

# Supplementary Material for Deblur *e*-NeRF: NeRF from Motion-Blurred Events under High-speed or Low-light Conditions

Weng Fei Low<sup>✉</sup> and Gim Hee Lee<sup>✉</sup>

The NUS Graduate School’s Integrative Sciences and Engineering Programme (ISEP)  
Institute of Data Science (IDS), National University of Singapore  
Department of Computer Science, National University of Singapore  
{wengfei.low, gimhee.lee}@comp.nus.edu.sg  
<https://wengflow.github.io/deblur-e-nerf>

## A Logarithmic Photoreceptor

**Model.** As mentioned in Sec. 3.1, we model the radiance-dependent band-limiting behavior of the logarithmic photoreceptor with the following unity-gain 2<sup>nd</sup>-order Non-Linear Time-Invariant (NLTI) Low-Pass Filter (LPF) with input  $u_p = \log L$ , state  $\mathbf{x}_p = [\partial \log L_p / \partial t \log L_p]^\top$  and output  $y_p = \log L_p$ :

$$\begin{aligned} \dot{\mathbf{x}}_p(t) &= A_p(u_p(t)) \mathbf{x}_p(t) + B_p(u_p(t)) u_p(t), \\ y_p(t) &= C_p \mathbf{x}_p(t) \end{aligned} \quad (14)$$

$$\text{where } A_p(u) = \begin{bmatrix} -2\zeta(u)\omega_n(u) & -\omega_n^2(u) \\ 1 & 0 \end{bmatrix}, B_p(u) = \begin{bmatrix} \omega_n^2(u) \\ 0 \end{bmatrix}, C_p = [0 \ 1].$$

The derivation of this model follows closely that of the small signal model for the original adaptive variant of the logarithmic photoreceptor circuit [1, 2], but we account for the absence of an adaptive element in the circuit.

The radiance-dependent damping ratio  $\zeta$  and natural angular frequency  $\omega_n$  are, respectively, given by:

$$\zeta(u) = \frac{\tau_{out} + \tau_{in}(u) + (A_{amp} + 1)\tau_{mil}(u)}{2\sqrt{\tau_{out}(\tau_{in}(u) + \tau_{mil}(u))(A_{loop} + 1)}}, \quad (15)$$

$$\omega_n(u) = \sqrt{\frac{A_{loop} + 1}{\tau_{out}(\tau_{in}(u) + \tau_{mil}(u))}}, \quad (16)$$

where  $A_{amp}$  and  $A_{loop}$  are the amplifier and total loop gains of the photoreceptor circuit, respectively, and  $\tau_{out}$  is the time constant associated to the output node of the photoreceptor circuit and inversely proportional to the *photoreceptor bias current*  $I_{pr}$  (Fig. 3). Furthermore,  $\tau_{in}$  and  $\tau_{mil}$  are, respectively, the radiance-dependent time constants associated to the input node and *Miller capacitance* of the photoreceptor circuit, given by:

$$\tau_{in}(u) = \frac{C_{in}V_T}{\kappa \exp u} = \frac{C_{in}V_T}{\kappa L}, \quad (17)$$

$$\tau_{mil}(u) = \frac{C_{mil}V_T}{\kappa \exp u} = \frac{C_{mil}V_T}{\kappa L}, \quad (18)$$

where  $C_{in}$  and  $C_{mil}$  are the (lumped) parasitic capacitance on the photodiode and Miller capacitance in the photoreceptor circuit, respectively,  $V_T$  is the thermal voltage, and  $\kappa$  is the signal photocurrent  $I_p$  to incident radiance signal  $L_{sig}$  ratio governed by the photodiode.

**Behavior under Extreme Low Light.** As  $\tau_{out} \ll \tau_{in} + \tau_{mil}$  under extreme low light, the model described above reduces to a unity-gain 1<sup>st</sup>-order NLTI LPF with input  $u_{\hat{p}} = \log L$ , state  $x_{\hat{p}} =$  output  $y_{\hat{p}} = \log L_p$ :

$$\begin{aligned} \dot{x}_{\hat{p}}(t) &= A_{\hat{p}}(u_{\hat{p}}(t)) x_{\hat{p}}(t) + B_{\hat{p}}(u_{\hat{p}}(t)) u_{\hat{p}}(t), \\ y_{\hat{p}}(t) &= C_{\hat{p}} x_{\hat{p}}(t) \end{aligned}, \quad (19)$$

where  $A_{\hat{p}}(u) = -\omega_{c,\hat{p}}(u)$ ,  $B_{\hat{p}}(u) = \omega_{c,\hat{p}}(u)$  and  $C_{\hat{p}} = 1$ .

The cutoff angular frequency of this non-linear filter:

$$\omega_{c,\hat{p}}(u) = \frac{A_{loop} + 1}{\tau_{in}(u) + (A_{amp} + 1) \tau_{mil}(u)} \quad (20)$$

is directly proportional to the effective radiance  $L = L_{sig} + L_{dark}$ . Nonetheless, it remains very much smaller than the radiance-independent cutoff angular frequencies of the source follower buffer  $\omega_{c,sf}$  and differencing amplifier  $\omega_{c,diff}$ . Therefore, this rather simple 1<sup>st</sup>-order model forms the dominant pole approximation of the full 4<sup>th</sup>-order pixel bandwidth model under extreme low-light, which is relatively accurate. Furthermore, when  $L(t) \approx L_{dark}$ , we can further approximate this model with its linearized variant, which has a constant cutoff angular frequency of  $\omega_{c,\hat{p}}(\log L_{dark}) = \omega_{c,dom,min}$  (cf. Eq. (11)).

**Fundamental Limitations.** The logarithmic photoreceptor 2<sup>nd</sup>-order NLTI LPF is characterized by  $A_{amp}$ ,  $A_{loop}$ ,  $\tau_{out}$ ,  $C_{in}V_T/\kappa = \tau_{in}L$  and  $C_{mil}V_T/\kappa = \tau_{mil}L$ . When the unknown logarithmic photoreceptor model parameters are jointly optimized, the predicted effective radiance  $\hat{L}$  from the reconstructed NeRF is only accurate up a scale, since  $\tau_{in}$  and  $\tau_{mil}$  are invariant to the common scale of  $L$ ,  $C_{in}V_T/\kappa$  and  $C_{mil}V_T/\kappa$ . This further necessitates a translated-gamma correction (Sec. 3.4) of  $\hat{L}$  post-reconstruction.

## B Event Simulator

As alluded in Secs. 3.2 and 4, our event simulator extends the improved ESIM [8] event simulator introduced in Robust e-NeRF with the proposed pixel bandwidth model, particularly the discrete-time model given by Eq. (8). We appropriately

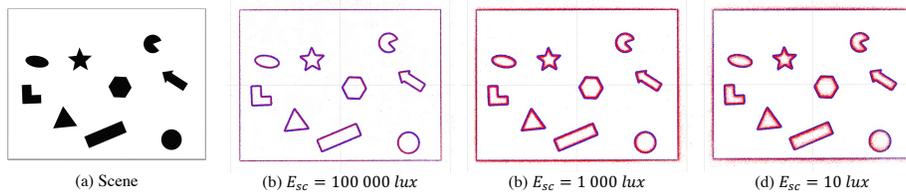


Fig. 6: Our simulated events transformed to the scene plane.

initialize the state of the 4<sup>th</sup>-order NLTI LPF with the steady-state  $\bar{\mathbf{x}}[k_0] = [0 \ u[k_0] \ u[k_0] \ u[k_0]]^\top$  on the initial input effective log-radiance  $u[k_0] = \log L[k_0]$ .

Fig. 6 depicts our simulated events on a simple planar scene of shapes under various scene illuminance  $E_{sc}$ . It can be observed that as the scene illumination improves, the events become more localized around the edges in the scene. Moreover, the spreading or blurring of negative events in red is more severe than that of positive events in blue. This happens because negative events involve a transition from a high to low effective log-radiance  $\log L$ , where the latter is associated to a low pixel bandwidth. All these observations validate the accuracy of our event simulator, as they conform to the expected behavior of an event sensor pixel.

## C Implementation Details

**Deblur *e*-NeRF.** The implementation of our method is based on Robust *e*-NeRF [5]. In particular, we adopt the same NeRF model architecture, parameterization of positive-to-negative contrast threshold ratio  $C_{+1}/C_{-1}$  and refractory period for joint optimization, NerfAcc [4]/Instant-NGP [7] parameters, training schedule, learning rates and constant-rate camera pose interpolation.

Nonetheless, since our method can theoretically reconstruct a NeRF with gamma-accurate predicted effective radiance  $\hat{L}$ , particularly under unknown contrast thresholds, we also parameterize the mean contrast threshold  $\bar{C} = \frac{1}{2}(C_{-1} + C_{+1})$ , which defines the scale of the contrast thresholds, via *SoftPlus* to ensure that it is always positive during its joint optimization. Such a parameterization of the contrast thresholds is optimal in the sense that the normalized predictions, which are  $\Delta \log \hat{L}_{blur}/\bar{C}$  for  $\ell_{diff}$  and  $\delta \log \hat{L}_{blur}/\bar{C}$  for  $\ell_{tv}$ , and normalized targets, which are  $pC_r/\bar{C}$  for  $\ell_{diff}$  and 0 for  $\ell_{tv}$ , are invariant to  $C_{+1}/C_{-1}$  and  $\bar{C}$ , respectively.

Furthermore, we parameterize the pixel bandwidth model parameters as  $A_{amp}^{-1}$ ,  $A_{loop}^{-1}$ ,  $\tau_{out}$ ,  $C_{mil}V_T/\kappa = \tau_{mil}L$ ,  $\tau_{sf} = \omega_{c,sf}^{-1}$  and  $\tau_{diff} = \omega_{c,diff}^{-1}$ , which generally has values smaller than 1, via *SoftPlus* as well for joint optimization. Note that we do not parameterize  $C_{in}V_T/\kappa = \tau_{in}L$ , but keep it fixed at an arbitrary positive value, as the predicted effective radiance is only accurate up to a scale when pixel bandwidth model parameters are jointly optimized (Appendix A).

This helps to clamp down on this gauge freedom during joint optimization, and yields a minimal parameterization of  $\tau_{in}$  and  $\tau_{mil}$  up to an arbitrary scale. Care must be taken to ensure that the predefined  $C_{in}V_T/\kappa$  is larger than the minimum effective radiance  $\epsilon = 0.001$  the NeRF model can output.

We adopt a sample size  $k - k_0 + 1$  of 30 for importance sampling of inputs  $u = \log L$  in all experiments. Moreover, we sample the optimal input sample timestamps  $T_i$  from the transformed exponential distribution given by Eq. (11), but truncated in practice to a finite support of  $(t_{k_0}, t_k]$  such that its cumulative probability is exactly 0.95. The sampling is done using a variant of inverse transform sampling, where instead of uniformly sampling the interval  $(0, 1]$  (and then applying the inverse cumulative distribution function), we directly take  $k - k_0 + 1 = 30$  evenly-spaced samples in the same interval. This helps to prevent significant under/over-representation of inputs  $u$  around certain time regions in the computation of the output  $\mathbf{y}[k]$ , due to randomness. Apart from that, since we assume  $u$  is stationary prior the start of the event sequence, we assign input samples with timestamps prior the start to have the same value as the initial input.

Furthermore, we sample each subinterval  $(t_{start}, t_{end}]$  between the interval  $(t_{ref}, t_{curr}]$  for use in  $\ell_{tv}$ , by first sampling the length of the subinterval  $t_{end} - t_{start}$  from a triangular distribution with a support of  $[0, t_{curr} - t_{ref})$  and a mode of 0, then sampling  $t_{start}$  from a uniform distribution with a support of  $[0, (t_{curr} - t_{ref}) - (t_{end} - t_{start}))$ . Joint optimization of the pixel bandwidth model parameters is done with the same learning rate of 0.01 as the NeRF model parameters. Moreover, we train our method with loss weights of  $\lambda_{diff} = 1$  and  $\lambda_{tv} = 0.1$ , as well as a batch size (defined according to Robust  $e$ -NeRF) of  $2^{17} = 131\,072$ , by default.

However, we observed that our loss values for the threshold-normalized difference loss  $\ell_{diff}$ , under the *hard* setting ( $v = 4\times, E_{sc} = 10lux$ ) in the collective effect synthetic experiment, is  $\sim 100\times$  smaller than that of other settings, but the loss values for threshold-normalized total variation loss  $\ell_{tv}$  (Eq. (12)) remains in the same order. This will cause the total training loss  $\mathcal{L}$  (Eq. (2), but with  $\lambda_{tv}\ell_{tv}(\mathbf{e})$  instead of  $\lambda_{grad}\ell_{grad}(\mathbf{e})$ ) to be inappropriately dominated by the regularization loss  $\ell_{tv}$ , instead of the primary reconstruction loss  $\ell_{diff}$ , if the default loss weights are used.

Thus, we adopt  $\lambda_{tv} = 0.001$ , which is  $100\times$  smaller than the default, under the *hard* setting to rebalance both losses. Apart from that, we also adopt  $\lambda_{tv} = 0.01$ , which is  $10\times$  smaller than the default, under the  $E_{sc} = 10lux$  setting in the synthetic experiment studying the effect of scene illuminance, due to similar observations. As we employ the Adam [3] optimizer, which is invariant to diagonal rescaling of gradients hence loss, this is equivalent to a loss weight  $\lambda_{diff}$  of  $100\times$  or  $10\times$  larger than that of the default, while maintaining  $\lambda_{tv}$  at the default.

**Baselines.** We employ the official implementation of Robust  $e$ -NeRF in our experiments. However, we adopt  $\lambda_{grad} = 0.00001$  and  $\lambda_{grad} = 0.0001$ , which are  $100\times$  and  $10\times$  smaller than the default at  $\lambda_{grad} = 0.001$ , under the *hard* ( $v =$

$4\times, E_{sc} = 10lux$ ) and  $E_{sc} = 10lux$  settings in synthetic experiment, respectively, due to similar observations made in our method. Moreover, we implement E2VID + NeRF according to how it is done for the experiments in Robust  $e$ -NeRF.

**Translated-Gamma Correction.** To account for the time-varying sensor gain (*i.e.* ISO) and exposure time of the captured reference images, particularly during evaluation, we additionally scale each correction with the known gain-exposure product of the corresponding reference image.

The optimal correction parameters  $a$ ,  $\mathbf{b}$  and  $\mathbf{c}$  are optimized using the Levenberg-Marquardt algorithm with a *Trust Region* strategy to determine the optimal damping factor at each iteration. We adopt the implementation provided by PyPose [9], as well as its default hyperparameters. Furthermore, we appropriately initialize the optimization with  $\mathbf{c} = \mathbf{0}$  and the solution of  $a$  and  $\mathbf{b}$  given by gamma correction (Eq. (5)). The optimization is performed until the sum of squared correction errors has converged, up to a maximum of 20 iterations.

## D Interpretation of Real Quantitative Results

Note that care must be taken when interpreting the quantitative results of the real experiments presented in Tab. 2 and Tab. 10, since they are not truly indicative of the *absolute* performance of all methods, but likely only indicative of their *relative* performance. This is due to the fact that the target novel views, given by a separate standard camera, suffer from motion blur, rolling shutter artifacts, and saturation, as a result of a significantly smaller dynamic range compared to an event camera. Furthermore, the target novel views are not raw images that does not depend on the unknown *Camera Response Function* (CRF), and are grayscale images converted from RGB images provided by the camera, which might not reflect the spectral sensitivity of the monochrome event camera.

## E Additional Experiment Results

### E.1 Per-Scene Breakdown

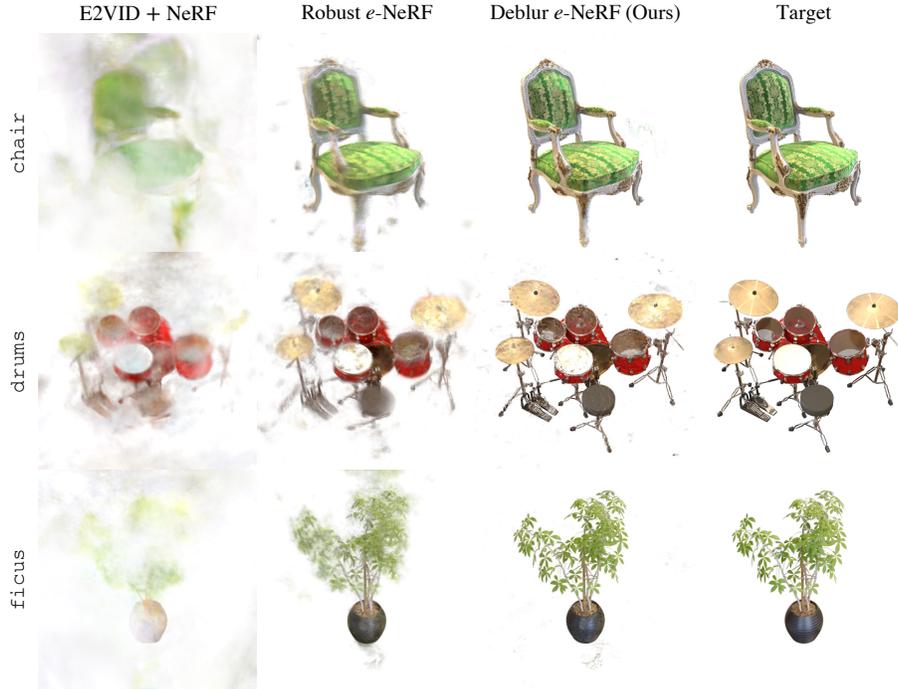
Tab. 6 and Figs. 7 and 8 show the quantitative and qualitative results of all methods, respectively, for all synthetic scene sequences simulated with the hard setting ( $v = 4\times, E_{sc} = 10lux$ ). The results clearly demonstrate our superior performance in reconstructing a blur-minimal NeRF from motion-blurred events.

### E.2 Ablation on Pixel Bandwidth Model

To further ascertain the role of the proposed pixel bandwidth model, we evaluate our method with and without the pixel bandwidth model incorporated, under the same settings as the synthetic experiment in studying the collective effect, without joint optimization of pixel bandwidth model parameters. The quantitative results given in Tab. 7 undoubtedly verifies the importance of the pixel bandwidth model in accounting for event motion blur.

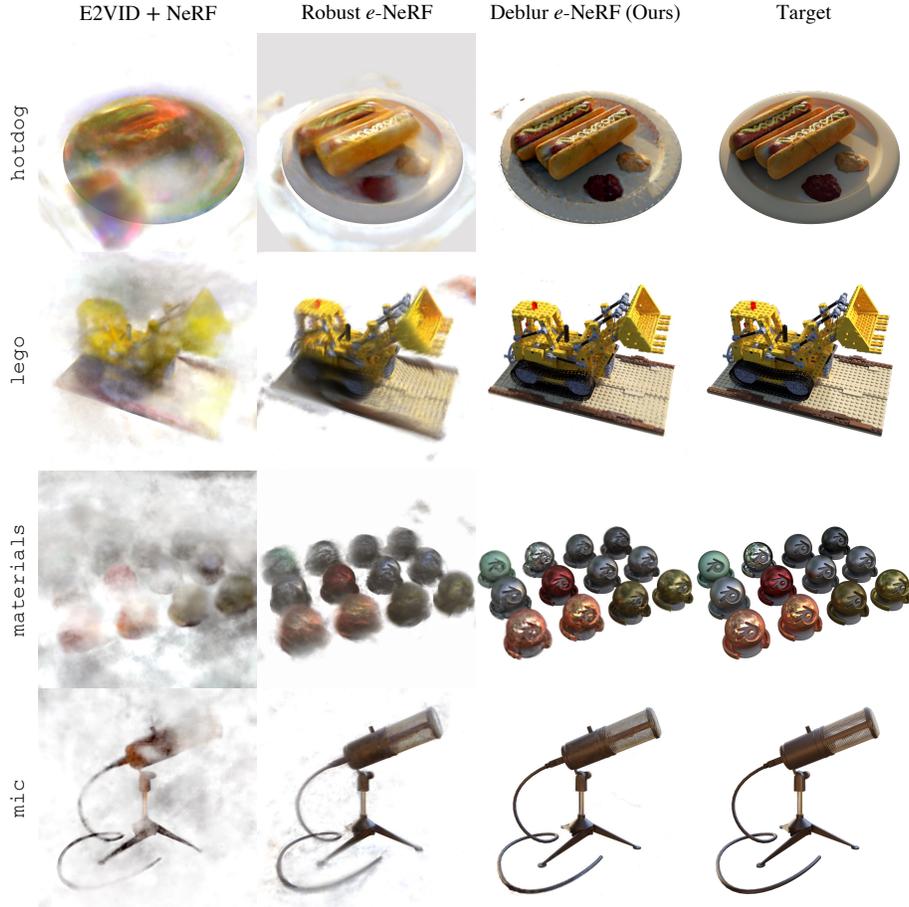
**Table 6:** Per-synthetic scene breakdown under the hard setting. <sup>†</sup>Trained with  $1/8\times$  the batch size of baselines.

Metric	Method	Synthetic Scene							Mean
		Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	
PSNR $\uparrow$	E2VID + NeRF	16.67	15.00	16.25	17.53	14.75	11.65	15.72	15.37
	Robust $e$ -NeRF	21.64	17.41	21.80	15.05	18.28	15.68	19.11	18.42
	Deblur $e$ -NeRF <sup>†</sup>	<b>27.39</b>	<b>22.14</b>	<b>29.10</b>	<b>23.69</b>	<b>27.69</b>	<b>24.49</b>	<b>28.53</b>	<b>26.15</b>
SSIM $\uparrow$	E2VID + NeRF	0.835	0.776	0.840	0.842	0.719	0.726	0.854	0.799
	Robust $e$ -NeRF	0.836	0.758	0.864	0.849	0.754	0.772	0.862	0.814
	Deblur $e$ -NeRF <sup>†</sup>	<b>0.902</b>	<b>0.839</b>	<b>0.944</b>	<b>0.904</b>	<b>0.896</b>	<b>0.890</b>	<b>0.951</b>	<b>0.904</b>
LPIPS $\downarrow$	E2VID + NeRF	0.374	0.498	0.310	0.391	0.509	0.589	0.380	0.436
	Robust $e$ -NeRF	0.216	0.336	0.146	0.287	0.279	0.295	0.228	0.255
	Deblur $e$ -NeRF <sup>†</sup>	<b>0.107</b>	<b>0.231</b>	<b>0.120</b>	<b>0.168</b>	<b>0.105</b>	<b>0.116</b>	<b>0.093</b>	<b>0.134</b>

**Fig. 7:** Synthesized novel views on chair, drums and ficus under the hard setting

### E.3 Effect of Reduced Batch Size

To assess the true impact of training with a reduced batch size, we also benchmark our method with a batch size of  $1/8\times$  and  $1\times$  that of our baselines, under



**Fig. 8:** Synthesized novel views on hotdog, lego, materials and mic under the hard setting

the same settings as the experiment in Appendix E.2, but only on the lego scene. The quantitative results reported in Tab. 8 provide a glimpse into the true strength of our method, as significant improvements can be observed as the batch size increases to that of the baselines.

#### E.4 Ablation on Input Sample Size

We perform a cost-benefit analysis on the input sample size  $k - k_0 + 1$  of our method on the lego scene under the hard setting ( $v = 4\times$ ,  $E_{sc} = 10lux$ ). The quantitative results presented in Tab. 9 suggests that our default input sample size of 30 strikes the best balance between cost and performance. Note that the computational and memory cost of our method is proportional to the input

**Table 7:** Ablation on pixel bandwidth model

Pixel Bandwidth Model	$v = 0.125\times, E_{sc} = 100\ 000lux$			$v = 1\times, E_{sc} = 1\ 000lux$			$v = 4\times, E_{sc} = 10lux$		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
$\times$	28.75	0.948	0.048	26.98	0.934	0.061	18.31	0.822	0.245
$\checkmark$	<b>29.00</b>	<b>0.950</b>	<b>0.043</b>	<b>28.41</b>	<b>0.947</b>	<b>0.049</b>	<b>26.15</b>	<b>0.904</b>	<b>0.134</b>

**Table 8:** Effect of reduced batch size on the **lego** scene

Batch Size, $\times$	$v = 0.125\times, E_{sc} = 100\ 000lux$			$v = 1\times, E_{sc} = 1\ 000lux$			$v = 4\times, E_{sc} = 10lux$		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
1/8	29.44	0.940	0.045	28.42	0.938	0.048	27.69	0.896	0.105
1	<b>31.27</b>	<b>0.953</b>	<b>0.030</b>	<b>30.43</b>	<b>0.950</b>	<b>0.038</b>	<b>30.72</b>	<b>0.948</b>	<b>0.037</b>

**Table 9:** Ablation of input sample size on **lego** under the hard setting

Input Sample Size	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
1	18.46	0.765	0.273
5	22.64	0.807	0.21
15	26.41	0.875	0.125
30	27.69	0.896	0.105
50	28.18	0.902	0.097
75	<b>28.21</b>	<b>0.903</b>	<b>0.096</b>

sample size, as alluded in Sec. 5, and an input sample size of 1 is equivalent to having the pixel bandwidth model removed.

### E.5 Results on 07\_ziggy\_and\_fuzz\_hdr

Apart from 08\_peanuts\_running and 11\_all\_characters, we also benchmark all methods on the 07\_ziggy\_and\_fuzz\_hdr sequence from the EDS dataset, which involves a HDR scene with occasional high-speed camera motion. The quantitative and qualitative results given in Tab. 10 and Fig. 9 once again demonstrates our superior performance, as the objects on the table are clearly more well-defined and the table surface, wall and curtains are much smoother, while preserving details and color accuracy of the scene.

### E.6 Comparison with Image Blur-Aware Baselines

While motion blur in standard and event cameras are vastly different, and thus incomparable, we provide additional quantitative results of 2 other *image* blur-aware baselines: E2VID + MPRNet [10] (a seminal image deblurring method) + NeRF and E2VID + Deblur-NeRF [6] (a seminal NeRF with image blur model), for selected synthetic experiments (*i.e.* upper bound performance and collective

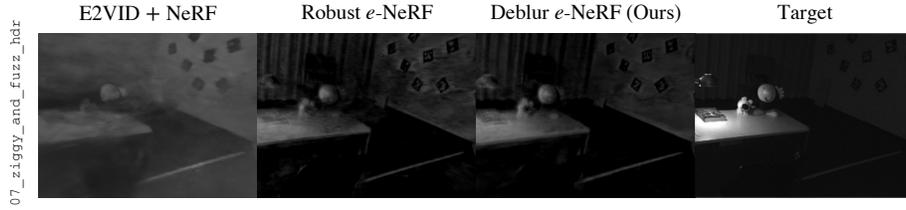


Fig. 9: Synthesized novel views on the 07\_ziggy\_and\_fuzz\_hdr scene

Table 10: Quantitative results on the 07\_ziggy\_and\_fuzz\_hdr scene

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
E2VID + NeRF	14.96	<b>0.691</b>	0.556
Robust $e$ -NeRF	18.02	0.631	0.464
Deblur $e$ -NeRF	<b>18.47</b>	0.648	<b>0.440</b>

Table 11: Comparison with image blur-aware baselines built upon E2VID.

Simulation Settings / Real Scene	E2VID + NeRF			E2VID + MPRNet + NeRF			E2VID + Deblur-NeRF		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
No event motion blur	19.49	0.847	0.268	19.44	<b>0.851</b>	<b>0.267</b>	<b>19.84</b>	0.839	0.291
$v = 0.125\times, E_{sc} = 100\ 000lux$	<b>19.19</b>	0.844	0.281	19.18	<b>0.849</b>	<b>0.260</b>	19.15	0.841	0.288
$v = 1\times, E_{sc} = 1\ 000lux$	18.85	0.839	0.278	<b>18.86</b>	<b>0.843</b>	<b>0.269</b>	18.73	0.818	0.317
$v = 4\times, E_{sc} = 10lux$	15.37	<b>0.799</b>	<b>0.436</b>	<b>15.44</b>	0.794	0.439	15.42	0.783	0.472
07_ziggy_and_fuzz_hdr	<b>14.96</b>	<b>0.691</b>	0.556	<b>14.96</b>	<b>0.691</b>	0.552	14.85	0.680	<b>0.504</b>
08_peanuts_running	14.85	<b>0.690</b>	0.595	14.81	<b>0.690</b>	0.604	<b>14.91</b>	0.682	<b>0.517</b>
11_all_characters	<b>13.12</b>	<b>0.695</b>	0.627	13.10	<b>0.695</b>	0.624	12.95	0.689	<b>0.576</b>

effect) and the real experiment, to make our experiments more complete. From the results reported in Tab. 11, it is evident that the incorporation of an image blur or deblur model is unable to account for event motion blur, as the performance is virtually the same with or without it. This reinforces the importance for our physically-accurate pixel bandwidth model to account for event motion blur under arbitrary speed and lighting conditions.

## References

1. Delbruck, T., Mead, C.A.: Analog VLSI Phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits. Vision Chips: Implementing vision algorithms with analog VLSI circuits (1995)
2. Delbruck, T.: Investigations of analog VLSI visual transduction and motion processing. Doctoral Thesis, California Institute of Technology (1993)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)

4. Li, R., Gao, H., Tancik, M., Kanazawa, A.: Nerfacc: Efficient sampling accelerates nerfs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
5. Low, W.F., Lee, G.H.: Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
6. Ma, L., Li, X., Liao, J., Zhang, Q., Wang, X., Wang, J., Sander, P.V.: Deblur-NeRF: Neural Radiance Fields From Blurry Images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
7. Müller, T., Evans, A., Schied, C., Keller, A.: Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* (2022)
8. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an Open Event Camera Simulator. In: Proceedings of The 2nd Conference on Robot Learning, PMLR (2018)
9. Wang, C., Gao, D., Xu, K., Geng, J., Hu, Y., Qiu, Y., Li, B., Yang, F., Moon, B., Pandey, A., Aryan, Xu, J., Wu, T., He, H., Huang, D., Ren, Z., Zhao, S., Fu, T., Reddy, P., Lin, X., Wang, W., Shi, J., Talak, R., Cao, K., Du, Y., Wang, H., Yu, H., Wang, S., Chen, S., Kashyap, A., Bandaru, R., Dantu, K., Wu, J., Xie, L., Carbone, L., Hutter, M., Scherer, S.: PyPose: A library for robot learning with physics-based optimization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
10. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)