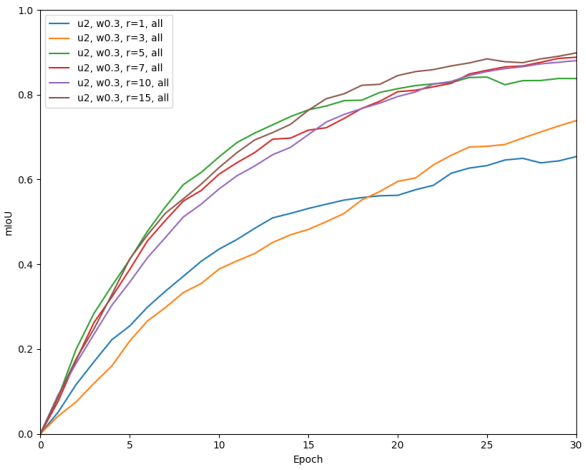
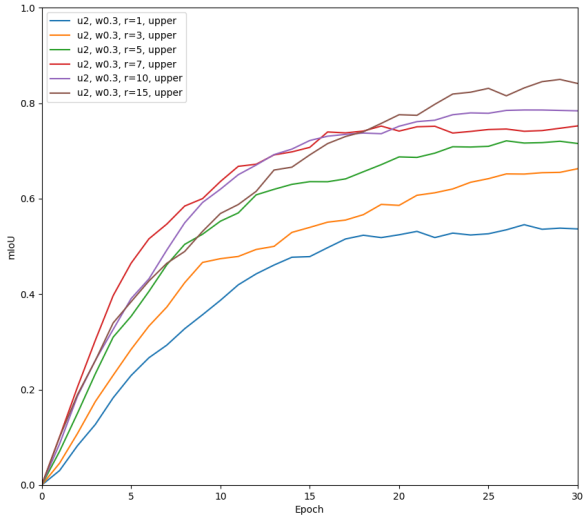
Learning Visual Motion Segmentation using Event Surfaces - Supplementary Material



(a)



(b)

Figure 1. *The effects of using edges computed within different radii r. (a) - all edges within a sphere are used; (b) - only edges in the upper hemisphere along time axis are used. We show the plots for the first 30 epochs on 'boxes' validation set, with subsampling $u = 2$ and slice width $w = 0.3$ sec.*

1. Edge Configurations - Ablation Study

Our main results were computed with 'upper hemisphere' edge configuration; the motivation to use only edges in the upper hemisphere is the ability to remove half the edges around each point, significantly reducing memory footprint on GPU. We perform additional ablation studies to compare how full (all edges in the sphere) and upper hemisphere edge configurations affect the quality of training for different values of r. All experiments use *boxes* validation set, and are trained on *boxes* train set, with $u = 2, w = 0.3$.

On the Fig. 1 (a) we show $mIoU$ scores for the first 30 epochs of training with the full edge configuration. Starting from $r = 5$ the results do not show significant improvements as r increases. In this setting, each point on the first layer aggregates features in a sphere with diameter $d = 2r$ - from 10 to 30 pixels for $r > 5$, and the network might reach saturation in the amount of locally available information. On the Fig. 1 (b) we perform similar experiments with our baseline upper hemisphere configuration. The results improve as r increases, but IoU is slightly lower than for the full configuration.

The experiments with $r = 1$ show the performance of the network with no spatial connections - since pixel coordinates are discretized, only connections along temporal axis can exist. We show additional side-by-side edge configuration comparisons for each radii on Fig. 3; the per-step runtime for these experiments is shown in Table 1.

2. Training Time

We show additional timing measurements in Table 1, with upper hemisphere and full configurations, for $u = 2, w = 0.3, r = 1..15$. Although slowdown is expected with full configuration, the main reason behind using only the upper hemisphere was high memory consumption with small u values and large r.

3. Additional Qualitative Results

3.1. Dataset

A sample of EVIMO dataset (*boxes* validation, sequence 0) is shown on Fig. 2. The points are colored according the

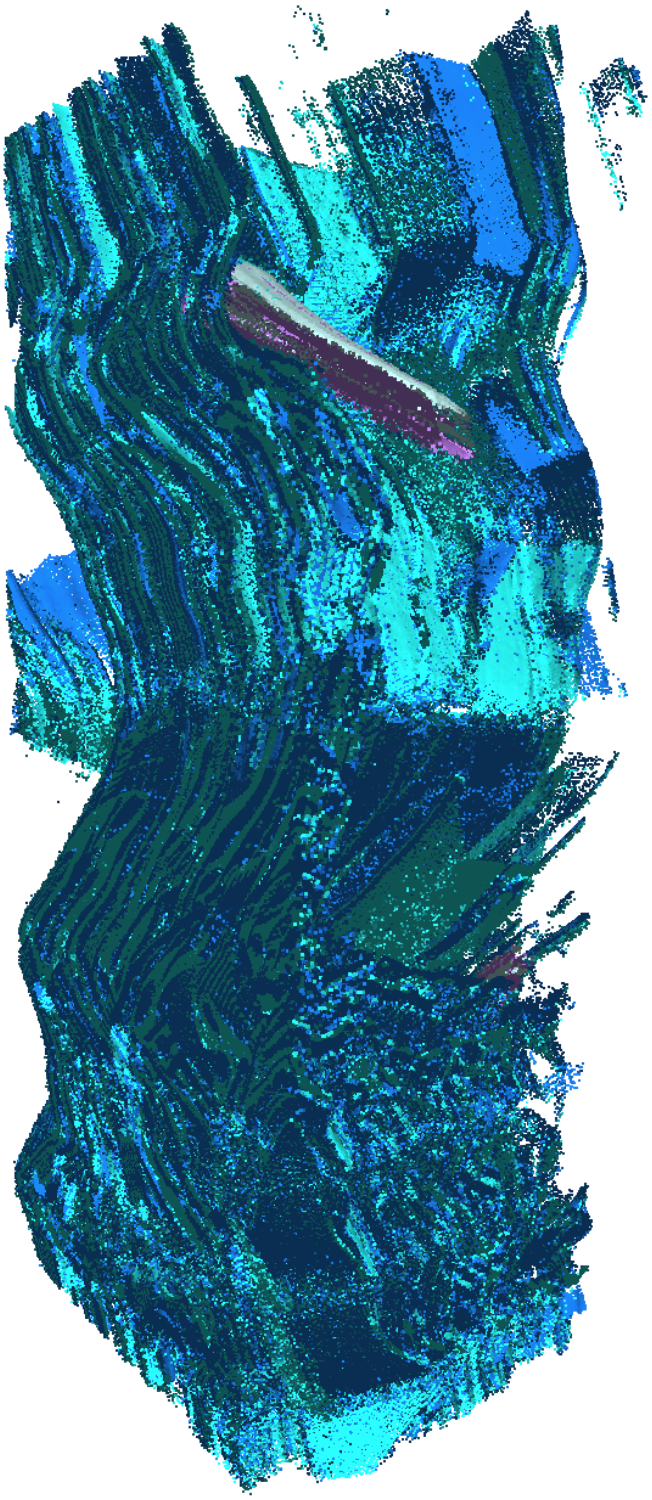


Figure 2. *A sample of the preprocessed EVIMO dataset, plotted in 3D (time axis is vertical). Red color channel encodes object id, and green encodes event polarity; a separately moving object is visible on the top part of the image in white and purple colors. The green and blue colors represent background.*

r	1	3	5	7	10	15
upper	0.293	0.415	0.636	0.748	0.975	1.340
full	0.267	0.669	0.858	0.984	1.282	1.751

Table 1. Average time per training step (forward and backward pass), in seconds, with batch size $b = 3$, subsampling $u = 2$, slice width $w = 0.3$ sec. for edge radii $r = 1$ to 15 pixels. $full$ corresponds to all edges in a sphere of radius r, $upper$ corresponds to edges only in the upper hemisphere along time axis. The results were collected using 3 Nvidia GTX 1080Ti GPUs.

corresponding event polarity in green channel, and object id in red channel. The background is shown in green and blue, and the object motion is visible on the upper part of the image, in white and purple. Note how the shape of the cloud follows scene motion.

3.2. Inference with Large w

We train a baseline neural network with $w = 0.3, r = 10, u = 2$ and apply it to the cloud with $w = 1.0$. The qualitative result is shown on Fig. 4 - the ground truth is on the left, and inference is on the right; the object is shown in blue color. We achieve similar $mIoU$ scores on larger slices as on the original ones.

3.3. Additional Qualitative Results

Fig. 5 shows 3D renderings on the segmentation results, on *boxes* validation set. For each figure, inference is on the left, ground truth is on the right; background shown in black. Most of the loss in IoU is due to lower recall, while precision is reasonably high. We attach the *.ply* models for these results together with this supplementary material.

3.4. Quality of Normal Estimation

Event surface normals are crucial features in our learning pipeline. We take a sample of EV-IMO dataset (box validation set, seq_00) to evaluate the quality and distribution of surface normal direction in both spatial and temporal domains. The histograms for a slice of $0.5s$. are presented on Fig. 6. The spatial quality of normals inevitably suffers from discretization artifacts caused by the limited resolution of the DAVIS346 sensor; the scene includes mostly horizontal and vertical edges, and the dominant direction is clearly seen on Fig. 6 (a). The temporal plot (b) is notably smoother, which is caused by the variation in scene depth and hence - normal flow magnitude, and also higher temporal resolution of the camera.

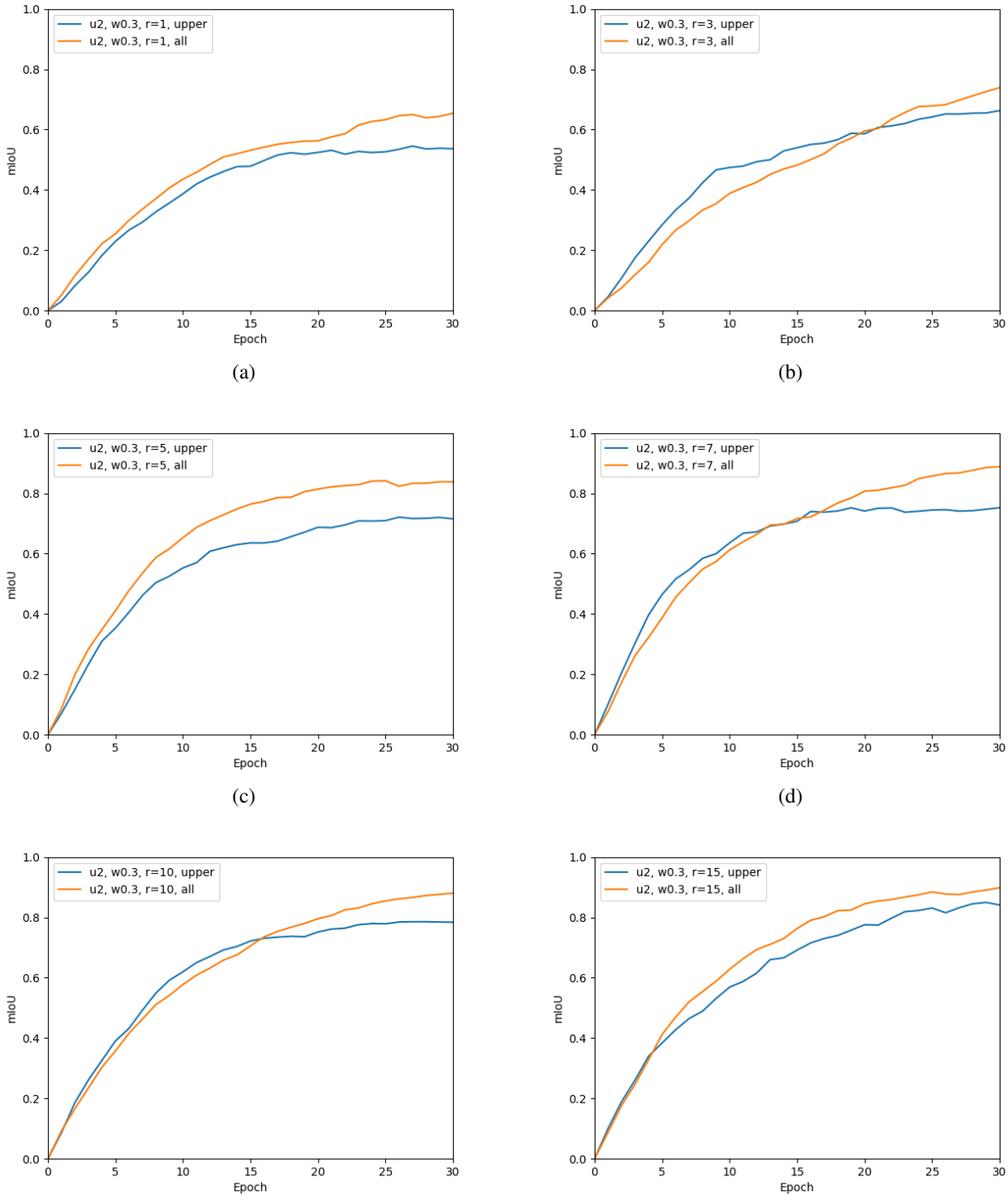


Figure 3. *Additional ablation study for full and upper hemisphere edge configurations and for different radii. For $r = 1$ pixels, most edges connect to events only along time axis. We show the plots for the first 30 epochs on 'boxes' validation set, with subsampling $u = 2$ and slice width $w = 0.3$ sec.*

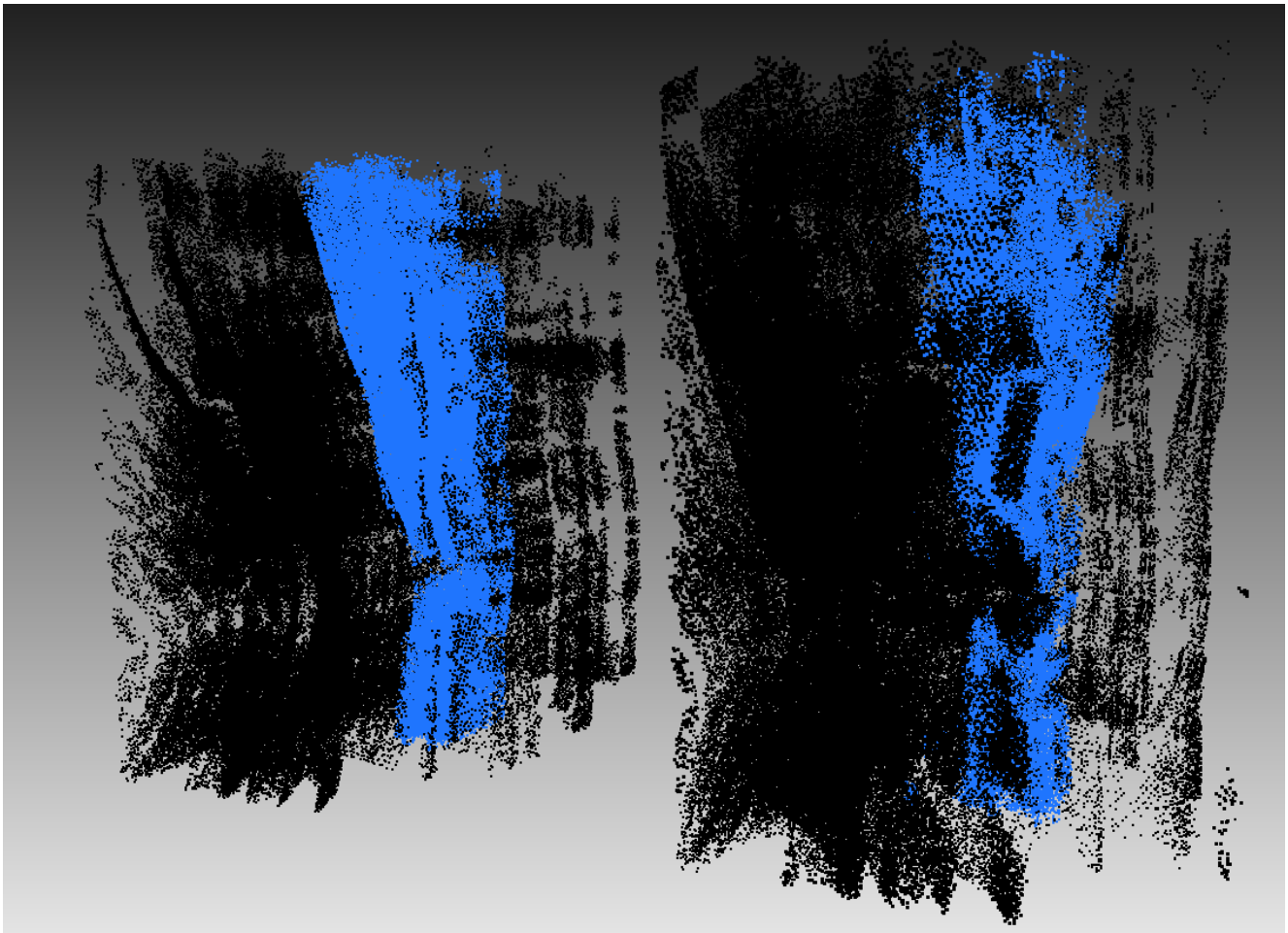


Figure 4. *Additional qualitative results: the network was trained on* 0.3 *sec. slices with* $r = 10, u = 2$ *and tested on* 1.0 *sec. slices to validate the invariance to slice width. Time axis is vertical; the object is in blue, black color is background. The left image represents ground truth, right is inference.*

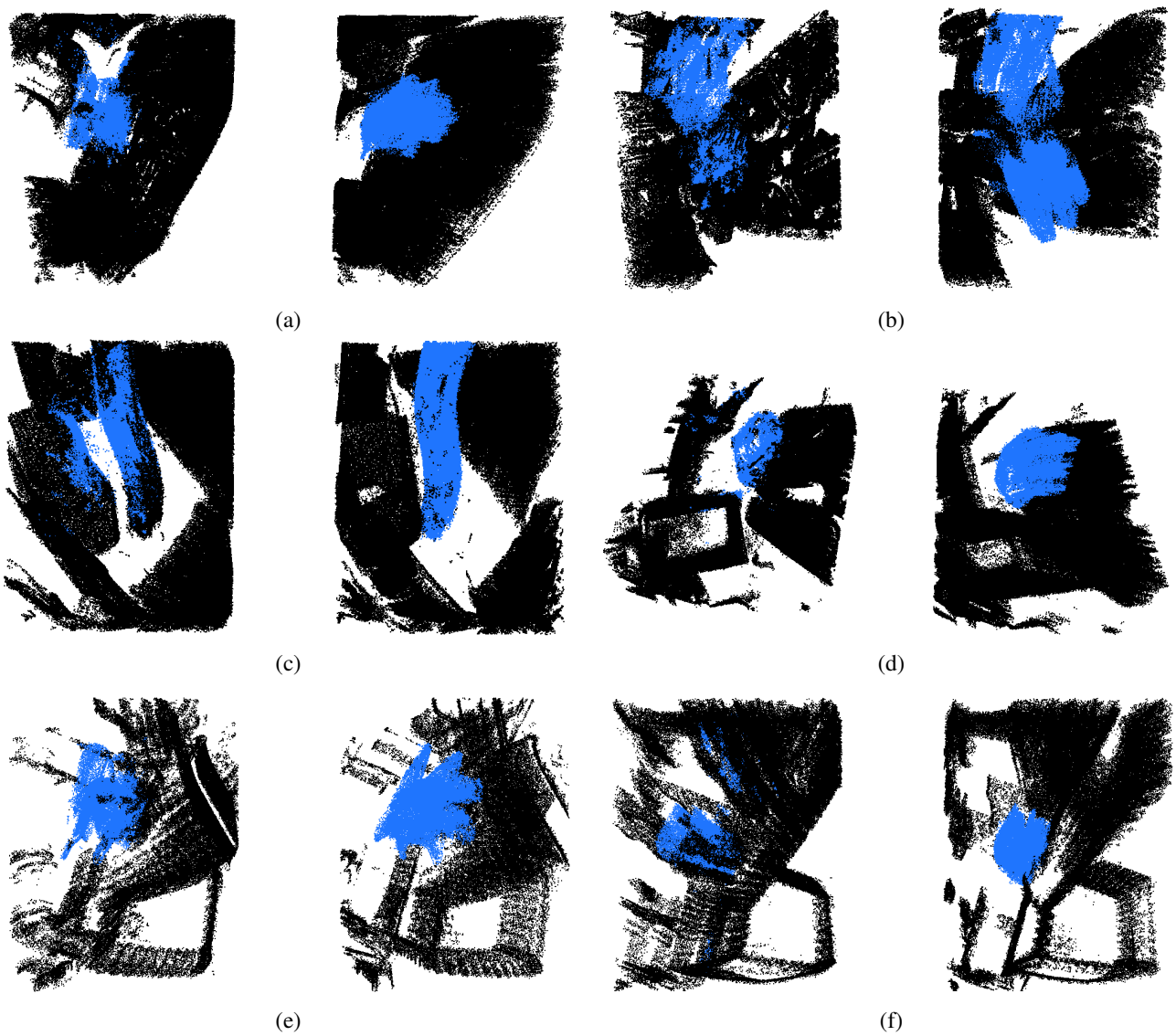
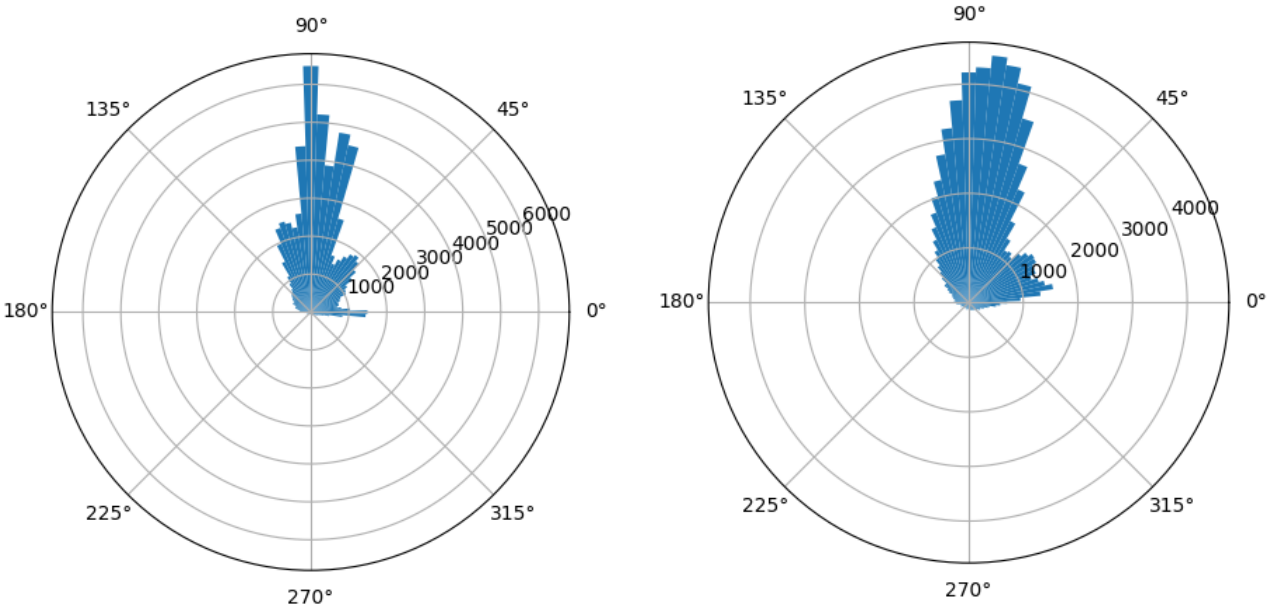


Figure 5. *Additional qualitative results: the network was trained with* $w = 0.3, r = 10, u = 2$*. For each figure: left is inference and right is ground truth (two figures per line are shown). Blue represents an object, black is background.*



(a) (b)

Figure 6. *Qualitative distribution of surface normals for 'box' validation dataset, sequence 0, for a 0.5 second event slice (featuring near constant velocity). (a) - distribution of normal direction in camera plane; (b) - distribution of normal direction in plane parallel to time axis and orthogonal to edge gradient.*