Neural Ganglion Sensors: Learning Task-specific Event Cameras Inspired by the Neural Circuit of the Human Retina

Supplementary Material

1. Code

Our code for learned RGC events will be open sourced on GitHub, along with how to integrate our differentiable event simulator.

2. Additional Experimental Details

2.1. Closed Form Binning, with refractory period

In the main paper, we show the closed form equations to extract binned events from high speed video. Here, we extend them to account for the refractory period, r, the "off" period of a pixel after it has fired an event, and p_r, the time since the last event timestamp, t^e_{mem}, during event generation from the previous frame.

$$p_r = t_k - t^e_{mem} \tag{1}$$

$$\alpha^r = (r//\alpha + 1) \cdot \alpha \tag{2}$$

$$\begin{aligned} \beta^r &= \beta + \max(0, ((r - p_r)//\alpha) \cdot \alpha) \\ &= \max(0, ((r - p_r)//\alpha) \cdot \alpha) \end{aligned} \tag{3}$$

The weight and number of events become

$$w^r_i = \frac{\beta^r + i * \alpha^r}{\text{bin}^+ - \text{bin}^-} \tag{4}$$

$$\mathcal{N}^r = (t_{k+1} - t_k - \beta^r)//\alpha_r \tag{5}$$

and the closed form bin^- and bin^+ are

$$\begin{aligned} \text{bin}^-_{frame} = pol \cdot &\left(1 - \frac{\beta^r}{\text{bin}^+ - \text{bin}^-}\right) \cdot \mathcal{N}^r \\ &- \left(\frac{\alpha^r}{\text{bin}^+ - \text{bin}^-}\right) \cdot pol \cdot \frac{(\mathcal{N}^r + 1)(\mathcal{N}^r)}{2} \end{aligned} \tag{6}$$

$$\begin{aligned} \text{bin}^+_{frame} = pol \cdot &\left(\frac{\beta^r}{\text{bin}^+ - \text{bin}^-}\right) \cdot \mathcal{N}^r \\ &+ \left(\frac{\alpha^r}{\text{bin}^+ - \text{bin}^-}\right) \cdot pol \cdot \frac{(\mathcal{N}^r + 1)(\mathcal{N}^r)}{2} \end{aligned} \tag{7}$$

Now with these closed form equations and straight-through gradient estimation, we can model the gradients to enable learning even with non-zero refractory periods. In our experiments, the refractory period was set to 1 millisecond, following V2E2V [2]. We also model noise non-idealities with Gaussian noise with sigma 0.03, following the V2E emulator [1]. Ultimately, our closed form solution with gradients produces event bins that match closely with current state-of-the-art non-differentiable video to binned event pipelines.

2.2. Dataset Considerations and Details

For video interpolation, we utilize the GoPRO dataset [3]. High speed video frames are captured at 240 fps. There are 22 scenes in the train set and 11 scenes in the test set. To train, each sample is a 9-frame sequential subset of a scene. The two end-point frames are used as input to the base model, REFID, while the rest are used to simulate the events. The seven frames in-between the end-points are also used as the ground truth frames for comparison and metric calculation.

For optical flow, we utilize the TartanAir Dataset: AirSim Simulation Dataset for Simultaneous Localization and Mapping [5]. In our experiments, we test our methods with the 'Hard' subset of the TartanAir Dataset. There are 18 environments total. We randomly set 'office2' and 'westerndesert' scenes as validation, and 'gascola' and 'japanesealley' as the test scenes prior to any experiments. As mentioned in the main paper, in order to simulate events, we first interpolate 15 frames between each original neighboring frames. Similar to previous work [7], we generate the dataset of high speed video using EMA-VFI to interpolate for training. Our resulting training dataset consists of random crops from the original dataset along with the interpolated frames, creating a dataset of 44,776 training, 5,000 validation, and 5,000 test sequences. All experiments shown in this work are trained and evaluated on this created dataset. TartanAir has ground truth masks for optical flow which mask out pixels that have become occluded or disoccluded. Following the convention of previous work, we train and evaluate on the masked optical flow.

2.3. Model Details

2.3.1. Event-based Video Interpolation

We utilize REFID [4], a state-of-the-art model for event-based video interpolation. It uses an event-guided adaptive channel attention and bidirectional event recurrent blocks. The model we use is the default from the repo with num-encoders=3, 32 base channels, num-block=1, and 2 residual blocks. With one type of RGC kernel, the default number of event channels is 2. As we experiment with different number of types of RGC kernels, we scale just the input layer accordingly. In all experiments, models are trained for 200k iterations using the PyTorch AdamW optimizer with a learning rate of $2e^{-4}$ for the interpolation model, $5e^{-5}$ for the RGC kernels, and weight decay $1e^{-4}$. For more details, see the code.

2.3.2. Event-based Optical Flow

We use IDNet [6], the light-weight state-of-the-art model for event-based optical flow. The backbone is the recurrent neural network ConvGRU that processes the event bins sequentially. Specifically, we use IDNet's id-8x variation. In all experiments, we train for 400k iterations with the PyTorch Adam optimizer with a learning rate of $1e^{-4}$.

2.4. Additional Interpolation Results

We include an extended comparison for bandwidth vs performance in fig. 2. For a no-events baseline, we train the same model with zero events. The average PSNR over the test set is 28.91dB and the SSIM is 0.912. In addition to the CSDVS and CSDVS-Delbrück shown in the main paper, we explore CSDVS-lin, CSDVS in the linear intensity domain. In neuroscience, another common way to model the center-surround is with a Difference of two Gaussian kernels of different sizes. Here, we model the Difference of Gaussians, learning the standard deviation of the two Gaussians as we tune the bandwidth via the weighting on the sparsity loss. Fig. 2 shows the full PSNR vs. Bandwidth trade-offs. We use a logarithmic fit for all event types. While CSDVS performs similarly to the DVS, CSDVS in the linear domain performs better. The Difference of Gaussians is also an improvement over the DVS kernel. While it may approximately model human RGCs well, for machine vision tasks, the best is still the learned RGC-lin and the spatially-varying RGC-lin-sv.

2.4.1. The effect of increasing the receptive field

We experiment with different kernel sizes $k = 3, 5, 7, 9, 11$ for the video interpolation task. All of these models are trained with the same settings, except for kernel size. All models converged at roughly $20,000$ events per bin with a PSNR in the range 34.36dB to 34.52dB and SSIM of 0.97 across the board. Although the interpolation performance does not increase significantly as we increase the receptive field, interestingly a faint structure is revealed. Immediately outside of the center pixel, we have a negative weighting, and just further out, a positively weighted ring, before decreasing once more.

2.4.2. The effect of increasing bandwidth

As we tune the bandwidth by weighting the sparsity loss more or less, the learned kernel also changes. In fig. 4, we show the learned RGC kernel for models trained at different bandwidths. The performance increases from 31.01dB with 436 events per bin to 34.45dB with $19,557$ events, to 36.52dB with $121,343$ events.

2.5. Multi-channel RGCs for Video Interpolation

2.5.1. Additional Multi-channel Results

In the main paper, we compared learning 1, 2, and 4 different, yet complementary, kinds of RGCs. In fig. 5, we show the learned kernels and the generated events. All models produced roughly $150,000$ events per bin. Here, we also show qualitative results for learning 16 kernels. Fig. 6 provides a view of the 16 learned kernels and the single learned kernel comparison at the ultra-low-bandwidth regime of roughly $10,000$ events. Mentioned in the main paper, at the same bandwidth, RGC_{16} achieves 35.16dB while RGC_1 achieve 33.76dB. We show two scenes and the events generated by each kernel. While the kernel learned for single RGC is quite symmetric, we can see the 16 kernels model can learn more specific features. We show samples of the reconstruction by the models with different number of channels in fig. 7 and fig. 8 with PSNR and SSIM metrics and some with zoom-in details. With more learned kernels, there is less warping and color discrepancies.

2.5.2. Trade-off of Performance and Bandwidth

Fig. 9 shows how using multiple kinds of events can further improve the performance at any given bandwidth. The i in RGC_i refers to how many parallel circuits are learned per pixel. This is similar to how our eyes operate. For a given receptive field, there can be multiple kinds of RGCs looking at the signals. In fig. 9, we also show DVS for reference as well as the RGC-lin-sv model, which is spatially varying kernels. RGC_4 and RGC-lin-sv are similar in that they both have 4 learned kernels. However, the spatially varying model is 1 kernel per pixel, as opposed to 4 per pixel. We can see that the performance for video interpolation is similar.

2.5.3. Additional considerations for Multi-event Bandwidth

Typically, the event packet that the traditional DVS outputs includes the pixel location and a bit denoting the polarity. In the multi-event case, we can send the location and a few bits. As we see in the generated events by each kernel in the main paper, one or two types of RGC seem to dominate or produce the most events, while the others complement with additional information. There can be an improvement in bandwidth if the system can support variable length packets. We can take advantage of Huffman encoding where fixed binary codes can be assigned to different RGC types based on their expected frequency. In this way, it can send the minimum number of bits and only send additional bits for rarer events when needed. Alternatively, when learning only a few RGC kernels, using spatially-varying kernels as demonstrated with RGC-lin-sv could potentially enable most of the performance benefits without requiring additional bits to be read out.

2.6. Additional Optical Flow Samples

We provide more qualitative results for optical flow in fig. 10. Again, we compared the DVS model with the best performance, DVS 0.1T, against our learned, single kernel

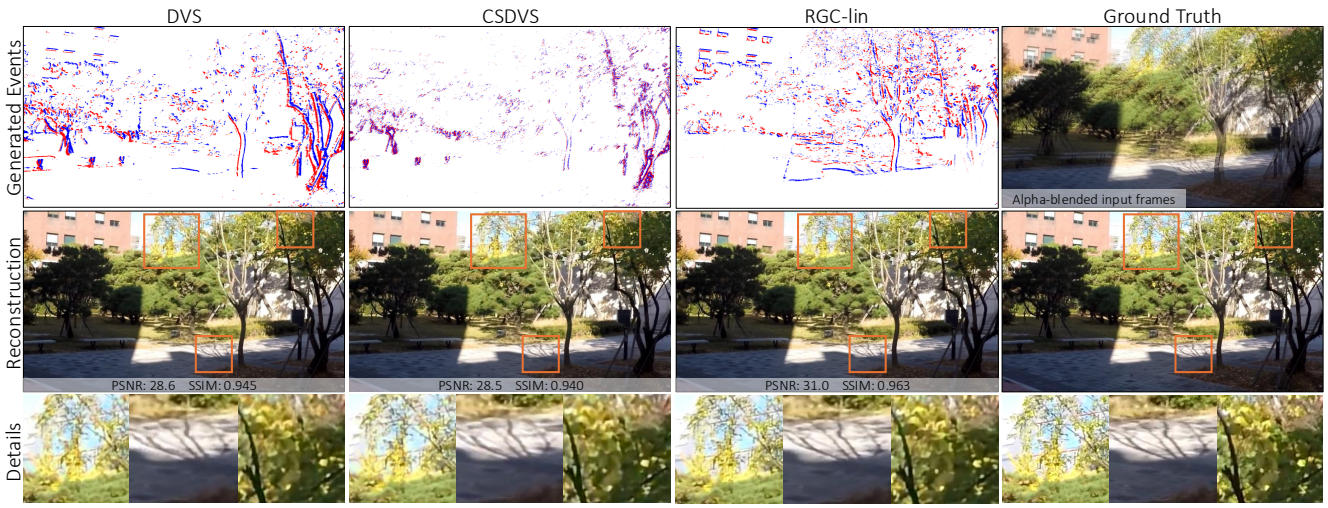

Figure 1. **Additional Video Interpolation Qualitative Results.** We compare reconstructions from DVS, CSDVS, and RGC-lin events. PSNR and SSIM metrics are included as well.

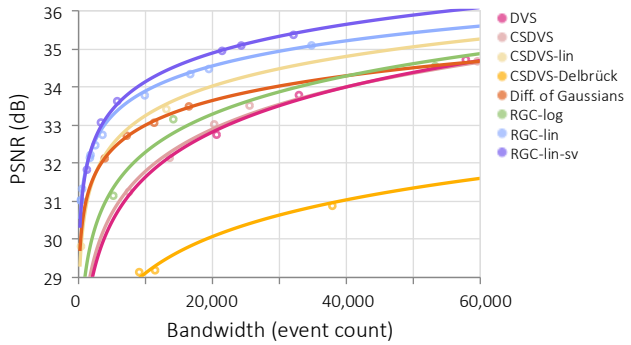

Figure 2. **Extended Comparisons for Video Interpolation.** Along with the comparisons in the main paper, we include the CSDVS- in the linear intensity and the difference of Gaussians.

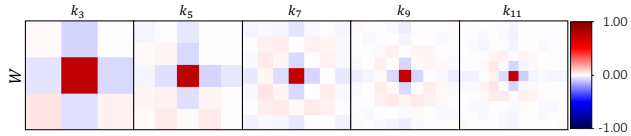

Figure 3. **Learned Kernels at Increasing Kernel Sizes.** As our eye's RGCs have varying receptive field sizes, we test the effect of increasing the kernel for the task of interpolation, given a set bandwidth of $\approx 20,000$ events per bin. As the kernel size increases, a subtle structure emerges where around the center pixel, it is negative, and then just further out is a ring of positive again before decreasing once more.

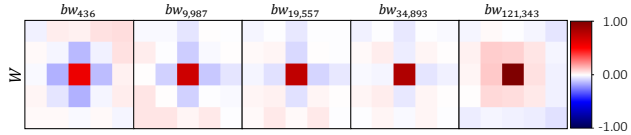

Figure 4. **Learned Kernels at Increasing Bandwidth.** We optimize performance while tuning the bandwidth jointly by varying the weighting of the sparsity loss during training. Here we show the learned 5×5 kernels for different bandwidths (bw).

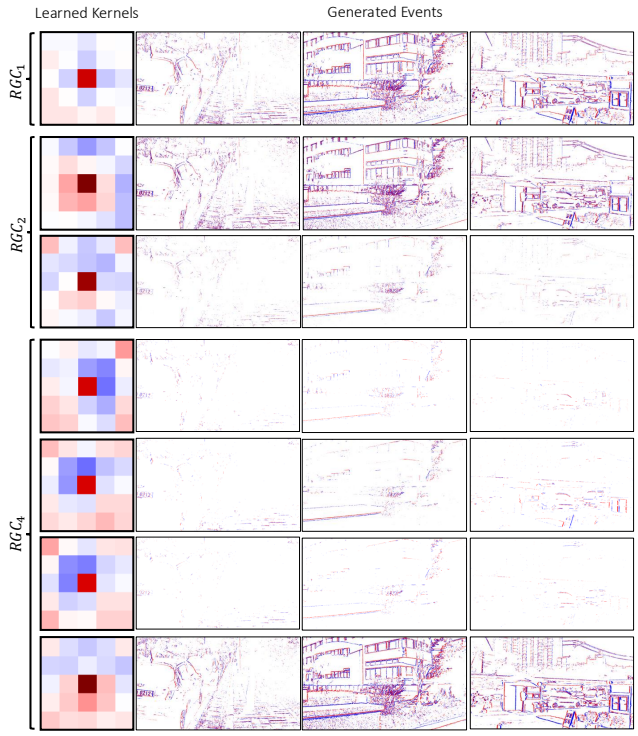

Figure 5. **Comparison between Learned Single and Multi-event.** We compare the learned kernels of a single RGC model, two RGC, and four RGC for interpolation and show the generated events for each. Each column is a different scene. At the same bandwidth, the 2-kernel model had a 0.40dB improvement over the single kernel, and the 4-kernel model had an additional 0.35dB over the 2-kernel model.

model. Even with half the average number of events, we achieve better optical flow.

Figure 6. **Multi-channel Comparison.** Above are the learned kernels for K_1 and K_{16} along with the events generated by the kernels for two scenes.

References

[1] Y Hu, S C Liu, and T Delbruck. v2e: From video frames to realistic DVS events. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2021. 1

[2] Siying Liu and Pier Luigi Dragotti. Sensing diversity and sparsity models for event generation and video reconstruction from events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–16, 2023. 1

[3] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 1

[4] 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. *arXiv preprint arXiv:2301.05191*, 2023. 1

[5] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of

Figure 7. **Interpolation Reconstructions from RGC_1, RGC_2, and RGC_4.** We show the middle reconstructed frame with PSNR and SSIM metrics.

visual slam. 2020. 1

[6] Yilun Wu, Federico Paredes-Vallés, and Guido C. H. E. de Croon. Lightweight event-based optical flow estimation via iterative deblurring. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'24)*, 2024. To Appear. 2

[7] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data dense pre-training, 2024. 1

Figure 8. **Interpolation Reconstructions from RGC_1 and RGC_{16}.** We show the middle reconstructed frame for the one learned kernel and 16 learned kernel models trained at the ultra-low bandwidth of roughly $10,000$ events.

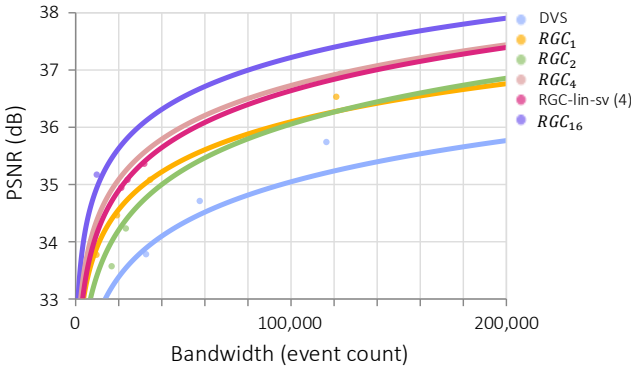

Figure 9. **Multi-channel RGCs for Video Interpolation.** We explore the idea of having mulitple learned retinal circuits per pixel, similar to the the human retina. $K1$ is a single kind of event, $K2$ is two kinds of events, and so on. In the main paper, we also had the option to learn spatially-varying kernels. This means there are 4 unique kernels, but each pixel only gets one circuit. We show this setting, RGC-lin-sv (4) as a comparison. K4 and RGC-lin-sv perform similarly. We also show DVS for reference.

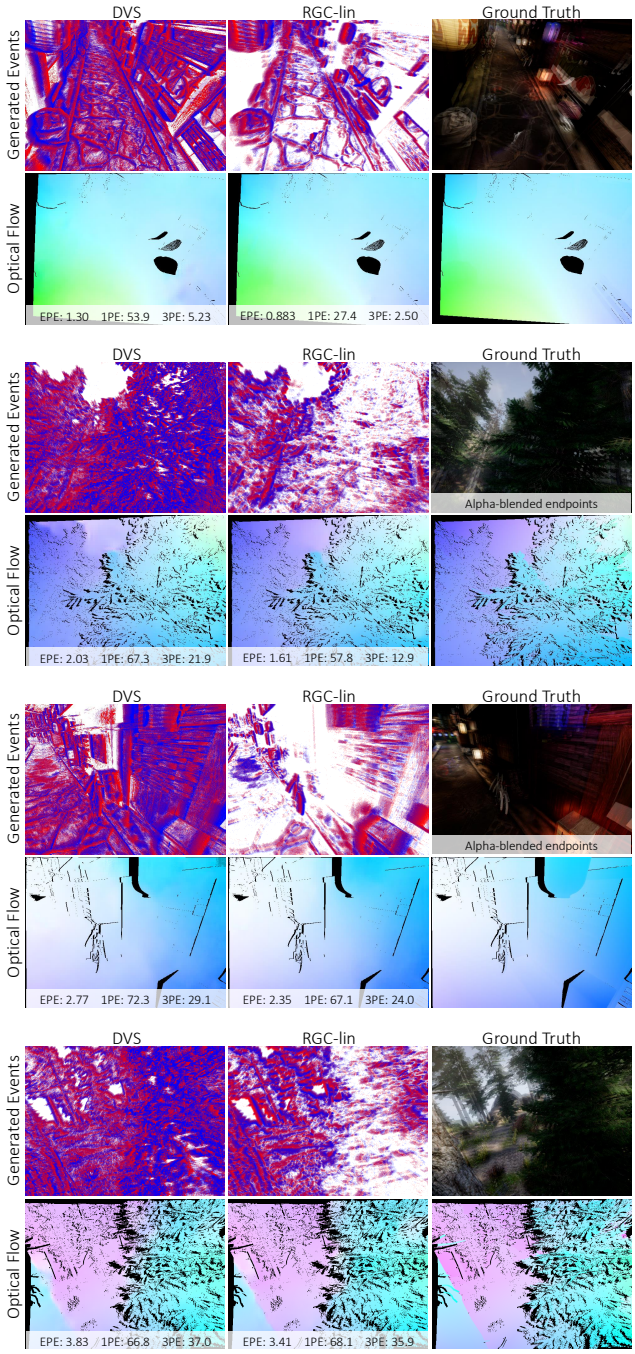

Figure 10. **Additional Optical Flow Qualitative Results.** We show EPE↓, 1PE↓, and 3PE↓ metrics for four additional scenes. With our learned RGC kernel, we achieve both sparser readout and better optical flow performance.