Supplementary Materials for Hierarchical Neural Memory Network for Low Latency Event Processing

The document explains 1) details of the window based multi-head cross-attention (W-MCA); 2) remaining details on the experimental setups; 3) details for hyperparameter tuning and 4) more detailed results of the experiments.

1. Details of the window based multi-head cross-attention (W-MCA)

The section explains the details of W-MCA used for "up-write" and "down-write" operations. We built W-MCA by extending the window based multi-head self-attention (W-MSA) [17] with a minor modification. We first explain W-MSA and then the modifications for our W-MCA.

Window based multi-head self-attention [17]. Given an input $\boldsymbol{X}$ with a size $(H \times W \times D)$, the W-MSA computes self-attention as follows:

$$\boldsymbol{H} = \text{W-MSA}(\boldsymbol{X}) \tag{1}$$

Below we describe the single-head operation for simplicity since the multi-head operation can be straightforwardly acquired by applying multiple single-head operations.

In the W-MSA, query, key, and value are first calculated by Layer Normalization (LN) [2] and three MLPs $(\mathcal{Q}, \mathcal{K}, \mathcal{V})$.

$$\hat{\boldsymbol{X}} = \text{LN}(\boldsymbol{X}) \tag{2}$$

$$\boldsymbol{Q} = \mathcal{Q}(\hat{\boldsymbol{X}}), \quad \boldsymbol{K} = \mathcal{K}(\hat{\boldsymbol{X}}), \quad \boldsymbol{V} = \mathcal{V}(\hat{\boldsymbol{X}}) \tag{3}$$

The query, key, and value are then divided into tiles, each with a size (7×7):

$$\boldsymbol{q}_n \in \mathcal{T}_{7\times7}(\boldsymbol{Q}), \quad \boldsymbol{k}_n \in \mathcal{T}_{7\times7}(\boldsymbol{K}), \quad \boldsymbol{v}_n \in \mathcal{T}_{7\times7}(\boldsymbol{V}) \tag{4}$$

where $\mathcal{T}_{7\times7}$ is a function for the tile division, and $\boldsymbol{q}_n, \boldsymbol{k}_n, \boldsymbol{v}_n$ are query, key, and value inside the n-th tile with a size $(49 \times D)$. Then, the attention is calculated inside each tile as follows:

$$\boldsymbol{h}_n = \text{softmax}(\boldsymbol{q}_n \boldsymbol{k}_n^T / \sqrt{D} + \boldsymbol{B})\boldsymbol{v}_n \tag{5}$$

where $\boldsymbol{B}$ is a relative position bias. Finally, the outputs $\{\boldsymbol{h}_1, ..., \boldsymbol{h}_N\}$ from all the N tiles are joined back to the original spatial size by the inverse function of the tile division:

$$\boldsymbol{H} = \mathcal{T}_{7\times7}^{-1}(\boldsymbol{h}_1, ..., \boldsymbol{h}_N) \tag{6}$$

Window based multi-head cross-attention. We build the window based multi-head cross-attention (W-MCA) to aggregate information from other memory states. W-MCA is based on the W-MSA [17] and the only difference is that we change the self-attention to the cross-attention. Specifically, we modified Eq. 1, Eq. 2, and Eq. 3 to have two features $\boldsymbol{X}_1, \boldsymbol{X}_2$ as inputs:

$$\boldsymbol{H} = \text{W-MCA}(\boldsymbol{X}_1, \boldsymbol{X}_2) \tag{7}$$

$$\hat{\boldsymbol{X}}_1 = \text{LN}(\boldsymbol{X}_1), \quad \hat{\boldsymbol{X}}_2 = \text{LN}(\boldsymbol{X}_2) \tag{8}$$

$$\boldsymbol{Q} = \mathcal{Q}(\hat{\boldsymbol{X}}_1), \quad \boldsymbol{K} = \mathcal{K}(\hat{\boldsymbol{X}}_2), \quad \boldsymbol{V} = \mathcal{V}(\hat{\boldsymbol{X}}_2) \tag{9}$$

The calculations after getting $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the same as W-MSA.

2. Setups for semantic segmentation

Dataset. The experiments are conducted on DSEC-Semantic dataset [23]. The dataset is a subset of DSEC dataset [9] that consists of event camera data and RGB frames recorded at the street scene. The resolution of the event camera and the RGB camera is 640×480 pixels and 1440×1080 pixels, respectively. For event-image fusion, we resized the RGB images to match the resolution of event data. Note that the cameras have different viewpoints, and the RGB frames are not perfectly aligned with the event data. The dataset contains pixel-wise annotations automatically generated from RGB images at 20Hz. In total, 8,082 and 2,809 frames are available for training and testing. Following [1], we used 11 classes for the experiments.

Task head. We used the decoder architecture of UPerNet [25] as our task head. For HMNet, we added bottom-up feature fusion in the task head for refreshing the high-level features with the up-to-date low-level features. We also omitted the Pyramid Pooling Module [26] of UPerNet.

Training. Table 1 shows the hyperparameters for training. The HMNet models and the baselines are trained for 90k and 120k iterations, respectively. We trained the recurrent baselines for 500 iterations using Truncated Backpropagation Through Time [24], with a sequence length of 5.0sec. We did not conduct the additional training on HMNet since it did not improve the accuracy. We used AdamW [18] as

Table 1. List of hyperparameters used in the experiments. For all the experiments we used initial learning rate of 2.0e-4 with cosine learning rate decay and batch size of 16. In the resize augmentation, the resizing factor is randomly selected from range $[0.5, 2.0]$.

Dataset	Model	Training settings								Additional training	
		Input size	Event repr.	Time step size	Sequence length	Data aug.	Train iter	Optimizer	Weight decay	Sequence length	Train iter
DSEC-Semantic	Baseline	640×440	Time Surface	50ms	50ms	resize, crop, flip	120k	AdamW	1.0e-2	-	-
	Baseline GRU	640×440	Time Surface	50ms	500ms (10 steps)	resize, crop, flip	120k	AdamW	1.0e-2	5.0sec (100 steps)	0.5k
	HMNet	640×440	ESCA	5ms	200ms (40 steps)	resize, crop, flip	90k	AdamW	1.0e-2	-	-
GEN1 dataset	Baseline	304×260	Time Surface	200ms	200ms	resize, crop, flip	270k	Adam	5.0e-4	-	-
	Baseline GRU	304×260	Time Surface	50ms	500ms (10 steps)	resize, crop, flip	135k	Adam	5.0e-4	5.0sec (100 steps)	4.5k
	HMNet	304×260	ESCA	5ms	200ms (40 steps)	resize, crop, flip	400k	AdamW	1.0e-2	8.1sec (1620 steps)	4.5k
Eventscape	Baseline	512×256	Time Surface	200ms	200ms	flip	135k	Adam	1.0e-4	-	-
	Baseline GRU	512×256	Time Surface	50ms	500ms (10 steps)	flip	135k	Adam	1.0e-4	-	-
	HMNet	512×256	ESCA	5ms	200ms (40 steps)	flip	135k	AdamW	1.0e-2	-	-
MVSEC	Baseline	346×260	Time Surface	200ms	200ms	flip	5.6k	Adam	1.0e-4	-	-
	Baseline GRU	346×260	Time Surface	50ms	500ms (10 steps)	flip	5.6k	Adam	1.0e-4	-	-
	HMNet	346×260	ESCA	5ms	200ms (40 steps)	flip	5.6k	AdamW	1.0e-2	-	-

Table 2. Results of hyperparameter tuning.

Parameter	Search space	Tuning result	
		Manual	Automatic
Event repr.	Time Surface, VoxelGrid(hist), VoxelGrid(time)	Time Surface	Time Surface
Time step size	50ms, 200ms	200ms	200ms
Optimizer	SGD, Adam, AdamW	Adam	AdamW
Weigth decay	5.0e-2, 1.0e-2, 5.0e-4, 1.0e-4	5.0e-4	5.0e-2
LR scheduler	linear, linear with warmup, cosine, cosine with warmup	cosine	cosine
Resize range	$[0.5, 1.0]$, $[0.5, 1.5]$, $[0.5, 2.0]$, $[1.0, 1.5]$, $[1.0, 2.0]$	$[0.5, 2.0]$	$[0.5, 1.5]$

Table 3. Constants used for Eventscape and MVSEC datasets

	d_{max}	α
Eventscape	1000	5.7
MVSEC	80	3.7

Table 4. Numerical values of the results in Fig.5 of the main paper (DSEC-Semantic dataset). The latency in brackets is measured using multi-GPU (one GPU per latent memory).

	Backbone	Decoder	Recurrent	Accuracy	mIoU	Latency [ms]
EV-SegNet [1]	Xception	UNet		88.6	51.8	-
ESS [23]	E2Vid	UNet	✓	89.4	53.3	16.2
Baseline	ResNet-18	UPerNet		90.3	53.4	13.2
	ResNet-50	UPerNet		90.4	54.1	19.0
	ConNeXt-Tiny	UPerNet		90.0	52.8	19.3
	Swin-Tiny	UPerNet		90.2	53.4	30.1
Baseline-GRU	ResNet-18	UPerNet	✓	90.5	53.6	18.2
	ConNeXt-Tiny	UPerNet	✓	90.8	54.0	21.8
	Swin-Tiny	UPerNet	✓	90.7	54.0	25.7
HMNet (ours)	HMNet-B1	UPerNet	✓	88.7	51.2	7.0
	HMNet-L1	UPerNet	✓	89.8	55.0	10.5
	HMNet-B3	UPerNet	✓	89.5	53.9	9.7 (8.0)
	HMNet-L3	UPerNet	✓	90.9	57.1	13.9 (11.9)

Table 5. Numerical values of the results in Fig.6 of the main paper (GEN1 dataset). The latency in brackets is measured using multi-GPU (one GPU per latent memory).

	Backbone	Head	Recurrent	mAP	Latency [ms]
MatrixLSTM [4]	DarkNet53	YOLOv3		31.0	-
NGA [11]	DarkNet53	YOLOv3		35.9†	-
RED [20]	ConvLSTM	SSD	✓	40.0	11.6
Asynet [19]	Sparse-Conv	YOLO	✓	12.9	-
AEGNN [22]	GNN	YOLO	✓	16.3	-
AED [16]	DarkNet-21	YOLOX		45.4	13.1
ASTMNet [13]	Rec-Conv	SSD	✓	46.7	35.6*
Baseline	CSPDarknet-53	YOLOX-Lite		45.3	16.3
	ResNet-50	YOLOX-Lite		44.7	14.6
	ConNeXt-Tiny	YOLOX-Lite		40.2	12.5
	Swin-Tiny	YOLOX-Lite		39.7	22.4
Baseline-GRU	CSPDarknet-53	YOLOX-Lite	✓	48.2	14.8
	ResNet-50	YOLOX-Lite	✓	46.6	23.3
	ConNeXt-Tiny	YOLOX-Lite	✓	45.0	11.9
	Swin-Tiny	YOLOX-Lite	✓	44.3	18.5
HMNet (ours)	HMNet-B1	YOLOX-Lite	✓	45.5	4.6
	HMNet-L1	YOLOX-Lite	✓	47.0	5.6
	HMNet-B3	YOLOX-Lite	✓	45.2	7.0 (5.9)
	HMNet-L3	YOLOX-Lite	✓	47.1	7.9 (7.0)

* The latency value is borrowed from [13], where they used Titan Xp GPU.
† The result of NGA [11] is borrowed from [16].

the optimizer with a large weight decay coefficient of 0.01 as it performed better than Adam [12] on the task. We used a cross-entropy loss for our loss function. We also appended auxiliary loss [26] on the output feature of z_3 for HMNet and stage3 for baselines.

3. Setups for object detection

Dataset. The experiments are conducted on GEN1 dataset [20]. GEN1 dataset is a dataset for detecting objects from event cameras mounted on vehicles. The dataset includes 2,358 event sequences; each has a length of 60 sec and a resolution of 304×240 pixels. The sequences are divided into 1,459, 429, and 470 for training, validation, and testing. The bounding box annotations are available at 1Hz to 4Hz, depending on the sequence. The labels are defined for two classes: pedestrian and car.

Task head. We built a lightweight detection head based on YOLOX [7]. Specifically, we replaced PAFPN in YOLOX with FPN [15] and added bottom-up feature fusion before top-down fusion of the FPN.

Training. On the dataset, the proposed models, the baselines, and the recurrent baselines are trained for 400k, 270k, and 135k iterations, respectively. Training more iterations

Table 6. Pre-training results on Eventscape dataset.

	Backbone	Input	$\delta 1 \uparrow$	$\delta 2 \uparrow$	$\delta 3 \uparrow$	REL$\downarrow$	RMS$\downarrow$	RMSlog$\downarrow$	Latency [ms]
E2Depth	ConvLSTM	event	0.801	0.890	0.943	0.320	92.520	0.390	10.2
RAMNet	ConvGRU	event+RGB	0.787	0.888	0.944	0.200	76.124	0.368	12.0
Baseline	ResNet-18	event	0.749	0.860	0.921	0.602	108.057	0.461	10.7
	ResNet-18	event+RGB	0.774	0.883	0.935	0.251	78.049	0.375	10.3
	ResNet-50	event	0.751	0.858	0.920	0.667	110.409	0.471	18.0
	ResNet-50	event+RGB	0.781	0.887	0.940	0.246	78.683	0.366	18.2
Baseline-GRU	ResNet-18	event	0.720	0.815	0.876	0.991	130.329	0.580	9.4
	ResNet-18	event+RGB	0.763	0.868	0.926	0.272	79.011	0.387	9.5
HMNet (ours)	HMNet-B1	event	0.729	0.827	0.894	0.686	120.928	0.535	3.3
	HMNet-L1	event	0.754	0.849	0.912	0.586	112.490	0.483	4.3
	HMNet-B3	event	0.770	0.871	0.929	0.623	109.451	0.450	5.8 (4.9)
	HMNet-B3	event+RGB	0.772	0.874	0.934	0.488	90.304	0.400	6.6 (4.9)
	HMNet-L3	event	0.484	0.714	0.858	0.755	104.193	0.571	7.5 (6.5)
	HMNet-L3	event+RGB	0.787	0.883	0.939	0.484	91.547	0.392	7.8 (6.5)

did not improve the performance of the baselines. As the labels have a low frame rate, the recurrent models require further training using a longer sequence. In this additional training, we trained HMNet and the recurrent baselines for 4.5k iterations using Truncated Backpropagation Through Time [24], with a sequence length of 8.1sec/5.0sec, respectively.

4. Setups for depth estimation

Dataset. Following Gehrig *et al*. [8], we pretrained our models on the synthetic Eventscape dataset [8]. We then fine-tuned and evaluated the models on real MVSEC dataset [27]. Eventscape dataset consists of synthetic street-scene data generated by CARLA simulator [5]. The dataset includes event data and RGB frames with a resolution of 512×256 pixels. The ground truth depth is generated by the simulator at 25Hz, resulting in 122k, 22k, and 26k frames for training, validation, and testing.

MVSEC dataset consists of event data and gray-scale images recorded by a DAVIS event camera with a resolution of 346×260 pixels, mounted on a driving car. Since the DAVIS camera is coaxial, events and gray-scale images are aligned initially. The ground truth depth map is recorded at 20Hz using a LiDAR sensor. The dataset includes several sequences recorded during daytime and night-time. We used *outdoor day2* for training and *outdoor day1* and *outdoor night1* for evaluation. The gray-scale images are recorded at 45Hz for the daytime sequence and 10Hz for the night-time sequence.

Task head. We built the task head of the baselines based on the decoder architecture of UNet [21]. Specifically, the task head applies six residual blocks on the output feature from the backbone's stage4. The task head then applies three bilinear upsampling layers, each followed by a concatenation of the skipped features from each stage and two residual blocks. Finally, the task head applies a conv3 $\times$ 3 with a BatchNorm and a ReLU, conv1 $\times$ 1, and a sigmoid function. The task head for HMNet has similar architecture to the baseline, but the FPN architecture replaces the UNet-like decoder part. Similar to other tasks, we added bottom-up feature fusion to the FPN.

Training. Following the previous works [8, 10], we trained the model to predict normalized log depth $\hat{d}$:

$$\hat{d} = \frac{1}{\alpha} \log \frac{d}{d_{\text{max}}} + 1 \tag{10}$$

where d is a metric depth, d_{max} is a maximum depth in a dataset, and α is a constant determined by a ratio between a maximum depth d_{max} and a minimum depth d_{min}.

$$\alpha = \log \frac{d_{\text{max}}}{d_{\text{min}}} \tag{11}$$

Table 3 shows the specific value of the constants d_{max} and α for each dataset.

We trained our models with the same loss function as the previous work [8]. Specifically, we used the scale-invariant loss [6] and the multi-scale scale-invariant gradient matching loss [14] for our loss function. We set a weight for the gradient matching loss as 0.25.

5. Details of hyperparameter tuning

Table 2 shows the search space and the result of the hyperparameter tuning on GEN1 dataset. The automatic tuning is conducted using Hyperopt [3] with 36 iterations. Hyperopt finds a similar configuration with manual tuning.

6. Detailed results

Tabels 4 and 5 report the numerical values of the results shown in Fig.5 and Fig.6 in the main paper. Figures 1, 2, 3, 4, and 5 show the additional qualitative samples.

Table 6 shows the pre-training results on the synthetic Eventscape dataset. While HMNet and baselines perform better than previous methods on real MVSEC dataset

(shown in the main paper), they perform worse in the pre-training phase. One reason is that we apply bilinear upsampling on the model prediction instead of the convolutional upsampling used in the previous works. We find that the convolutional upsampling impairs the performance on the real MVSEC dataset while it improves accuracy on the Eventscape dataset. Another reason might be the low temporal resolution of synthetic event data. The synthetic data has the temporal resolution of millisecond order since the data is generated based on 500Hz image frames, which might be insufficient for HMNet that works at a high operation rate (*i.e*. 200Hz), leading to poor performance.

References

[1] Inigo Alonso and Ana C. Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *CVPRW*, 2019. 1, 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *CoRR*, 2016. 1

[3] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *ICML*, 2013. 3

[4] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *ECCV*, 2020. 2

[5] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017. 3

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 3

[7] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. In *CoRR*, 2021. 2

[8] Daniel Gehrig, Michelle Ruegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6, 2021. 3

[9] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6, 2021. 1

[10] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *3DV*, 2020. 3

[11] Yuhuang Hu, Tobi Delbruck, and Shih Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *ECCV*, 2020. 2

[12] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[13] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31, 2022. 2

[14] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 3

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[16] Bingde Liu. Motion robust high-speed light-weighted object detection with event camera. In *CoRR*, 2022. 2

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[19] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *ECCV*, 2020. 2

[20] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. In *NeurIPS*, 2020. 2

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[22] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *CVPR*, 2022. 2

[23] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. Ess: Learning event-based semantic segmentation from still images. In *ECCV*, 2022. 1, 2

[24] Ronald J. Williams and David Zipser. *Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity*, page 433–486. L. Erlbaum Associates Inc., 1995. 1, 3

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 1

[26] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2

[27] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multi-vehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3, 2018. 3

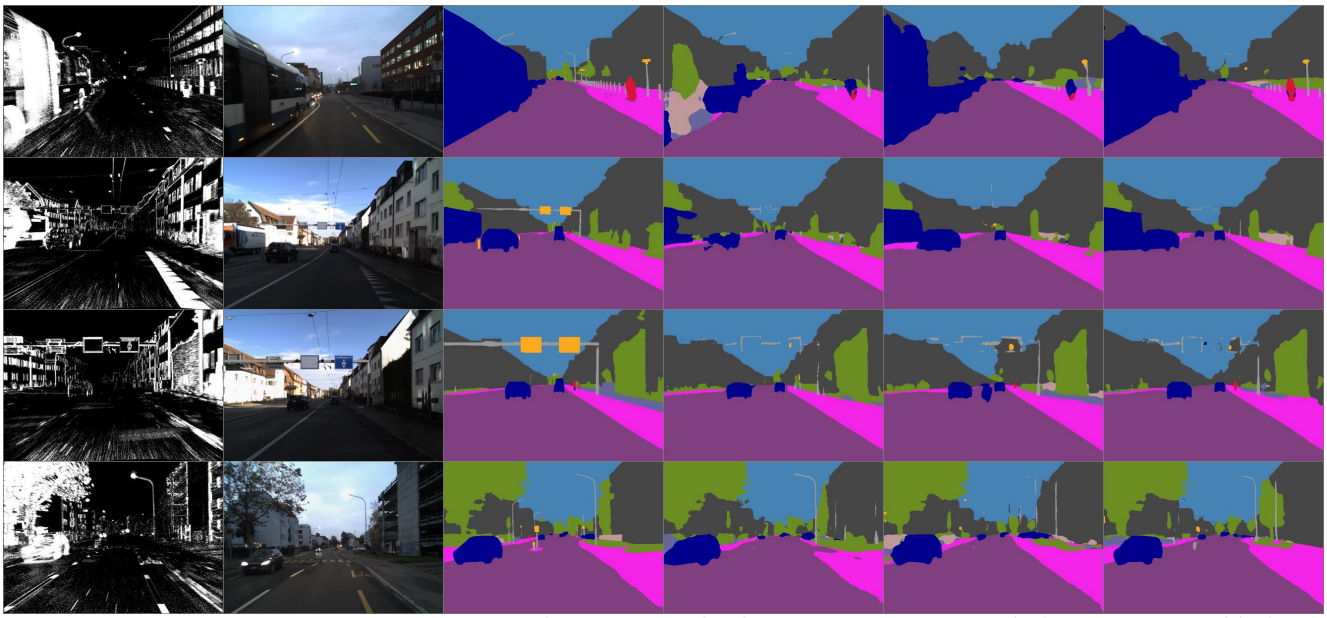

Figure 1. Qualitative results on DSEC-Semantic dataset.

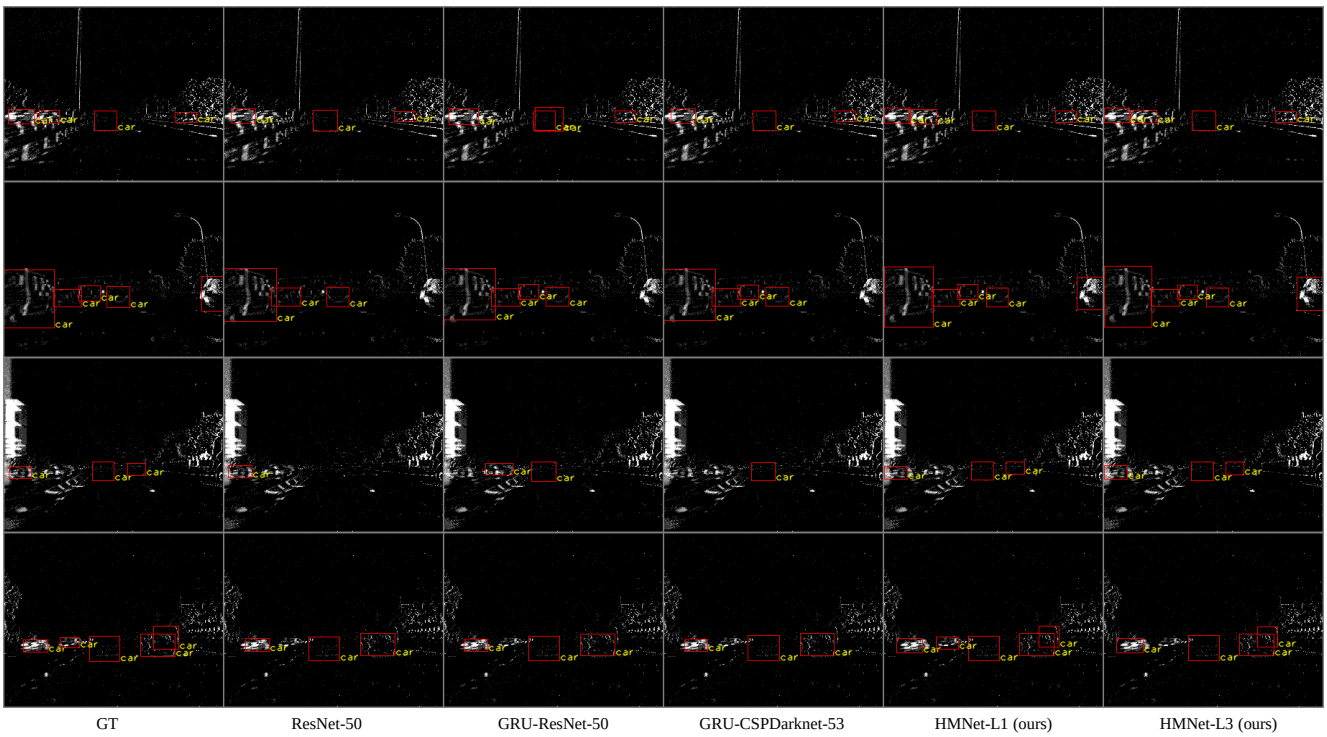

Figure 2. Qualitative results on GEN1 dataset. HMNet-L1 performs better than HMNet-L3 in some cases (4-th row)

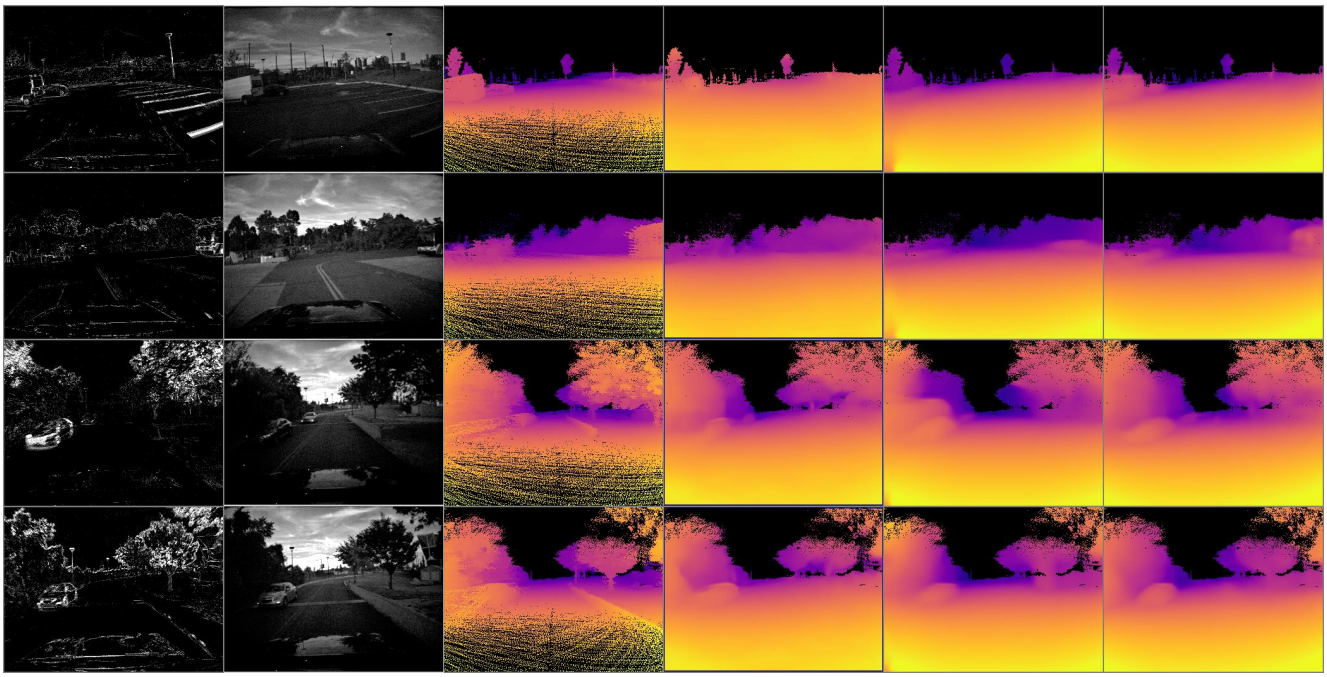

Figure 3. Qualitative results on *outdoor day1* sequence in MVSEC dataset.

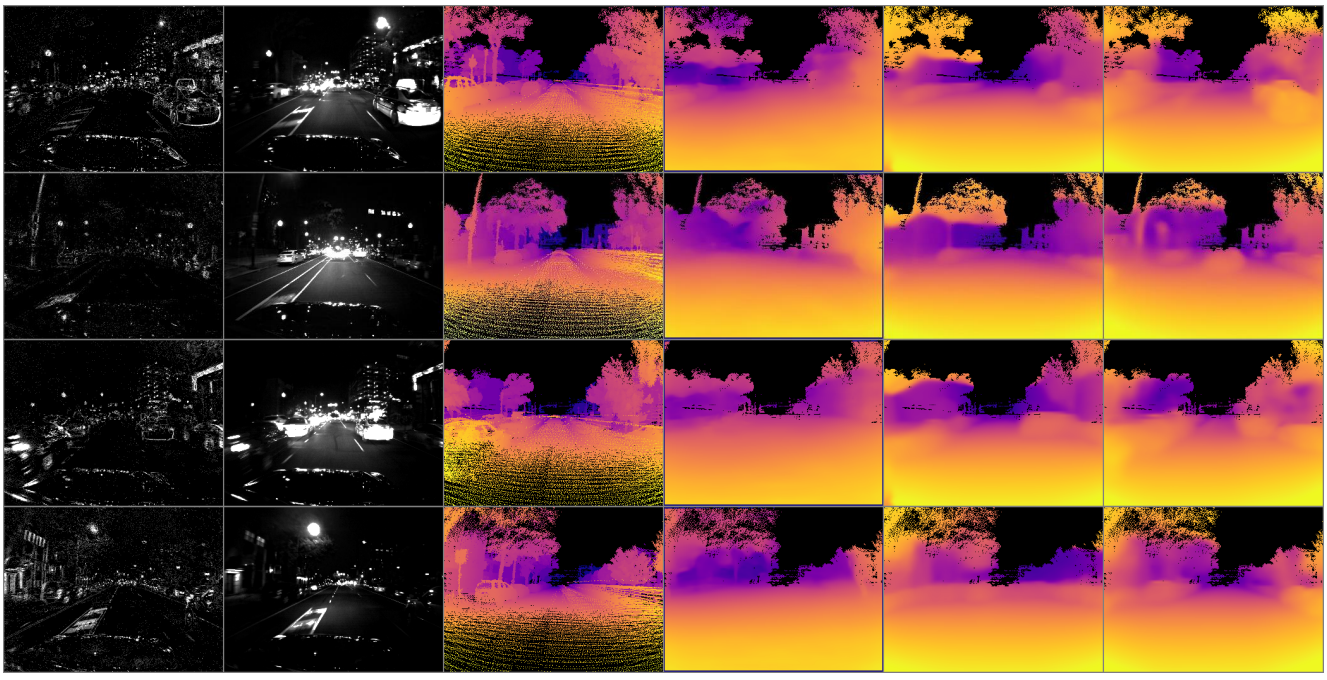

Figure 4. Qualitative results on *outdoor night1* sequence in MVSEC dataset.

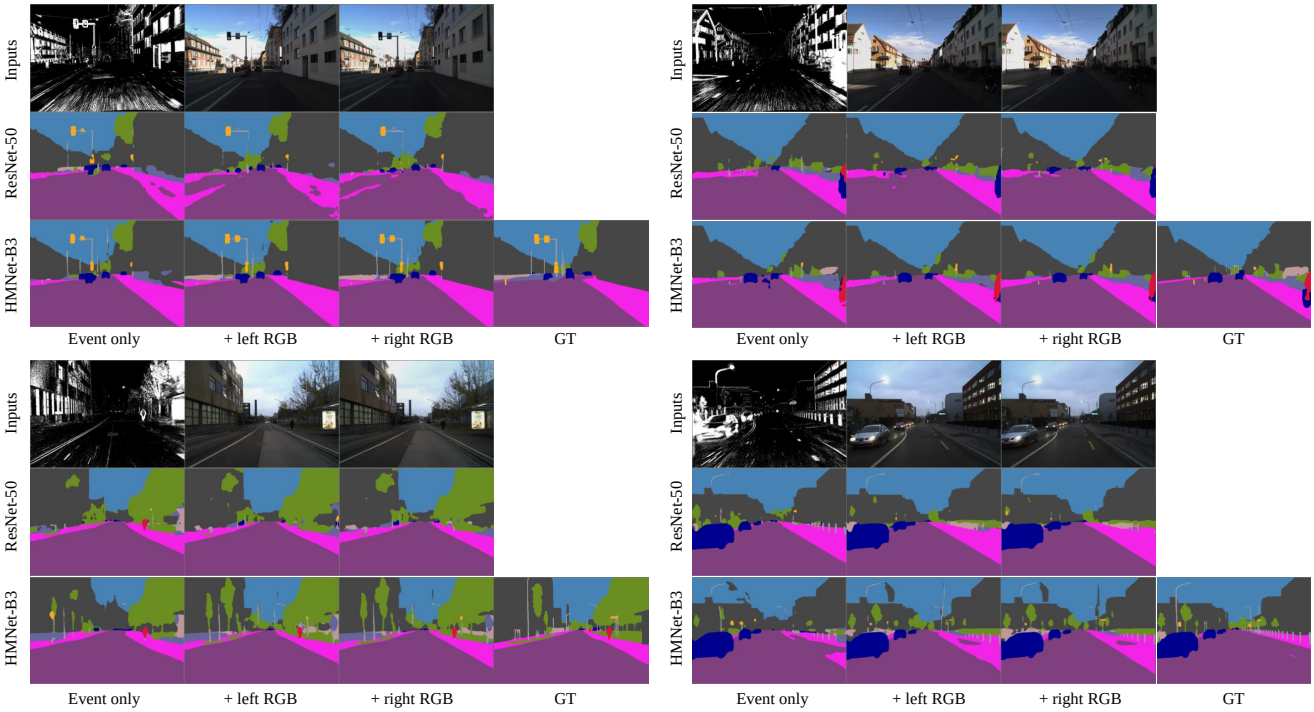

Figure 5. Qualitative results of event-image fusion on DSEC-Semantic dataset.