Only Interpolation" experiment, the first and last RGB frames are also taken directly from the sharp frames in the original video, testing only the interpolation performance of the algorithms. This setup eliminates the influence of blurred RGB frames on critical processes such as optical flow estimation.

For the comparison of event-based interpolation methods, we use the official network structures and pretrained weights for real-world evaluations. For CBMNet [2], we adopt the "our_large" network architecture and the pretrained "ours_large_weight.pth" model, which was trained on the ERF-X170FPS dataset. For REFID [8], we use the official network structure along with the "REFID-GoPro-15skip.pth" weight file. (We also tested the provided deblurring + interpolation weights, "REFID-HighREV-3skip.pth," but possibly due to the smaller interpolation gap during its training, it performed worse than the former in our tests.) For TimeLens-XL [4], we utilize the "Expv8_large" model with the "Expv8_large_HQEVFI.pt" weight file. In

Table 1. Detailed architecture of our Bi-directional Recurrent model for the first stage

Stage	Layer Type	Input Ch.	Output Ch.	Attn. head dim	Additional Info
Input Layer					
Input	Conv2D	13	160	-	Kernel=3, Padding=1
Time Embedding Layer					
Time Embed	Linear	160	640	-	[160, 640], [640, 640]
Downsampling (Forward Encoder)					
Down	DownBlock2D	320	160	-	ResnetBlock * 4, Downsample (conv)
	AttnDownBlock2D	320	320	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
	AttnDownBlock2D	640	320	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
	AttnDownBlock2D	640	640	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
Downsampling (Backward Encoder)					
Down	DownBlock2D	320	160	-	ResnetBlock * 4, Downsample (conv)
	AttnDownBlock2D	320	320	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
	AttnDownBlock2D	640	320	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
	AttnDownBlock2D	640	640	32	(ResnetBlock + Attn.) * 4, Downsample (conv)
Middle Layer (Bottleneck)					
Mid	UNetMidBlock2D	640	640	32	ResnetBlock + Attn.
Upsampling (Decoder)					
Up	AttnUpBlock2D	1920	640	32	(ResnetBlock + Attn.) * 5, Upsample (conv)
	UpBlock2D	1280	320	-	ResnetBlock * 5, Upsample (conv)
	UpBlock2D	960	320	-	ResnetBlock * 5, Upsample (conv)
	UpBlock2D	640	160	-	ResnetBlock * 5, Upsample (conv)
Output Layer					
Output	Conv2D	160	3	-	Kernel=3, Padding=1

Fig. 5 in the main article, all systems' RGB pathways are configured to 30 FPS, simultaneously recording the same scene. The event-based method follows the original implementation with 16× interpolation, whereas ours employs 25×.

4. Additional experiments

4.1. Per-frame comparison results

Per-frame visualization results are presented in Fig. 1. For a fast-spinning windmill, the comparison methods exhibit blurring and significant failure in intermediate frames far from the RGB keyframes, whereas our sensor-algorithm combination can still reconstruct sharp and color-rich intermediate frames.

4.2. Diffusion on pixel spaces and two-stage generation

Due to its high computational cost, the diffusion model faces challenges in generating high-resolution images and videos. Our method employs two-stage generation in pixel space, whereas an alternative approach, proposed by [6], performs generation in a downsampled latent space. However, for detail-rich low-level vision tasks, this can lead to texture deformation and distortion [1, 9], particularly in facial features and structured grid patterns. In our experimental setup, the latent space method uses a pre-trained $\times 8$ VAE from [5] to generate in a single stage. The single-stage pixel space experiment is conducted directly at 320×640 resolution, while the two-stage pixel space experiment first generates at 48×96 resolution and then super-resolves to 320×640. None of the networks incorporates the Bi-directional Recurrent Blocks or any other temporal

Table 2. Detailed architecture of our model for the super-resolution stage

Stage	Layer Type	Input Ch.	Output Ch.	Attn. head dim	Additional Info
Input Layer					
Input	Conv2D	16	160	-	Kernel=3, Padding=1
Time Embedding Layer					
Time Embed	Linear	160	640	-	[160, 640], [640, 640]
Downsampling (Forward Encoder)					
Down	DownBlock2D	160	160	-	ResnetBlock * 2, Downsample (conv)
	DownBlock2D	160	320	-	ResnetBlock * 2, Downsample (conv)
	DownBlock2D	320	320	-	ResnetBlock * 2, Downsample (conv)
	AttnDownBlock2D	320	640	32	(ResnetBlock + Attn.) * 2, Downsample (conv)
Middle Layer (Bottleneck)					
Mid	UNetMidBlock2D	640	640	32	ResnetBlock + Attn.
Upsampling (Decoder)					
Up	AttnUpBlock2D	1280	640	32	(ResnetBlock + Attn.) * 3, Upsample (conv)
	UpBlock2D	960	320	-	ResnetBlock * 3, Upsample (conv)
	UpBlock2D	640	320	-	ResnetBlock * 3, Upsample (conv)
	UpBlock2D	480	160	-	ResnetBlock * 3, Upsample (conv)
Output Layer					
Output	Conv2D	160	3	-	Kernel=3, Padding=1

Table 3. Two-stage ablation study results on the SportsSlomo dataset.

Diffusion Space	Two Stage	PSNR↑	SSIM↑	LPIPS↓
Latent	×	21.25	0.6469	0.3670
Pixel	×	16.77	0.6687	0.3037
Pixel	✓	**25.24**	**0.8127**	**0.1959**

Table 4. Network parameters scaling results on SportsSlomo dataset.

Num resblocks	Channels	Params	PSNR↑	SSIM↑	LPIPS↓
2	[160,320,320,640]	113.63M	23.92	0.8512	0.0994
4	[160,320,320,640]	192.43M	**24.78**	**0.8715**	**0.0839**
2	[320,640,1280,1280]	641.94M	24.58	0.8651	0.0901

awareness mechanisms. Evaluation metrics are provided in Tab. 4, and visualization results are shown in Fig. 2. It can be observed that when using $\times 8$ VAE to downsample the original feature maps, the network struggles to accurately recover fine details, resulting in unnatural textures in the reconstruction results. On the other hand, when directly generating high-resolution images in a single-stage approach, limitations in network capacity and receptive field lead to color artifacts in the reconstruction. The two-stage cascaded network achieves the best performance. (Note that at this stage, the two-stage network does not incorporate any temporal awareness mechanisms, so its color restoration is not fully optimized. This setup is intended for ablation comparisons.)

4.3. Scaling capability of network parameters

We discuss the scalability of our network in terms of parameter efficiency. Our baseline employs a single-stage denoising model in a 48×96 pixel space, without any temporal awareness mechanisms. We conduct two sets of experiments: one increases the number of convolutional residual blocks, while the other expands channels of convolutional layers. The evaluation metrics are presented in Tab. 4. Evaluation results show that increasing the network's parameter count significantly improves performance across all metrics. Compared to increasing the number of channels per layer, adding more network layers proves more efficient, achieving greater performance gains with a smaller increase

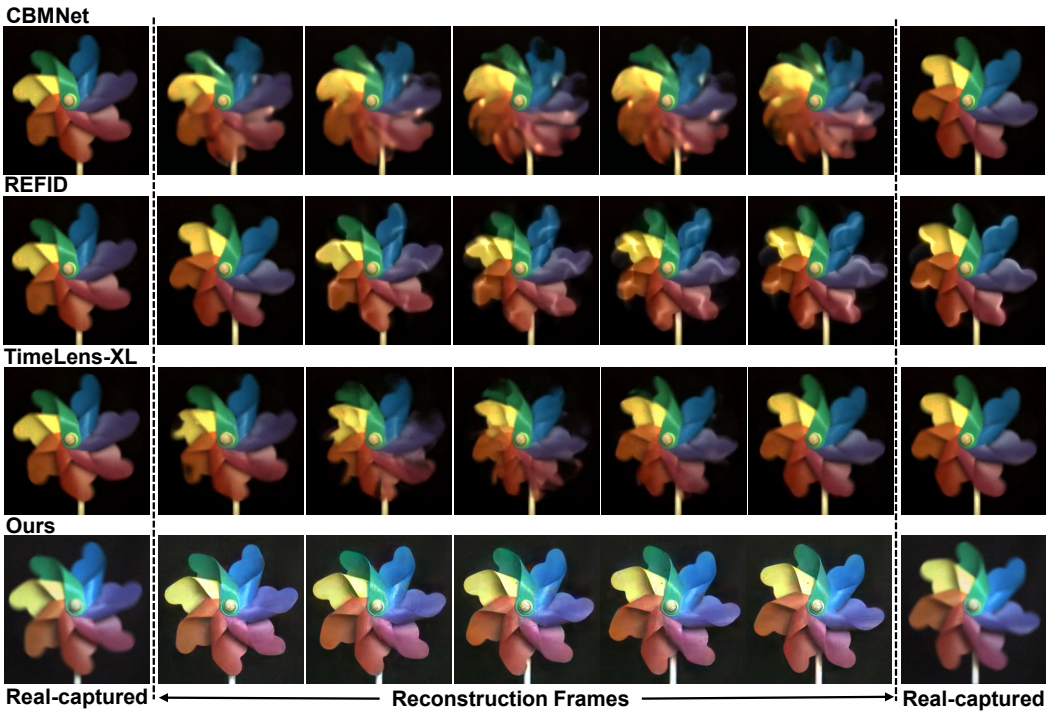

Figure 1. Per-frame reconstruction results on real-captured data compared with event-camera-based methods.

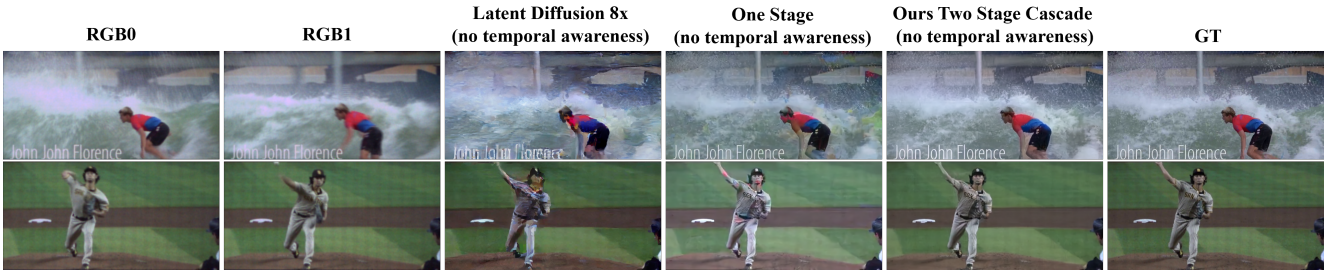

Figure 2. Visualization of the two-stage ablation experiment.

in parameters.

5. More visualization results

In Fig. 3 and Fig. 4, we provide additional frame-by-frame visualization results to better observe both our conditional input data and the reconstruction quality.

References

[1] Duolikun Danier, Fan Zhang, and David Bull. Ldmvfi: Video frame interpolation with latent diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1472–1480, 2024. 1, 2

[2] 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 (CVPR)*, pages 18032–18042, 2023. 1

[3] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023. 1

[4] 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. 1

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the*

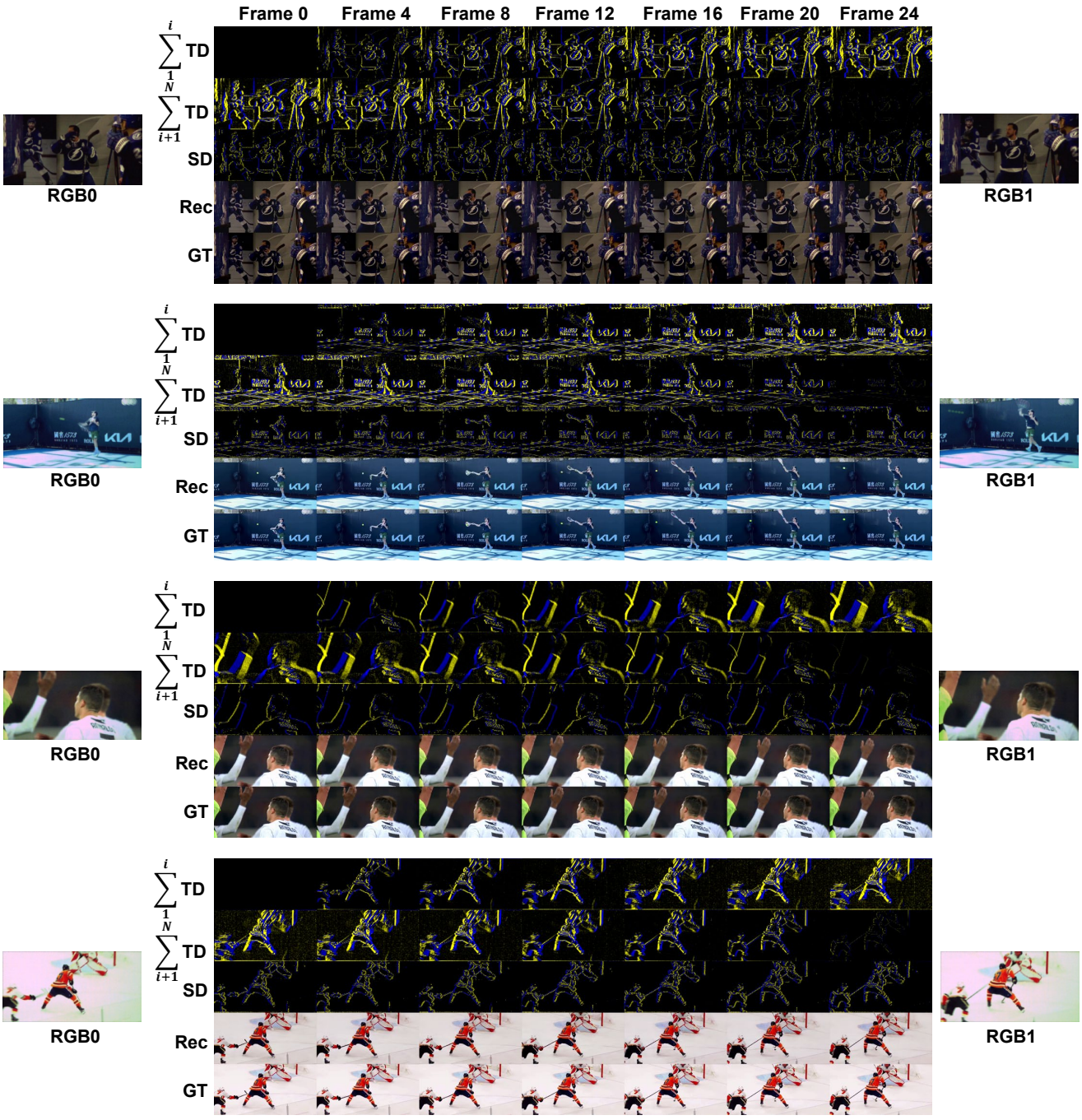

Figure 3. Frame by frame visualization results of our method, including input data and reconstruction result.

IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022. 2

[7] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14489–14498, 2021. 1

[8] 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 (CVPR)*, pages 18043–18052, 2023. 1

[9] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware

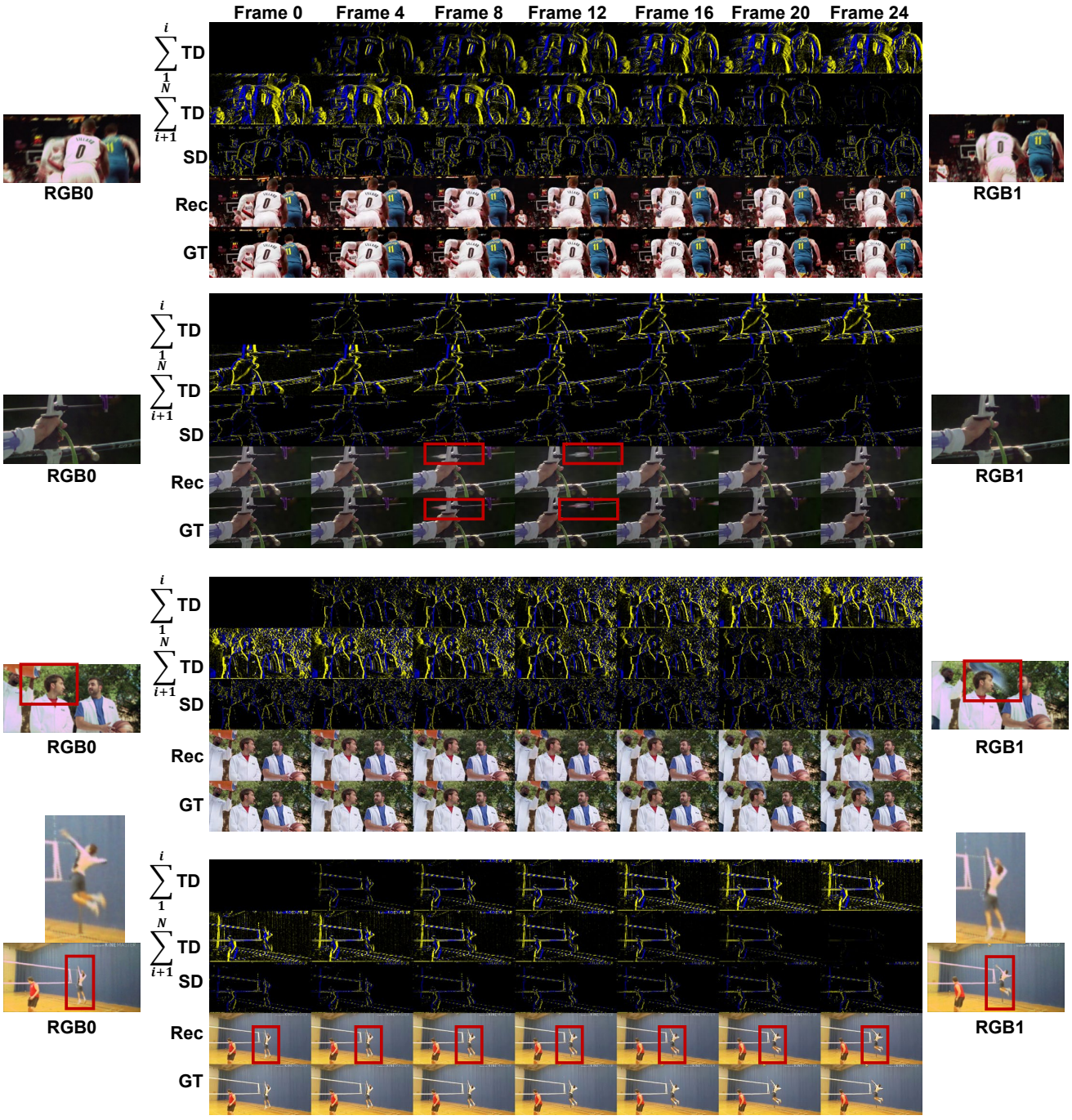

Figure 4. Frame by frame visualization results of our method, including input data and reconstruction result.

real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024. 2