Event Stream Super-Resolution via Spatiotemporal Constraint Learning
Supplementary Material

1. Object Classification Experiment Details

Datasets. To demonstrate the accuracy and availability of the HR event streams generated by our method, we test the classification performance on the N-MNIST [8], CIFAR10-DVS [6], and ASL-DVS [1] datasets. The split of training and test sets of the classification task is the same way as the super-resolution task, *i.e.* we use the ground truth HR event streams from the training set of the super-resolution task as the training set in the classification task. The test sets of the classification task are the ground truth HR event streams in the test set of the super-resolution task (denoted as GT events), the HR event streams generated by our method on the test set of the super-resolution task (denoted as Ours) and those generated by Li *et al.* [5] (denoted as Li *et al.*), respectively.

Model and training procedure. We use the model proposed in [3] as the classification network, which contains a quantization module and a classifier. The quantization module takes the event stream as input and converts it into a voxel-based representation in a differentiable way. In practice, the quantization kernel is selected as the trilinear kernel and the quantized voxel is designed to contain 9 channels. The classifier takes the quantized voxel as input, extracts feature and classifies it. In practice, the classifier is selected as ResNet34 and pre-trained on ImageNet. The whole model is trained end-to-end by optimizing the cross-entropy loss for 30 epochs with a batch size of 16. The optimization method is Adam [4] with a learning rate of $1e-4$, multiplied by 0.5 every 10 epochs.

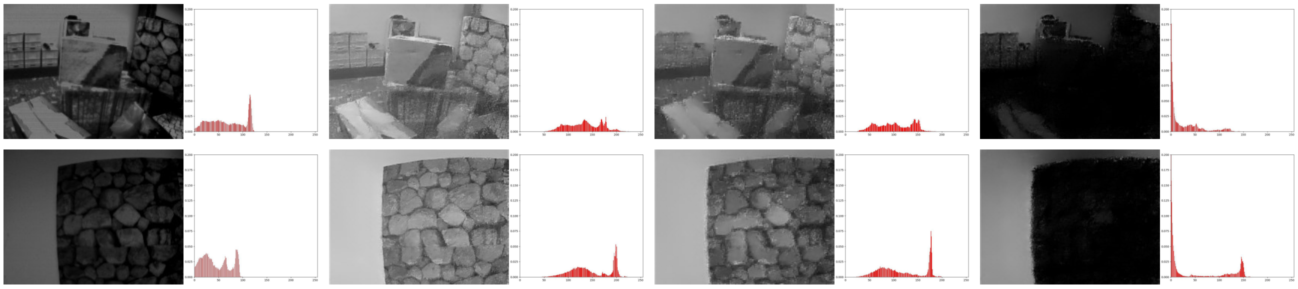
2. Image Reconstruction Experiment Details

Datasets. To further prove the practicability and robustness of the proposed method, we test it on a more challenging task, *i.e.* the image reconstruction task. The dataset we choose is the Event Camera Dataset [7], which contains 25 sequences captured by a DAVIS240 in real scenes. Each sequence contains event stream and corresponding ground truth gray-scale frames. Following the settings in [9], 7 sequences among them are chosen as the test set, *i.e.* dynamic_6dof, boxes_6dof, poster_6dof, shapes_6dof, office_zigzag, slider_depth, and calibration. Then, 11 sequences containing similar scenes from the remaining are selected as the training set, *i.e.* boxes_rotation, boxes_translation, office_spiral, slider_close, slider_far, dynamic_rotation, dynamic_translation, poster_translation, poster_rotation, shapes_rotation, and shapes_translation. For the event stream super-resolution task, we split each sequence into samples with a duration of 50 milliseconds without overlapping. In this way, the training set contains $10,050$ samples and the test set contains $6,269$ samples. For the evaluation of image reconstruction task, we choose the same ground truth frames with [9] from the 7 test sequences. The beginning and ending timestamps of the chosen frames are shown in Table 1.

Sequence	Begin time (s)	End time (s)
dynamic_6dof	5.0	20.0
boxes_6dof	5.0	20.0
poster_6dof	5.0	20.0
shapes_6dof	5.0	20.0
office_zigzag	5.0	12.0
slider_depth	1.0	2.5
calibration	5.0	20.0

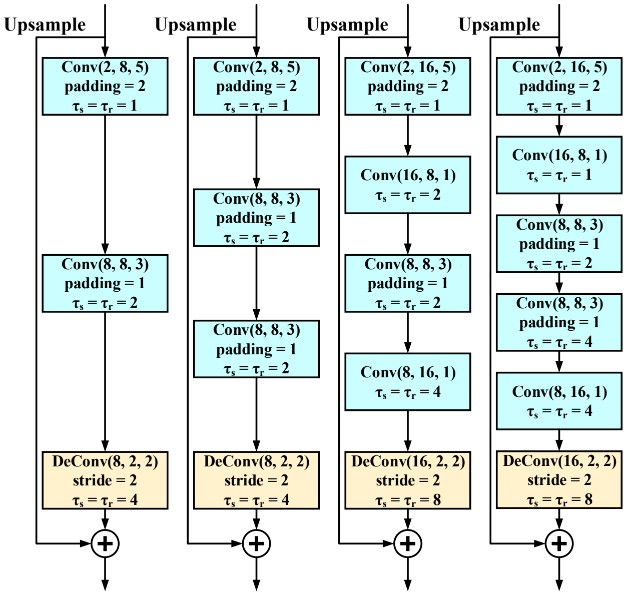
Table 1. The chosen frames for evaluation. For each sequence, we use all the frames between the begin timestamp and the end timestamp for evaluation.

Implementation details. The ground truth HR event streams, the HR event streams generated by our model and by Li *et al.* [5] on the test set are used to reconstruct images. In the image reconstruction task, we use the pre-trained model provided in [9]. To match the super-resolution results, we done the image reconstruction with a fixed time windows duration of 50ms. This is why our results are different from those reported in the original paper. For each ground truth frame, we following the setting in [9] to choose the reconstructed image with the closest timestamp (tolerance of ± 5 ms). The criteria of the image reconstruction task are mean square error (MSE, lower is better) and structural similarity (SSIM, higher is better). Before the evaluation, we apply histogram equalization to the output images and the ground truth images to make them more comparable. Note that this pre-process is favored for the results generated by [5], because our results are more balanced in



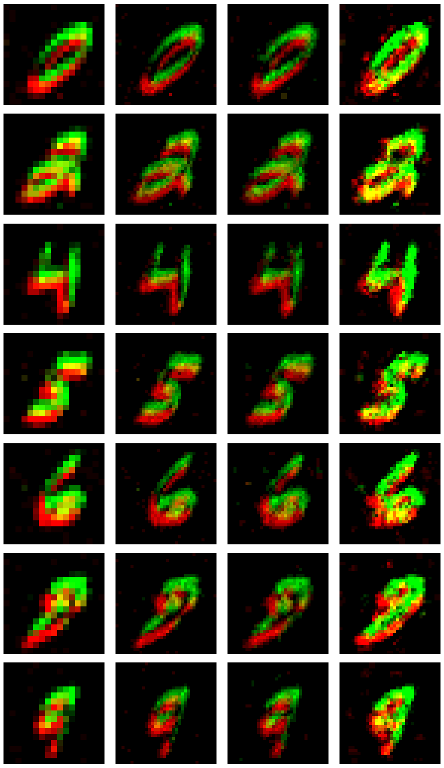
(a) Frame-GT **(b) Frame reconstructed by GT events** **(c) Frame reconstructed by ours events** **(d) Frame reconstructed by Li *et al.*'s events**

Figure 1. The reconstructed images and corresponding intensity histograms. From left to right: (a) Ground truth frames. (b) Images reconstructed from the HR ground truth event streams. (c) Images reconstructed from the HR event streams generated by our method. (d) Images reconstructed from the HR event streams generated by Li *et al.* [5]. It can be seen that images reconstructed from our event streams are more balanced in intensity.


Upsample
Conv(2, 8, 5) padding = 2 $\tau_s = \tau_r = 1$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 2$
DeConv(8, 2, 2) stride = 2 $\tau_s = \tau_r = 4$
Upsample
Conv(2, 8, 5) padding = 2 $\tau_s = \tau_r = 1$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 2$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 2$
DeConv(8, 2, 2) stride = 2 $\tau_s = \tau_r = 4$
Upsample
Conv(2, 16, 5) padding = 2 $\tau_s = \tau_r = 1$
Conv(16, 8, 1) $\tau_s = \tau_r = 2$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 2$
Conv(8, 16, 1) $\tau_s = \tau_r = 4$
DeConv(16, 2, 2) stride = 2 $\tau_s = \tau_r = 8$
Upsample
Conv(2, 16, 5) padding = 2 $\tau_s = \tau_r = 1$
Conv(16, 8, 1) $\tau_s = \tau_r = 1$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 2$
Conv(8, 8, 3) padding = 1 $\tau_s = \tau_r = 4$
Conv(8, 16, 1) $\tau_s = \tau_r = 4$
DeConv(16, 2, 2) stride = 2 $\tau_s = \tau_r = 8$

(a) 3 layers **(b) 4 layers** **(c) 5 layers** **(d) 6 layers**

Figure 2. The network architecture with different numbers of layers in the ablation study. We list the parameters of the convolutional and deconvolutional layers. The hyperparameters of the spiking neurons are also provided.



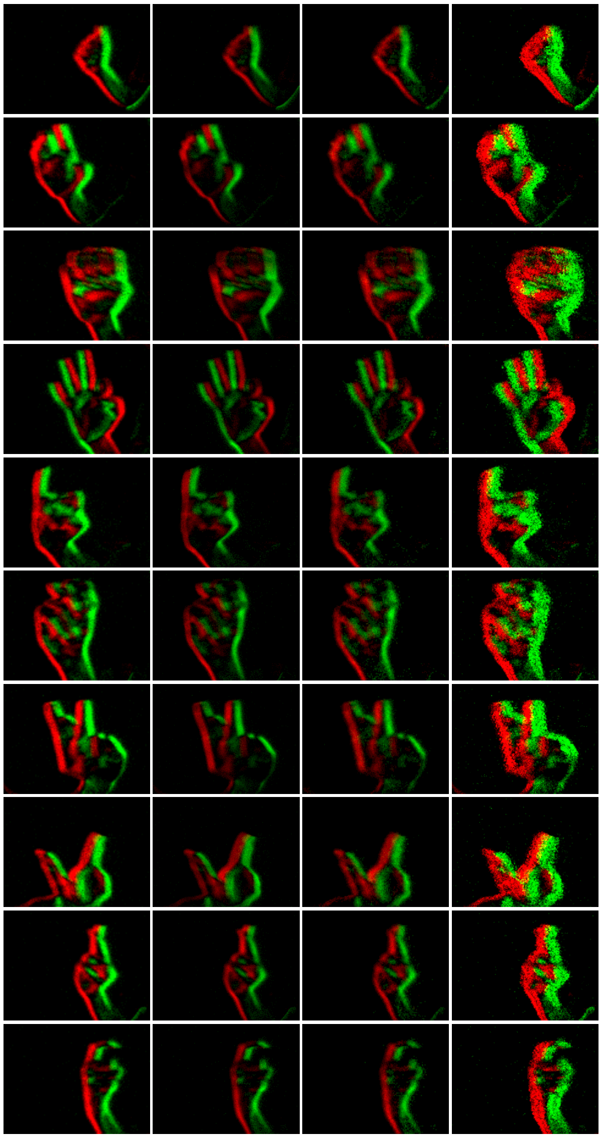
(a) LR (b) HR (c) Ours (d) Li *et al.*

Figure 3. Visualization results on N-MNIST dataset.

intensity, as shown in Figure 1.

3. Network Architecture Experiment Details

In this ablation study, we test our model with different numbers of layers and find that the performance decline as the model becomes deeper. Figure 2 shows the architecture of the models with different numbers of layers in the experiment. Column (a) is our final model, consisting of 2 convolutional layers and 1 deconvolutional layer, and the hyperparameters of the spiking neuron are $\tau_s = \tau_r = 1, 2, 4$, respectively. Then, we add a convolutional layer with hyperparameters of $\tau_s = \tau_r = 2$, as shown in column (b), to expand the respective field and extract features. Column (c) and (d) show models with 5 and 6 layers, respectively. Inspired by FSRCNN [2], we use two convolutional layers with a kernel size of 1 to shrink and expand the feature.

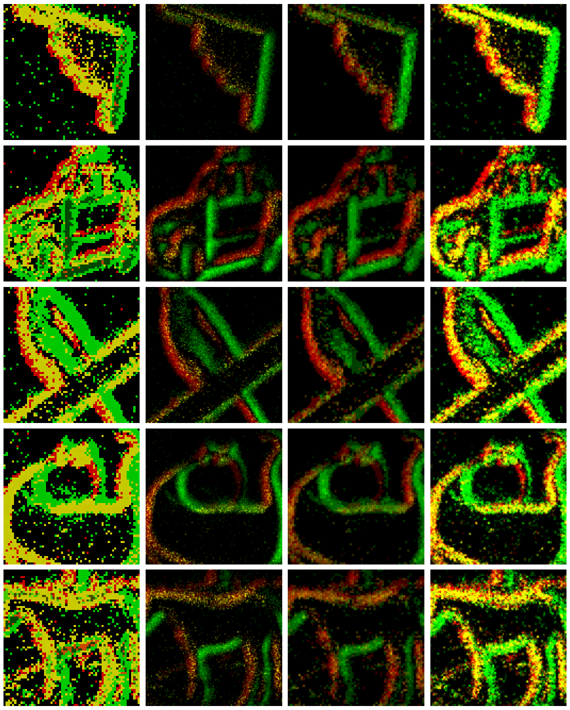


(a) LR **(b) HR** **(c) Ours** **(d) Li *et al.***

Figure 4. Visualization results on ASL-DVS dataset.

4. More Visualization Result

In this supplementary material, we provide more event stream super-resolution results on N-MNIST [8], CIFAR10-DVS [6], and ASL-DVS [1] datasets. We visualize the input LR event streams, the ground truth HR event streams, results generated by our method and Li *et al.* [5], respectively. Figure 3 shows the visualization results on the N-MNIST dataset. Figure 4 shows the visualization results on the ASL-DVS dataset, and Figure 5 shows the visualization results on the CIFAR10-DVS dataset. It can be seen that the proposed method could generate better HR event streams with a clearer boundary and less noise than [5].



(a) LR (b) HR (c) Ours (d) Li *et al.*

Figure 5. Visualization results on ASL-DVS dataset.

5. Embedded Implementation

In the supplementary material, we also provide a video to demonstrate the efficiency and usability of our method. In the provided video, the super-resolution result generated by the embedded implementation of our method in real scenarios is recorded. The result shows that our system can generate high-quality HR event streams in real-time, which proves the potential of our method for deployment on mobile systems, *e.g.* quadrotors and driverless cars.

References

[1] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-Based Object Classification for Neuromorphic Vision Sensing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 491–501, 2019. 1, 3

[2] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the Super-Resolution Convolutional Neural Network. In *Proceedings of the European Conference on Computer Vision*, pages 391–407. Springer, 2016. 2

[3] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-End Learning of Representations for Asynchronous Event-Based Data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5633–5643, 2019. 1

[4] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. 1

[5] Hongmin Li, Guoqi Li, and Luping Shi. Super-Resolution of Spatiotemporal Event-Stream Image. *Neurocomputing*, 335:206–214, 2019. 1, 2, 3

[6] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11:309, 2017. 1, 3

[7] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The Event-Camera Dataset and Simulator: Event-Based Data for Pose Estimation, Visual Odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1

[8] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. *Frontiers in Neuroscience*, 9:437, 2015. 1, 3

[9] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1