Event-guided HDR Reconstruction with Diffusion Priors

Supplementary Material

Yixin Yang[1,2,†] Jiawei Zhang[3] Yang Zhang[1,2]
Yunxuan Wei[3] Dongqing Zou[4] Jimmy S. Ren[3,5] Boxin Shi[1,2,*]
[1] State Key Lab of Multimedia Info. Processing, School of Computer Science, Peking University
[2] Nat'l Eng. Research Ctr. of Visual Technology, School of Computer Science, Peking University
[3] SenseTime Research [4]PBVR [5] Hong Kong Metropolitan University
{yangyixin93, shiboxin}@pku.edu.cn, Github Page: github.com/YixinYang-00/HDRev-Diff

In the supplementary material, we provide more implementation details and comparison results. The details of different conditioning settings are described in Section 6. More details about our networks are provided in Section 7. The dataset extraction strategy of DSEC [8] is illustrated in Section 8. The training details are shown in Section 9. We provide more qualitative comparisons of ablation studies in Section 10. We show the diverse diffusion results in Section 11 to support the proposed structure loss as described in Section 3.3. We include human study to further support our results in Section 12. Efficiency comparison of existing methods is illustrated in Section 13. More results on synthetic and real data are provided in Section 14 and Section 15. Finally, we provide the consecutive results of our method to show its limited performance on video generation in Section 16.

6. Different conditioning settings

In Section 3.2, we introduce two other kinds of conditioning settings, concatenating LDR images and events, and adopting restored images as conditions, respectively. In this section, we provide more details about those two settings.

ConCond: concatenating LDR images and events. As described in Section 3.2 by Equation (5), one of the conditioning settings is directly concatenating LDR images and events as the input for embedding module $\mathcal{T}_{\textbf{ConCond}}$, which is denoted as "ConCond". The architecture of the embedding module is shown in Figure 9. The input condition channel K is set to $3 + C$, in which 3 is the number of channels of LDR image and C is the number of channels of stacked events voxel described by Equation (4). The embedding module $\mathcal{T}_{\textbf{ConCond}}$ aligns the shape of the inputs with latent space, which consists of one convolutional layer, three down-sample layers, and a final convolutional layer. Each down-sample block is composed of two convolutional layers with a kernel size of 3×3, where the stride is 1 and 2, respectively. The results of embedding module $\mathcal{T}_{\textbf{ConCond}}$ are provided to the control module $\mathcal{C}$ to control the diffusion process as described in Equation (5) and Equation (9).

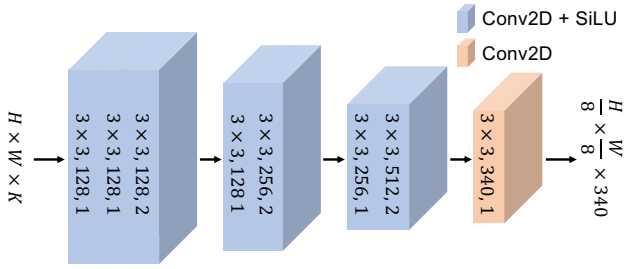

Figure 9. Architecture of embedding module $\mathcal{T}_{\text{ConCond}}$ ($K = 3+C$) and $\mathcal{T}_{\text{RestCond}}$($K = 3$). The number in each box is kernel size, output channel, and stride, respectively.

RestCond: adopting restored images As described in Section 3.2 by Equation (7), another conditioning setting is applying restored image provided by HDRev [49] as the input for embedding module $\mathcal{T}_{\textbf{RestCond}}$, denoted as "RestCond". The embedding module $\mathcal{T}_{\textbf{RestCond}}$ for "RestCond" is modified from embedding module $\mathcal{T}_{\textbf{ConCond}}$ by setting the input condition channel K to 3, as illustrated in Figure 9, in which 3 is the number of channels of the restored image. The control module $\mathcal{C}$ takes embedding results $\mathcal{E}_{\text{RestCond}}$ in Equation (7) as input to control the denoising process as illustrated by Equation (9).

7. Networks details

Event-image Encoder $\mathcal{H}$ The architecture of the event-image encoder $\mathcal{H}$ in our implementation is derived from the original implementation of HDRev [49] as shown in Figure 3. The modality-specific encoders of events and LDR images follow the original implementation, and the parameters are loaded from the released files. For the modality fusion module, we remove the handcrafted confidence map in the original implementation to avoid filtering useful information. The architecture remains the same as the original, and all the parameters are initialized from released files.

Control Module $\mathcal{C}$ The architecture of control module $\mathcal{C}$ follows the architecture of the denoiser encoder by replacing input latent z_t with the summation of z_t and condition $\mathcal{E}(I_{\text{LDR}}, E)$. To simplify training, we initialize

†This work is done during Yixin's internship at SenseTime.
*Corresponding author.

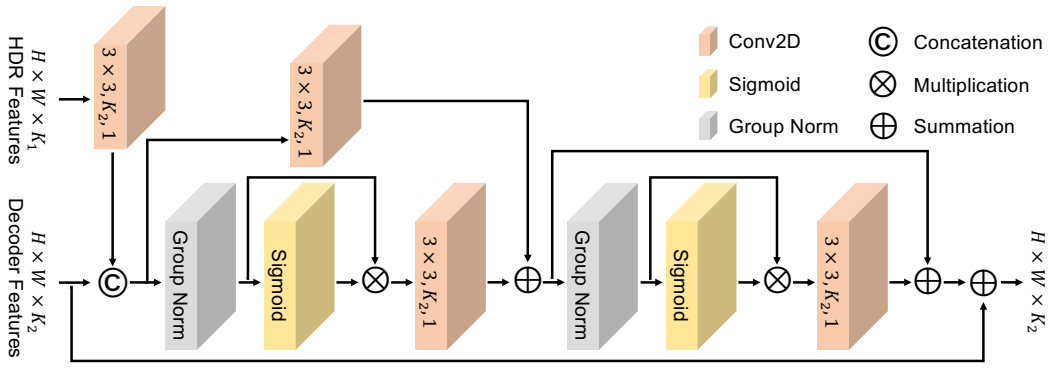

Figure 10. Architecture of feature fusion layer in refinement module $\mathcal{D}$. A fusion layer is applied before each up-sample layer in the original VAE.

the parameters of our control module C using the "control_v11e_sd15_ip2p" version of ControlNet [51].

Refinement module $\mathcal{D}$ The refinement module D decodes the estimate latent z_0 to undistorted HDR image H with the HDR features from the event-image encoder. Therefore, we modify the original VAE decoder implementation to add HDR features into its decoding process. Specifically, we add a feature fusion layer to each up-sampler layer of the VAE decoder, which fuses the HDR feature with the original VAE decode feature by the convolutional layers and residual layers as shown in Figure 10. K_1 and K_2 are the dimensions of input HDR features and decoder features from the original VAE implementation, respectively. Only the added fusion layers and convolutional output layers are trained in our experiments.

Noise scheduling For the noise scheduler, we adopt DDIM scheduler as described in Section 3.4 and set the number of training timesteps to 1000. The noise is added using a scaled linear beta schedule ranging from 0.00085 to 0.012. The model predicts pure noise (epsilon) during denoising, following the standard DDPM approachβ_1

8. DSEC [8] dataset

DESC [8] is a dataset for driving scenarios, which contains paired events and LDR images in different light conditions. We specifically choose 6 HDR scenes from its test dataset as our test dataset: "interlaken_00_a", "interlaken_00_b", "interlaken_01_a", "zurich_city_13_a", "zurich_city_13_b", "zurich_city_15_a", respectively. For "zurich_city_15_a", we only choose the 960th frame to the 1059th frame since it is not a typical HDR scene.

9. Training details

Dataset preparation We generate the synthetic dataset for training and testing as described in Section 3.4. After obtaining the generated HDR images and events, we generate LDR images from HDR images with the image formulation pipeline, which consists of exposing, dynamic range clipping, and quantization. We randomly generate exposure time t to let $x\%$ pixel to be over-/under-exposed, and x is uniformly sampled in $[0.2, 0.5]$. In dynamic range clipping, the values larger than 1 are clipped to 1. For quantization, we quantize the original float values into 8 bit integer in $[0, 255]$ and remap them to $[0, 1]$ as the final input. In this way, we obtain the LDR image L from HDR image O by:

$$L = \lfloor \text{Clip}(O \cdot t, \max = 1) * 255 \rfloor / 255 \qquad (15)$$

Histogram matching To refine the distortion existing in our diffusion results (shown by "W/o Refinement"), we adopt local histogram matching [46] to reduce the brightness and color gap between diffusion results and ground truth. To perform local histogram matching [46], we first split the whole image into $2P_h \times 2P_w$ patches, while $P_h = P_w = 8$ in our experiments. For each 2×2 patch, we calculate the adjustment parameters based on original histogram matching, which are q paired pixel values defining the brightness mapping from the original image to the destination image. We directly record the paired pixel value of the original image and destination image as the adjustment parameters. We set $q = 6$ in our experiments. To smooth the parameters of nearby patches, a convolutional layer with 5×5 Gaussian kernels is applied to the adjustment parameters with shape $P_h \times P_w$. The smoothed adjustment parameters are applied to each patch to obtain the final adjusted results. We perform local histogram matching for color images, which is implemented by separately processing each channel.

The adjusted examples are shown in Figure 11. The pur-

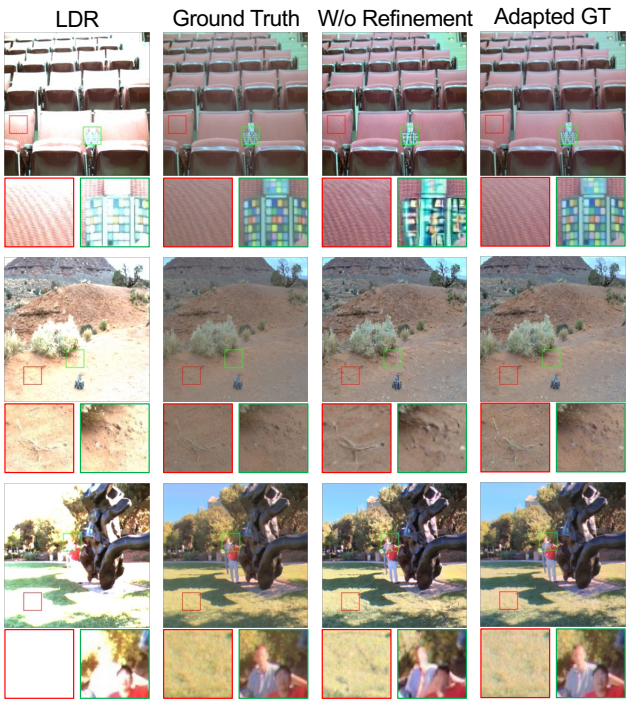

Figure 11. Adapted GT results on training data. Although our diffusion results (shown by "W/o Refinement") suffer from distortion as shown in the green box, it can serve as a good guidance to adjust the original ground truth. Adjusting the color and brightness with the histogram also improves the contrast and color as shown in the red box.

pose of our histogram matching is to reduce the color and brightness gap between the supervision target and diffusion results, which is already achieved as shown in Figure 11. Besides, the original ground truth suffers from color shifting in the first and third row, and low contrast by tone mapping in the second row. Benefiting from diffusion priors embedded in our diffusion results (shown by "W/o Refinement"), the adapted GT images have better contrast and color, which can improve the visual quality of the final refined results.

10. Ablation study

Quantitative comparison on iteration steps The impacts of different iteration steps are shown in Table 4. Balancing performance and inference speed, we finally select 9 iterations.

Quantitative comparison on loss hyperparameter The hyperparameter experiment for Equation (14) with $\alpha = 0.01, \beta = 10^{-4}$ (Param 1) and $\alpha = 0.01, \beta = 0.01$ (Param 2) are shown in Table 4.

Table 4. Quantitative comparison on iteration steps

	PSNR↑	SSIM↑	LPIPS↓	CIEDE↓	FID↓	NIQE↓
Iteration-5	24.96	0.918	0.111	6.39	29.70	3.83
Iteration-9	25.67	0.926	0.099	6.01	27.09	3.86
Iteration-15	25.89	0.928	0.096	5.90	26.43	3.88
Param 1	24.85	0.906	0.128	6.56	34.15	3.93
Param 2	24.76	0.909	0.124	6.61	32.01	3.93

Table 5. Efficiency comparison of ablation studies.

	FLOPs (G)	Params (M)	Time (s)
ConCond	8387.563	1270.09	1.33
RestCond	9207.416	1328.03	1.42
W/o Refinement	9122.417	1320.32	1.40
Ours-complete	10380.843	1349.31	1.41

Qualitative comparison on conditioning The qualitative evaluation of different conditioning and generation processes is illustrated in Figure 12. Directly concatenating events and LDR image, denoted by "ConCond", cannot well-utilize both LDR image and events to provide accurate and sufficient details as depicted by the first row and the green box of remaining rows in Figure 12. HDRev [49] fuses LDR image and events to provide better details as shown by "Restored" in Figure 12, while severe artifacts exist. And it is difficult to reconstruction faithful details in high-frequency and significantly over-exposed areas, *e.g.*, the red box of the first row in Figure 12. Employing restored results as condition, adopting restored image as condition, indicated by "RestCond", suffers from information lost exists in "Restored" as highlighted in the red box of the first row and the green box of the third row in Figure 12 Also, the artifacts in "Restored" also influence "RestCond" to provide unfaithful results as shown by the green box of the second row in Figure 12. Leveraging the HDR features provided by the event-image encoder as described in Section 3.2, the proposed method recovers faithful and colorful results by making better use of input LDR image and events with the proposed conditioning and generation method.

Qualitative comparison on structure loss The qualitative evaluation of structure loss is depicted in Figure 13. Our diffusion results (shown by "W/o Refinement") may exist distortion as shown by the green box in Figure 13. To refine the distortion and provide natural results, we perform fine-grained detail refinement with the structure loss. With the proposed structure loss, images with higher contrast and more details can be generated, as indicated by the green box of the first row, and the red box in the second and third row of Figure 13. Besides, introducing structure loss reduces

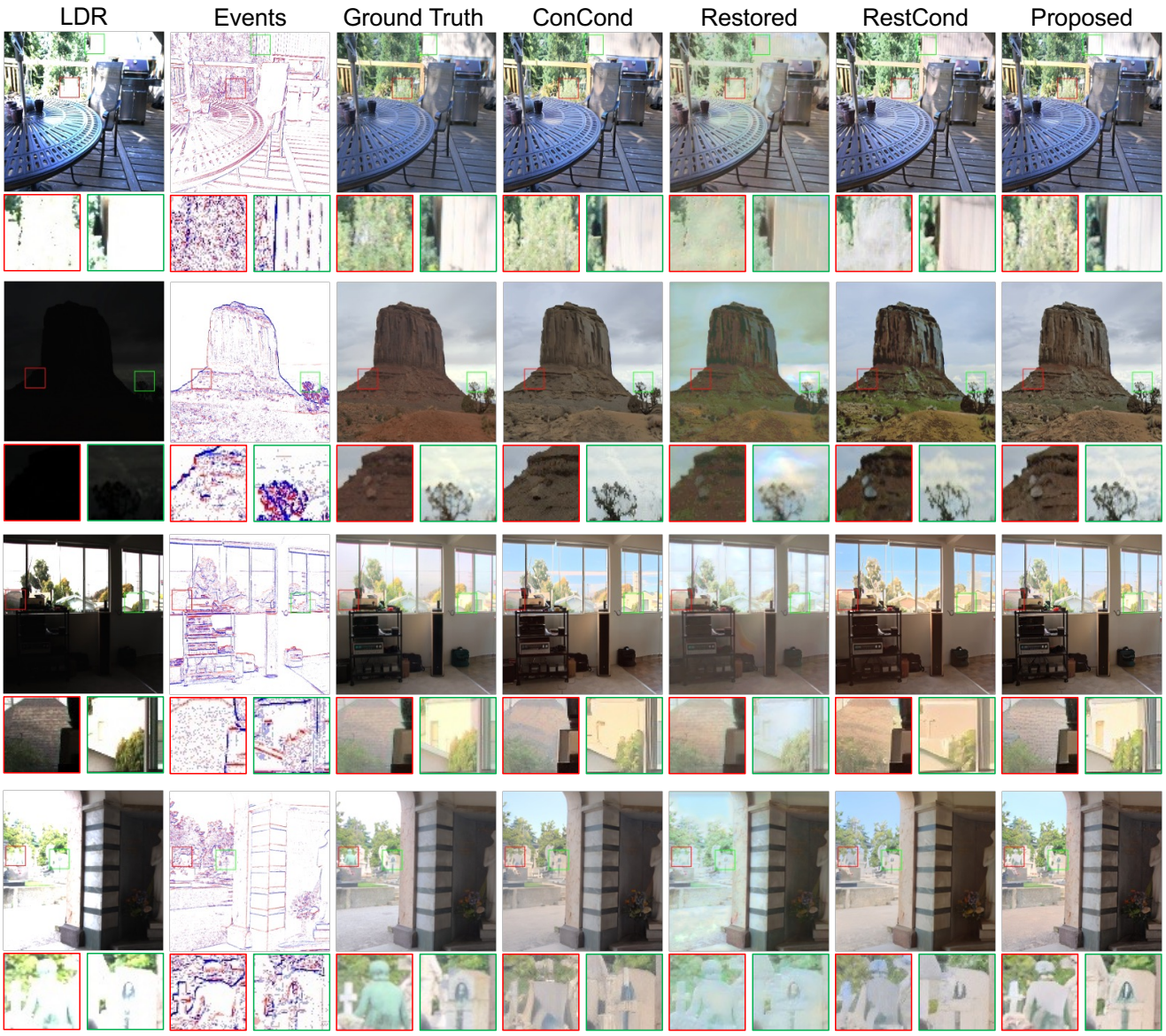

Figure 12. Qualitative comparisons on different conditioning and generation processes as described in Section 3.2. It is hard for "ConCond" to integrate events and LDR image to provide faithful results. "Restored" [49] effectively extracts details by fusing events and LDR image, while its results exist some artifacts. It also struggles with compensating for large over-exposed areas. "RestCond" provides better details than "ConCond" in nearly well-exposed areas, but it is misled by "Restored" to provide unsatisfactory results in large over-/under-exposed areas. The proposed method exploits the information in events and LDR image to provide faithful and colorful recovery results.

the difficulty of color prediction, as discussed in Section 11, which may lead to unnatural color transition as shown in the green box of the second row and red box of the fourth row in Figure 13. Introducing structure loss not only reduces the difficulty of learning but also provides pleasant visual results with natural contrast.

Efficiency comparison We calculate the Floating Point Operations Per Second (FLOPs), the total parameters (Params), and the running time of all ablation studies, as shown in Table 5. The ablation studies of structure loss share the same pipeline, which is the same as our complete model. The proposed conditioning and generation method is more efficient than "RestCond", which indicates that removing redundant decoder-encoder modules improves the efficiency of leveraging events and LDR image information.

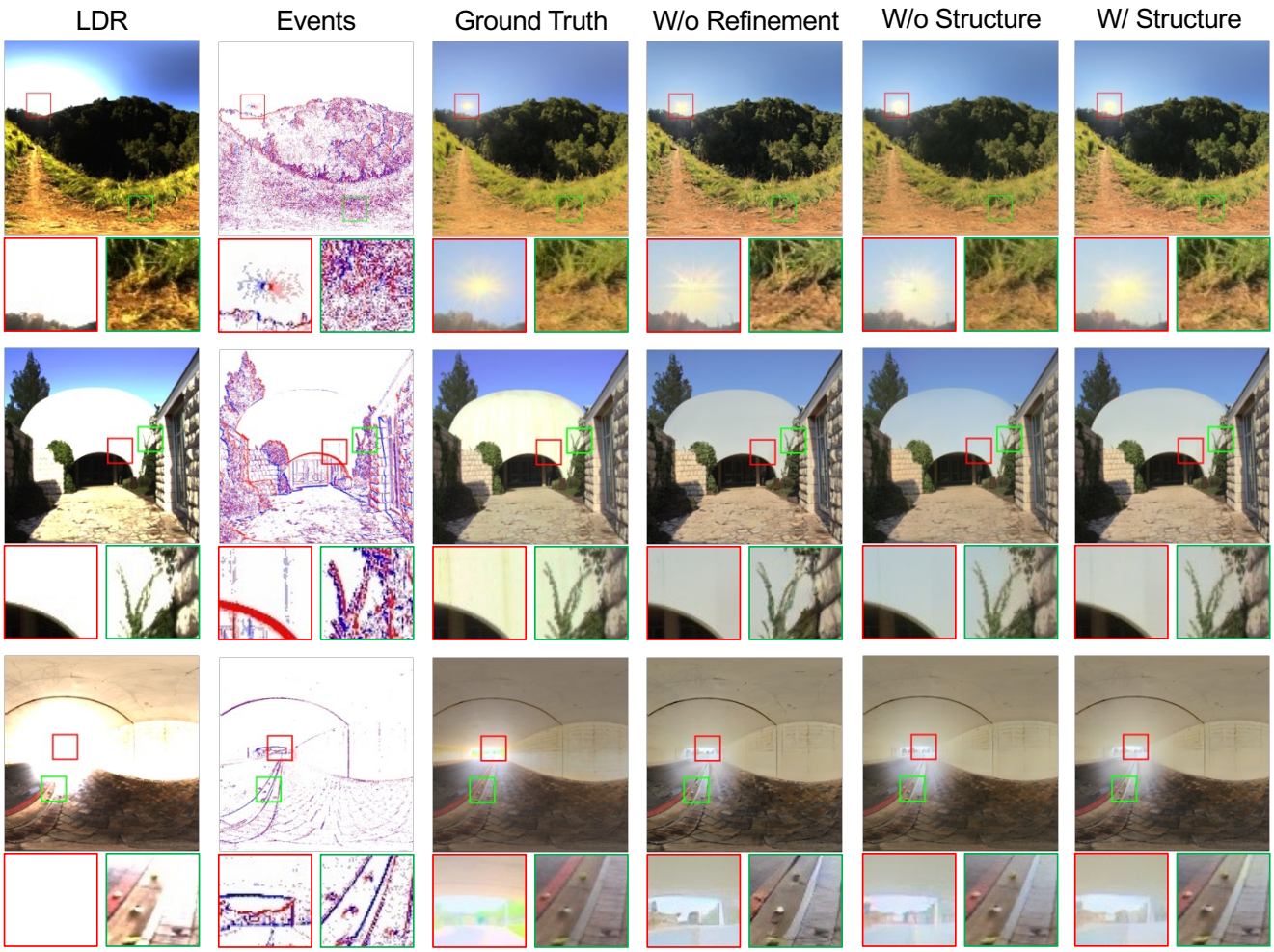

Figure 13. Qualitative comparisons on structure loss as described in Section 3.3. More natural images with higher contrast and better color appearance can be reconstructed with the proposed structure loss.

Table 6. Efficiency comparison of existing methods.

	FLOPs (G)	Params (M)	Time (s)
Liu *et al.* [25]	451.004	29.03	0.03
EventHDR [54]	1229.133	3.14	0.17
NeurImg [14]	301.385	37.39	0.08
HDRev [49]	821.551	57.94	0.79
Sagiri [22]	29327.032	1328.23	5.52
Ours	10380.843	1349.31	1.41

11. Diverse diffusion results

To demonstrate the uncertainty of our diffusion process, we randomly sample different results with the same input events and LDR image but different initial noise. The results are shown in Figure 14. Even with the sample input, the generation results by the diffusion process have large color differences even in training datasets. Directly training the refinement module with ground truth brings color uncertainty, resulting in unsatisfactory color adjustment as illustrated in Figure 5 and Figure 13. As demonstrated in Figure 11, the adapted ground truth, denoted by "Adapted GT", has a similar color appearance as diffusion results. Applying adapted ground truth as supervision targets makes the refinement module focus on detail refinement and retains the generation properties of diffusion models.

12. Human study

We conduct a human study on the real data (DSEC [8]) containing over-/normal-/under-exposed images for perceptual evaluation. We pick up 97 samples at equal intervals to construct our human perceptual dataset[1] to evaluate high-illuminance, low-illuminance, and overall quality by a sur-

[1]Please refer to our Github:github.com/YixinYang-00/HDRev-Diff.

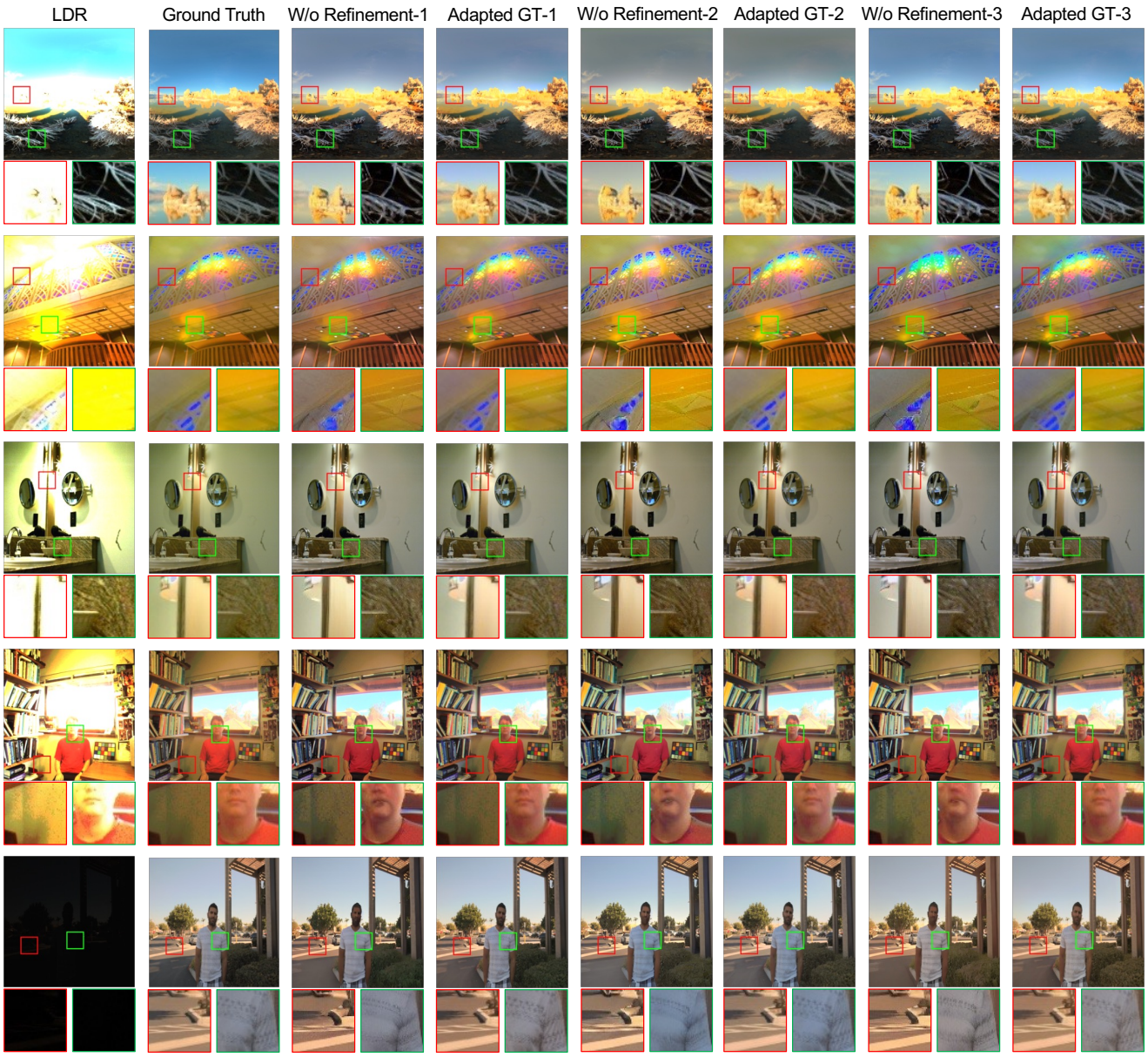

Figure 14. Diversity results with different initial noise. The adapted ground truth has similar color properties as diffusion results, which makes the refinement module focus on details refinement. Areas with obviously different colors are pointed out by red boxes.

vey link[2]. All methods are shuffled to avoid bias. Reports from 40 participants on all the samples, as shown in Table 7, the proposed method achievs highest preference among those three aspects.

13. Efficiency comparison of existing methods

The efficiency comparison is shown in Table 6. Regression-based methods [14, 25, 49, 54] have lower FLOPs, parameters, and running time. With a similar diffusion model back-

[2]Survey link:https://www.wjx.cn/vm/QzCmzw2.aspx#

Table 7. Quantitative evaluation of human study, which is evaluated in three aspects: `High`-illuminance, `Low`-illuminance, and `Overall` preference ratio.

	High	Low	Overall
Liu *et al.* [25]	23.00%	17.60%	17.78%
NeurIng [14]	2.58%	1.80%	0.90%
HDRev [49]	2.77%	7.80%	2.96%
Sagiri [22]	5.03%	1.74%	1.10%
Ours	66.62%	71.06%	77.26%

Figure 15. Qualitative evaluation of synthetic data on dataset collected by Yang *et al.* [49].

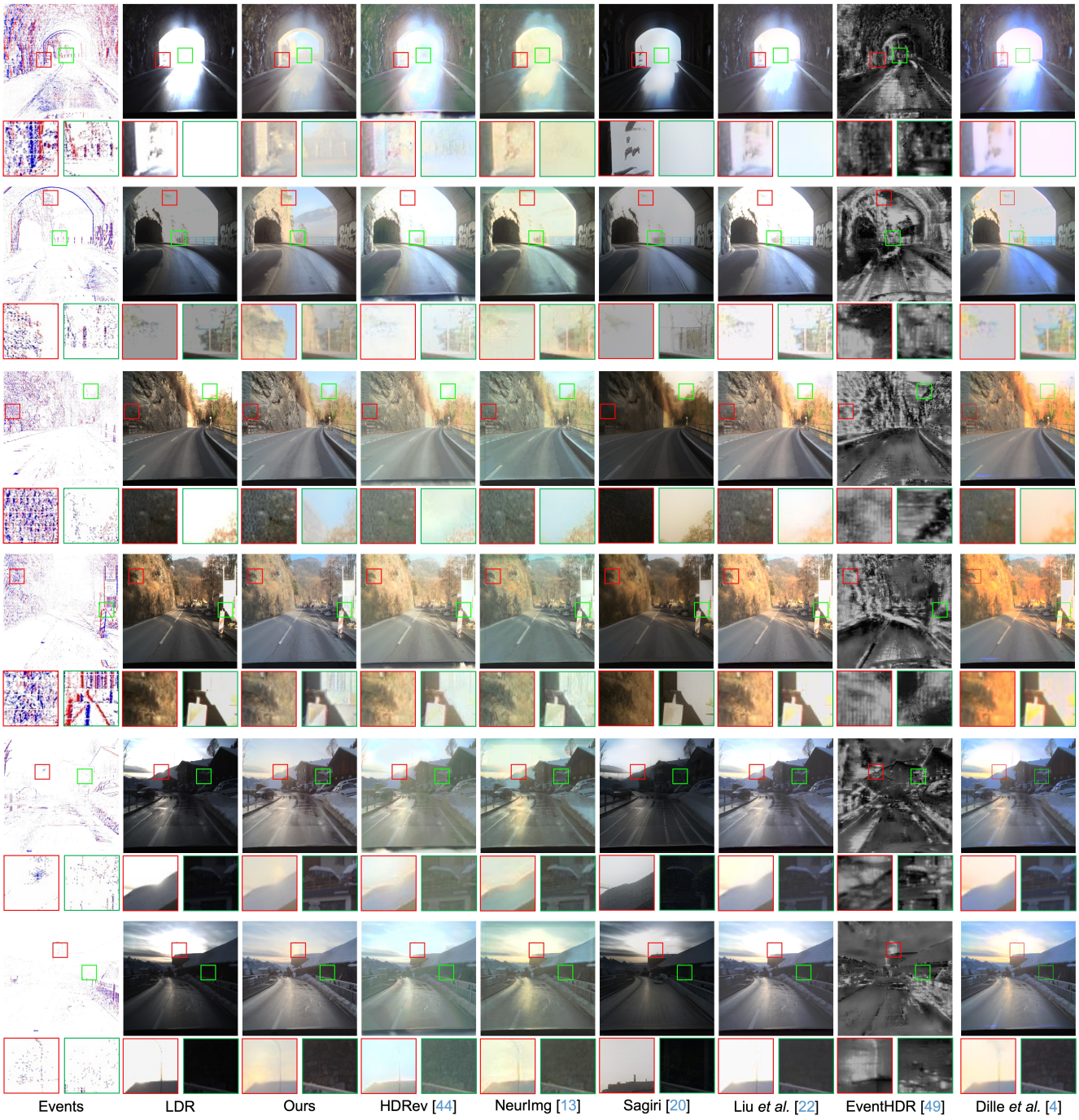

Figure 16. Qualitative comparisons of real data on DSEC [8].

bone, the proposed method has lower FLOPs (G) and faster running time than Sagiri [22]. Only with a slight parameter increase, we achieve better performance compared to Sagiri [22] as shown by Table 1.

14. More results on synthetic data

More quantitative comparisons are shown in Table 8. We add five metrics laid in two categories to further support our results. More qualitative comparisons are shown in Figure 15. EventHDR [54] only reconstructs HDR intensity. Liu *et al.* [25] cannot reconstruct HDR scenes only with a single LDR image as input. Besides, the details in dark regions are wiped as indicated by the green box of the fourth column in Figure 15. Sagiri [22] has difficulty maintaining consistency with LDR images and predicting missing information in over-/under-exposed areas. However, compared

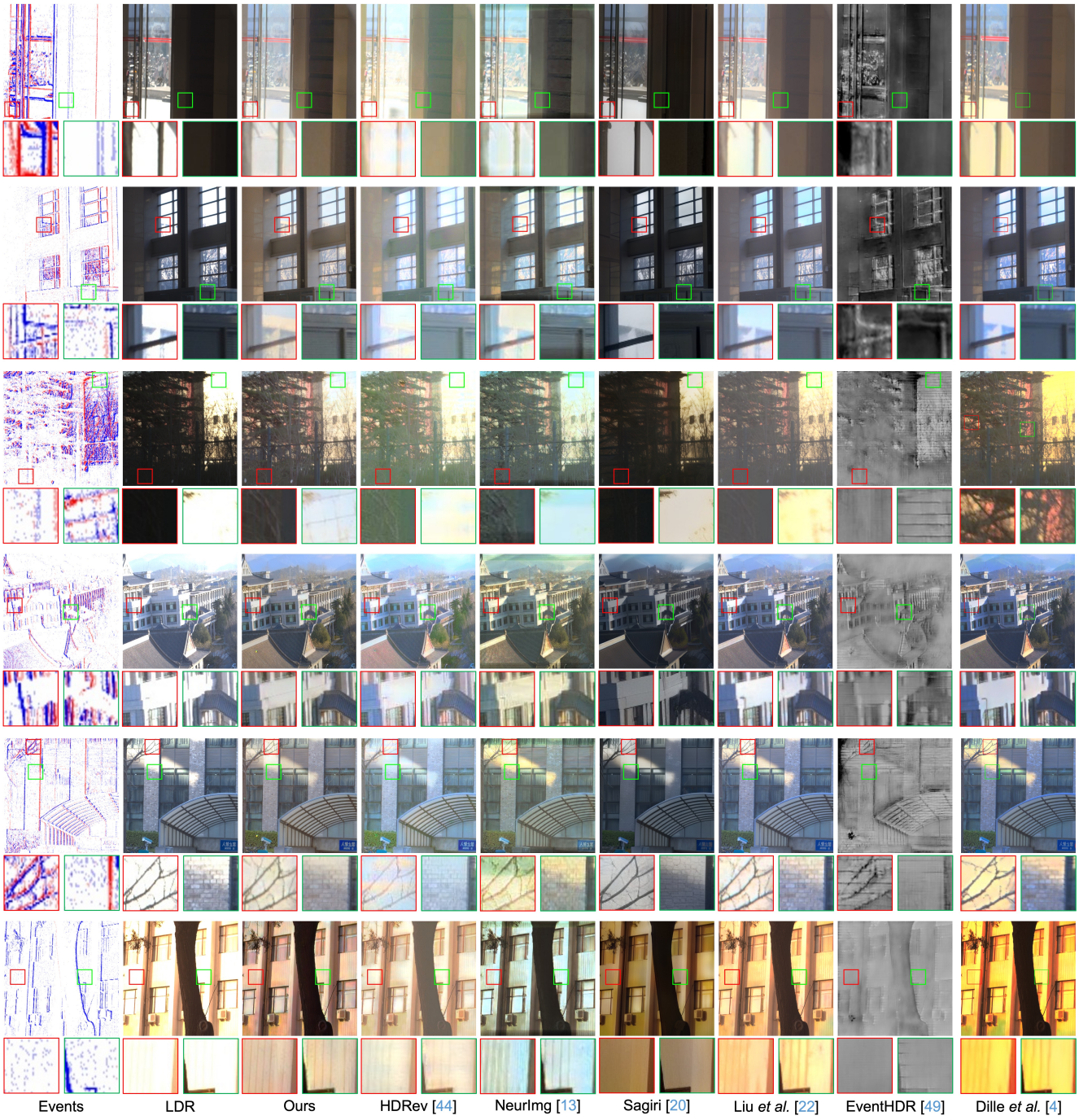

Figure 17. Qualitative comparisons of real data on HES-HDR [14].

with Liu *et al.* [25], Sagiri [22] can predict the handrail with diffusion priors as depicted in the second column of Figure 15. Although with events as input, NeurImg [14] and HDRev [49] are challenged to leverage the HDR information in events to reconstruct plausible results. Meanwhile, HDRev [49], which fuses events and images in the feature domain, shows better results than NeurImg [14], which fuses intensity images reconstructed from events with LDR images in the image domain. The better performance inspires us to extract conditions in the feature domain, instead of reconstructed HDR image at first. The proposed method reconstructs colorful and plausible results consistent with LDR images and events.

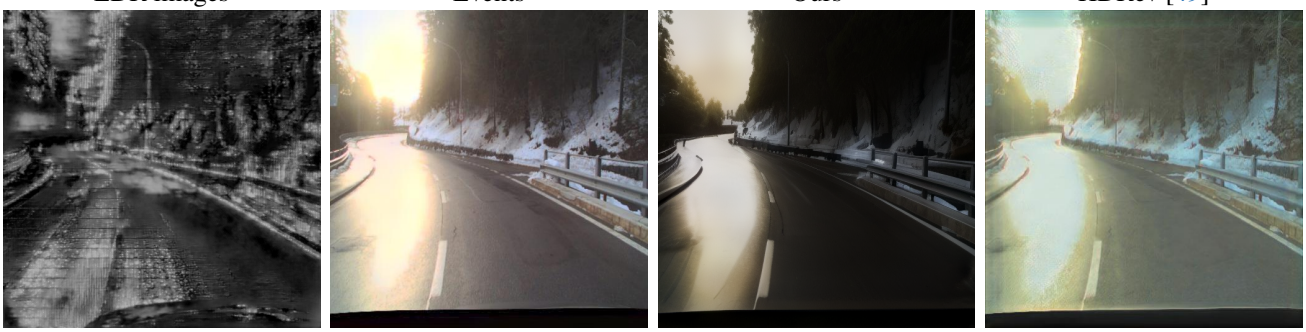

Figure 18. Consecutive results of the proposed method on DSEC [8] dataset. The proposed method shows the natural results with fewer artifacts and proper brightness. GIF animations could be displayed properly when viewed with Adobe Acrobat or KDE Okular.

Table 8. Additional quantitative evaluation of synthetic data.

		Liu *et al.*[25]	EventHDR [54]	Sagiri [22]	HDRev [49]	NeurImg [14]	Liang *et al.* [23]	Dille *et al.* [4]	Ours
Video Metrics	t-LPIPS[5]↓	0.025	0.112	0.107	0.024	0.021	0.086	0.012	0.018
	HDR-VQM↓	1.052	1.174	1.138	1.010	1.020	0.958	0.745	0.278
HDR Metrics	HDR-VDP-3↑	3.540	3.500	3.334	3.537	3.543	3.160	5.870	7.21
	PU-PSNR↑	24.643	23.432	23.49	23.71	24.73	21.96	32.39	32.20
	PU-SSIM↑	0.451	0.412	0.474	0.460	0.481	0.429	0.800	0.838

15. More results on real data

Comparison on DSEC dataset Additional results on DSEC [8] dataset are shown in Figure 16. It is difficult for EventHDR [54] to reconstruct distinguishable details on real data. Liu *et al.* [25] and Sagiri [22] are challenged in predicting over-exposed areas and retaining details in well-exposed and dark areas. NeurImg [14] is able to predict some of the information in over-exposed areas, while the results are low quality and have obviously artifacts in dark areas. HDRev [49] better preserves detail in well-exposed areas than NeurImg [14], while it is difficult to leverage the information in events to predict HDR images. The proposed method leverages the advantage of events and diffusion priors, providing natural and high-quality HDR images.

Comparison on HES-HDR dataset Additional results on the HES-HDR [14] dataset are shown in Figure 17. EventHDR [54] is challenged to reconstruct HDR information. Sagiri [22] and Liu *et al.* [25] can only hallucinate HDR information, which is difficult for large over-exposed areas, as shown in the third and sixth row in Figure 17. HDRev [49] and NeurImg [14] are also hard to compensate for over-exposed areas. The proposed method demonstrates superior performance in both compensating missing information for over-/under-exposed areas and preserving details in well-exposed regions as depicted by Figure 17.

16. Failure case in consecutive frames

We provide two results in Figure 18 to show our limitation on video generation. The proposed method does not consider the consecutive connection between adjacent frames. Although the proposed method maintains consistency with input LDR images and events, it cannot restore consecutive details for adjacent frames. Therefore, the over-exposed areas obviously flicker as demonstrated by the sky of Figure 18.