# Efficient Event Camera Data Pretraining with Adaptive Prompt Fusion

Quanmin Liang[1,2*] Qiang Li[4*] Shuai Liu[1] Xinzi Cao[1,2] Jinyi Lu[1,2]
Feidiao Yang[2] Wei Zhang[2†] Kai Huang[1†] Yonghong Tian[2,3]
[1] School of Computer Science and Engineering, Sun Yat-Sen University
[2] Department of Intelligent Computing, Pengcheng Laboratory
[3] School of Computer Science, Peking University [4] Xpeng Motors Technology Co Ltd
{liangqm5@mail2,liqiang27@mail2,huangk36@mail}.sysu.edu.cn, zhangwei1213052@126.com

## Appendix

## 1. Experiment Settings

We use ViT-S/16 as the pretraining model, freezing its weights during pretraining and jointly training it with STP during fine-tuning stage to adapt to downstream tasks.

### 1.1. Pre-training

Our pretraining setup primarily follows the methodology outlined in previous work [14]. The hyperparameters are detailed in Tab. 1(a). Specifically, the learning rate is linearly scaled with the batch size, i. e., lr = base lr $\times$ batch size / 256.

### 1.2. Object Recognition

We fine-tuned our STP on the N-ImageNet [7], N-Caltech101 [9], N-Cars [10], and CIFAR-10-DVS [4] datasets to evaluate its performance on the object recognition task (Tab. 1(b)). For the N-Caltech101, N-Cars, and CIFAR-10-DVS datasets, we adjusted the final classification head of the VIT model to match the number of classes in these datasets. Additionally, since the N-Caltech101 and CIFAR-10-DVS datasets do not have predefined training and testing splits, we followed previous work [14] and randomly split these datasets, using 80% for training and 20% for testing.

### 1.3. t-SNE Visualization Analysis

In Fig. 4 of maintext, we present the results of the t-SNE visualization analysis. To make the t-SNE analysis more challenging and better highlight the advantages of our method, we selected 10 visually similar classes (all belonging to the bird category) from the N-ImageNet test set [7], as detailed in Tab. 2. We first reduced the dimension of the ViT classification token from 384 to 50 using PCA, then projected it onto a 2D plane using the t-SNE algorithm for visualization.

Table 1. Hyperparameters for pretraining (a) and for finetuning on the object recognition task (b).

(a) Pre-training

| Hyperparameters | Value |
|---|---|
| optimizer | AdamW |
| base lr | $1.5 \times 10^{-4}$ |
| weight decay | $3 \times 10^{-2}$ |
| batch size | 512 |
| epochs | 100 |
| warmup epochs | 20 |
| lr scheduler | cosine |
| label smoothing | 0.8 |

(b) Fine-tuning on the object recognition task

| Hyperparameters | N-ImageNet | N-Caltech101 | N-Cars | CIF10 |
|---|---|---|---|---|
| optimizer | AdamW | AdamW | AdamW | AdamW |
| base lr | $1 \times 10^{-4}$ | $2.5 \times 10^{-4}$ | $1.25 \times 10^{-4}$ | $2.5 \times 10^{-4}$ |
| weight decay | $1 \times 10^{-1}$ | $5 \times 10^{-2}$ | $5 \times 10^{-2}$ | $3 \times 10^{-1}$ |
| batch size | 256 | 512 | 512 | 512 |
| epochs | 20 | 100 | 100 | 100 |
| warmup epochs | 5 | 20 | 20 | 20 |
| lr scheduler | cosine | cosine | cosine | cosine |
| gradient clipping | 5 | 5 | 5 | 5 |
| drop path rate | $1 \times 10^{-1}$ | $1 \times 10^{-1}$ | $1 \times 10^{-1}$ | $1 \times 10^{-1}$ |

### 1.4. Semantic Segmentation

For the semantic segmentation task, we conducted two sets of experiments. In the first set, we pretrained the model on N-ImageNet [7] and then trained and tested it on the DDD17 [3] and DSEC [5] datasets. To ensure a fair comparison with ECDDP [15], we conducted a second set of experiments by pretraining STP and the image-pretrained model on the

*Equal Contribution
†Corresponding Author

Table 2. Selected Categories and Their Names for t-SNE Analysis.

| Classes | Name |
|---|---|
| n01530575 | goldfinch |
| n01531178 | house finch |
| n01532829 | snowbird |
| n01534433 | indigo bird |
| n01537544 | American robin |
| n01558993 | bulbul |
| n01560419 | jay |
| n01580077 | magpie |
| n01582220 | chickadee |
| n01592084 | water ouzel |

E-TartanAir [15] dataset (hyperparameters detailed in Tab. 3) and finetuning it on downstream tasks. Following ECDDP [15], we generated the E-TartanAir dataset by performing frame interpolation on TartanAir using EMA-VFI [17], followed by event synthesis with V2E [6]. Ten scenes from the TartanAir [12] dataset were selected for this process (see Tab. 5 for scene details), and the same V2E hyperparameter settings as ECDDP were used. During pretraining, the weights of the image-pretrained model were frozen, while all weights were optimized during fine-tuning. For the semantic segmentation task, we embedded the UperNet decoder [1, 13] into the pretrained model and fine-tuned it alongside STP on the dataset. We trained using cross-entropy and Dice loss [11], and evaluated performance with the mean Intersection over Union (mIoU) metric. Table 4 shows our finetuning hyperparameters. We present more semantic segmentation results on the DSEC dataset in Figure 1.

Table 3. Pretraining hyperparameters on the E-TartanAir [15] dataset.

| Hyperparameters | E-TartanAir |
|---|---|
| optimizer | AdamW |
| batch size | 512 |
| epochs | 100 |
| lr | $1 \times 10^{-3}$ |
| lr scheduler | cosine |
| warmup epochs | 10 |
| weight decay | $4 \times 10^{-2}$ |
| momentum | 0.992 |
| momentum scheduler | cosine |
| drop path rate | $1 \times 10^{-1}$ |

Additionally, in STP, the model generates hierarchical features, which can be utilized for semantic segmentation tasks. To leverage these features, we apply a linear projection layer to transform them into the same embedding dimension and connect them to the backbone (w/ STP). The specific implementation is illustrated in Fig. 2. This approach effectively provides more detailed temporal information, significantly improving the performance of semantic segmentation.

Table 4. Fine-tuning hyperparameters on the DDD17 [3] and DSEC [5] datasets.

| Hyperparameters | DDD17 | DSEC |
|---|---|---|
| optimizer | AdamW | AdamW |
| lr | $1 \times 10^{-3}$ | $1 \times 10^{-3}$ |
| weight decay | $5 \times 10^{-2}$ | $5 \times 10^{-2}$ |
| batch size | 32 | 32 |
| epochs | 100 | 100 |
| warmup epochs | 10 | 10 |
| lr scheduler | cosine | cosine |
| gradient clipping | 3 | 3 |
| drop path rate | $1 \times 10^{-1}$ | $1 \times 10^{-1}$ |

Table 5. Scene details of the E-TartanAir [15] dataset.

| Scene name |
|---|
| amusement |
| carwelding |
| endofworld |
| japanesealley |
| office |
| ocean |
| oldtown |
| office2 |
| seasonsforest |
| seasidetown |

## 1.5. Optical Flow Estimation

Table 6. Fine-tuning hyperparameters on the MVSEC [18] datasets.

| Hyperparameters | MVSEC |
|---|---|
| optimizer | AdamW |
| lr | $1 \times 10^{-3}$ |
| weight decay | $1 \times 10^{-4}$ |
| batch size | 256 |
| epochs | 100 |
| warmup epochs | 10 |
| lr scheduler | cosine |
| gradient clipping | 1 |

Similar to the semantic segmentation task, we added an additional experiment for fair comparison with ECDDP [15]. In this experiment, we pretrained the model on the E-TartanAir [15] dataset and fine-tuned it on the MVSEC [18]

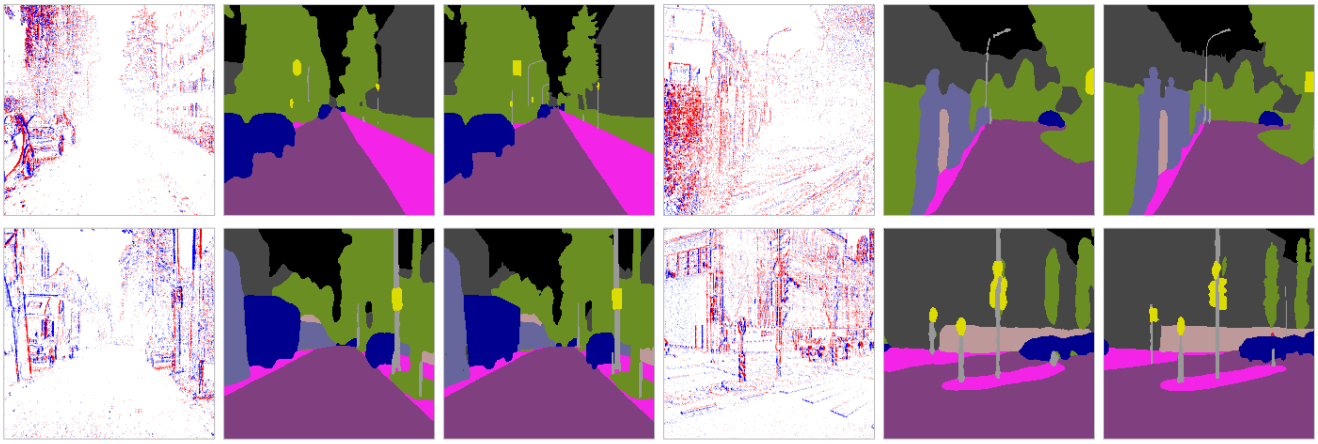

Figure 1. Examples of semantic segmentation on the DSEC dataset. Columns 1/4 show event images, columns 2/5 show segmentation results, and columns 3/6 show the ground truth.

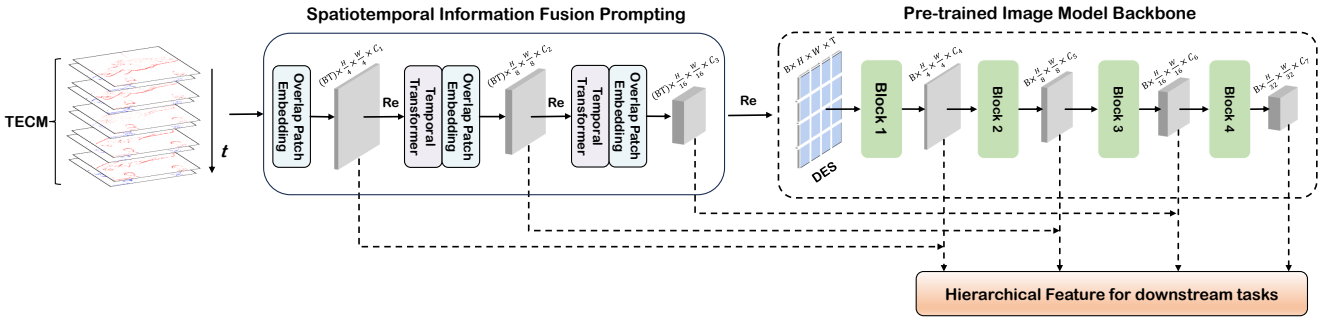


Figure 2. The framework for utilizing the Hierarchical Features from STP for semantic segmentation.

dataset. For models pretrained on N-ImageNet, we attached a UperNet decoder [1, 13] to our pretrained network for optical flow estimation. Additionally, inspired by previous work [14], we added a patch embedding layer as used in [16] to the ViT. We use the $L1$ loss for supervision and train using the MVSEC dataset [18] setup defined by [14]. Detailed optimization settings can be found in Tab. 6.

For models pretrained on the E-TartanAir dataset, we adopted the TMA architecture [8], consistent with ECDDP [15]. Specifically, we utilized four transformer blocks from the image-pretrained model as the encoder for TMA. The weights of these blocks were frozen during the pretraining stage and trained during fine-tuning. The pretraining hyperparameters on E-TartanAir are detailed in Tab. 3. Subsequently, we fine-tuned the model on the MVSEC dataset, with dataset splits following the protocols outlined in [2, 17]. The finetuning hyperparameters on MVSEC are identical to those listed in Tab. 6. The visual results of the optical flow estimation can be seen in Figure 3.

Table 7. Ablation studies on the Model Hyperparameters and Number of Event Stream Segments $T$.

(a) Kernel size of OPE

| $\{k_1, k_2, k_3\}$ | #Params | Ft. Acc |
|---|---|---|
| $\{6,\ 4,\ 4\}$ | **1.1 M** | 68.11 |
| $\{8,\ 6,\ 6\}$ | 2.2 M | 68.87 |
| $\{10,\ 8,\ 8\}$ | 3.7 M | **69.01** |

(b) Ablation of $T$

| $T$ | Pr. Acc | Ft. Acc |
|---|---|---|
| 3 | 65.69 | 68.65 |
| 5 | 66.01 | 68.87 |
| 7 | **66.13** | **68.96** |

## 2. Ablation Studies

**Ablation Study on Model Hyperparameters**. We further explored the impact of the kernel size in the Overlap Patch Embedding on model performance, as this pa-

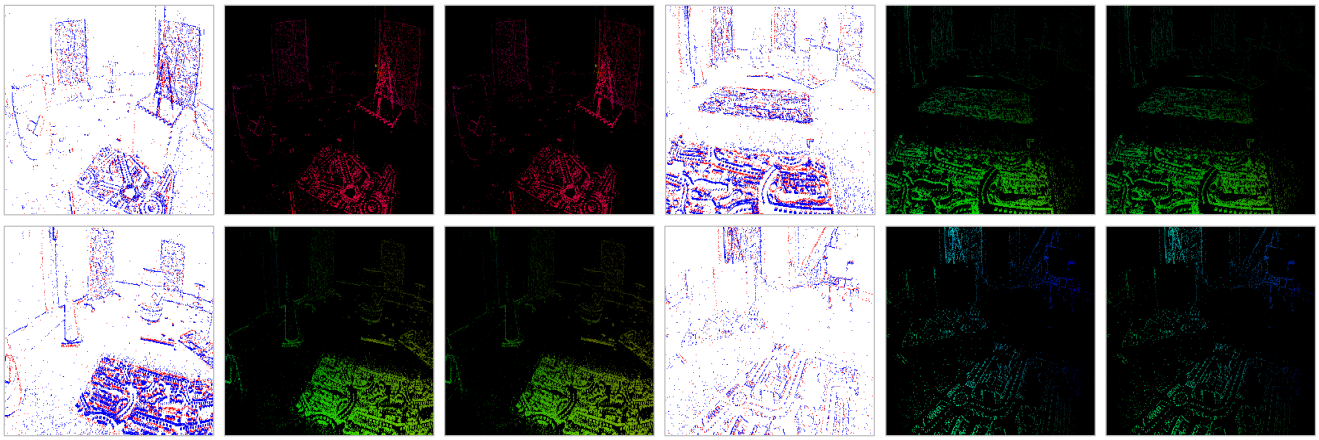

Figure 3. Visualization of the optical flow estimation results on MVSEC dataset. Columns 1/4 show event images, columns 2/5 show optical flow estimation results, and columns 3/6 show the ground truth.

rameter determines the size of the local receptive field during event data encoding. In our previous training, we set $\{k_1 = 8,\ k_2 = 6,\ k_3 = 6\}$. As shown in Tab. 7(a), the kernel size has a significant effect on the parameter count of the STP model and also influences its performance on downstream tasks. This demonstrates that increasing the local receptive field can effectively alleviate the overfitting caused by data sparsity.

**Ablation Studies on Number of Event Stream Segments** $T$. Segmenting the event stream effectively preserves its temporal information. However, increasing the number of segments also increases the computational cost, impacting the model's runtime performance. Following the approach used in Voxid grid [19], we set $T = 5$ (Additional visualizations of DES and TECM are provided in Fig. 4). Additionally, we explored the impact of different values of $T$ on STP performance. The results are shown in Table 7(b).

## References


[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2, 3

[2] Sami Barchid, José Mennesson, and Chaabane Djéraba. Exploring joint embedding architectures and data augmentations for self-supervised representation learning in event-based vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3912, 2023. 3

[3] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*, 2017. 1, 2

[4] Wensheng Cheng, Hao Luo, Wen Yang, Lei Yu, and Wei Li. Structure-aware network for lane marker extraction with dynamic vision sensor. *arXiv preprint arXiv:2008.06204*, 2020. 1

[5] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021. 1, 2

[6] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1312–1321, 2021. 2

[7] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021. 1

[8] Haotian Liu, Guang Chen, Sanqing Qu, Yanping Zhang, Zhijun Li, Alois Knoll, and Changjun Jiang. Tma: Temporal motion aggregation for event-based optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9685–9694, 2023. 3

[9] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:159859, 2015. 1

[10] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018. 1

[11] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017. 2

[12] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and

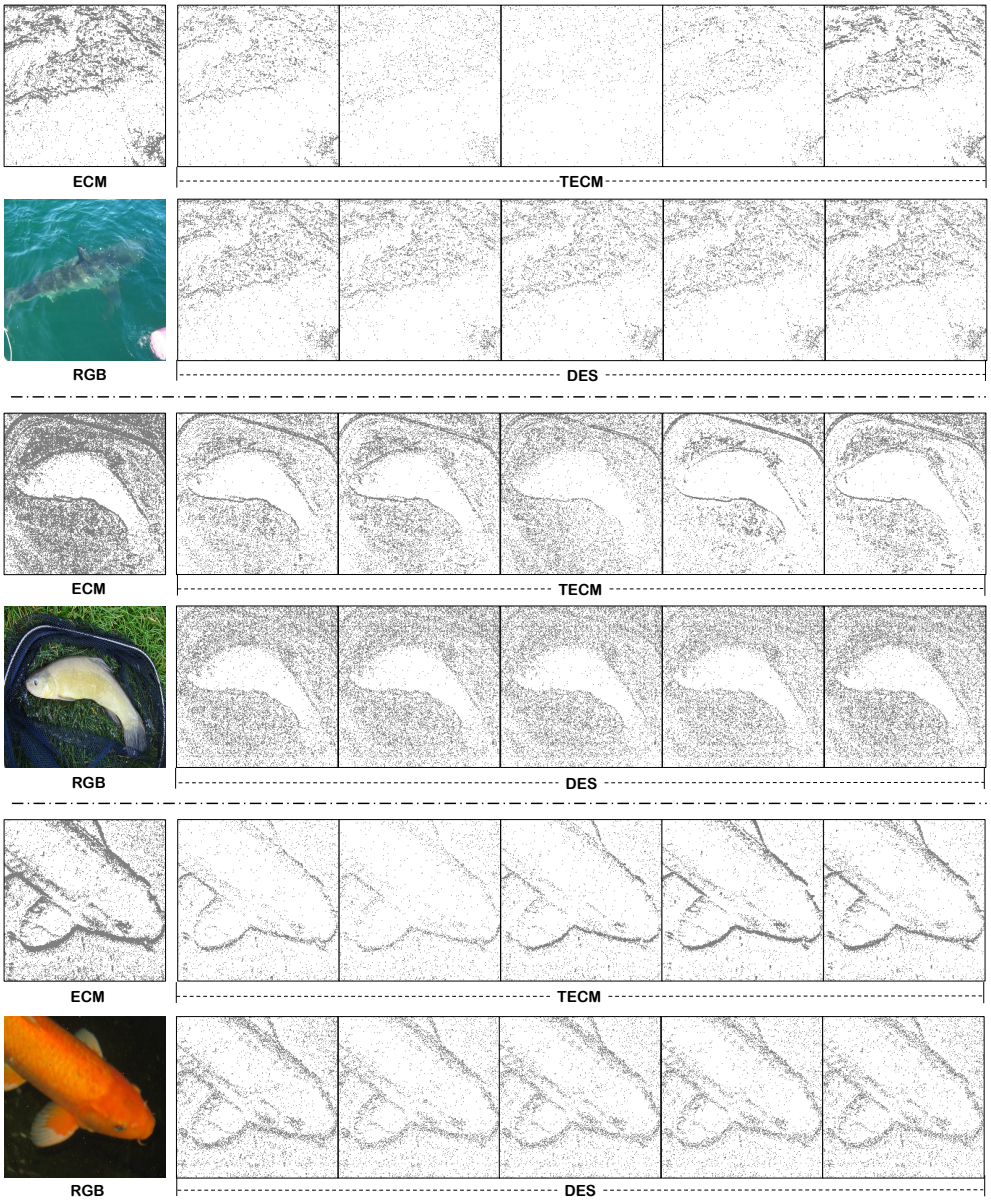


Figure 4. The representations of event data (ECM and TECM) and their corresponding RGB images are shown, with DES generated from TECM through STP. For ease of visualization, we overlay positive and negative events.

Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. 2020. 2

[13] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 2, 3

[14] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10699–10709, 2023. 1, 3

[15] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data dense pre-training. In *European Conference on Computer Vision*, pages 292–310. Springer, 2025. 1, 2, 3

[16] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip HS Torr, Wayne Zhang, and Dahua Lin. Vision transformer with progressive sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 387–396, 2021. 3

[17] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5682–5692, 2023. 2, 3

[18] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3): 2032–2039, 2018. 2, 3

[19] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 4