Supplemental Materials for "Event Camera Data Pre-training"

Yan Yang[1] Liyuan Pan[2] Liu Liu[3]
[1]BDSI, ANU [2]BITSZ & School of CSAT, BIT [3]Cyberverse Dept., Huawei
Yan.Yang@anu.edu.au liyuan.pan@bit.edu.cn liuliu33@huawei.com

In this supplementary material, we provide contents that are omitted in the main paper due to space restrictions. Specifically, i) the details of event data augmentation are given in Sec. 1; ii) experiment details for pre-training and fine-tuning are given in Sec. 2; iii) more ablation studies are given in Sec. 3.

1. Event Data Augmentation

Generating different views of the same event data is one of the most important parts of our self-supervised event data pre-training framework. Directly using augmentation methods from the RGB domain leads to meaningless event images. For example, blurring a binary event image [4], whose pixel values are either 1 or 0, violates the representation definition. In the following, we consider the four main augmentation methods, RandomResizedCrop, Gaussian Blur, ColorJitter, and Random Patch Masking [2, 9]. We show how to perform the four augmentation methods on event data.

RandomResizedCrop. Different from the commonly used bicubic or bilinear interpolations [9, 2], the nearest neighbor interpolation is the only way to avoid wrong interpolations at discontinuity regions in the event data. The comparison of using past RGB-based bicubic/bilinear and event-based nearest neighbor interpolations on sample event data is given in Fig. 1 (a, c, d). Please note that the bicubic/bilinear interpolation (c) has wrongly changed the color of the original event image (a). In contrast, our event-based nearest neighbor interpolation (d) correctly preserves the color (polarity) of the original event image.

Gaussian Blur. To blur an event data, we distort the position of events, i. e., adding random Gaussian noise to spatial positions of events. One should also round distorted spatial positions to the nearest integers, avoiding wrong interpolations when converting to event images. The comparison of using traditional Gaussian Blur and our event data blur on sample event data is given in Fig. 1 (a, e, f). Please note that traditional Gaussian Blur (e) has wrongly changed the color of the original event image (a). In contrast, our blur method (f) correctly preserves the color (polarity) of the original event image.

ColorJitter. We treat each channel of an event image as a gray-scale image, thus we only need to change the brightness and contrast of an event image. This is achieved by scaling and shifting the occurrence of events generated at each spatial position. Please refer to Fig. 1 (a, g, h) for an example showing that the occurrences of positive and negative events are adjusted. Note that traditional ColorJitter (g) wrongly modifies the pixel values of positions without any events, compared to the original event image (a). In contrast, our method (h) correctly changes the color (polarity) of the original event image.

Random Patch Masking. The event data is usually spatially sparse, which generally occurs around the edges of a scene. When masking event image patches, the information quantity of a patch should be considered, to avoid amplifying the event sparsity. The details of our masking method are given in Section 3 of the main paper. Fig. 1 (i, j) shows a sample comparison between previous random masking and our proposed conditional masking strategy. Note that the previous random masking method (i) amplifies the event sparsity.

2. Experiment Details

2.1. Pre-training

We separately explore the standard ViT-S/16 and ResNet50 architecture for f_{e} in our pre-training tasks [6, 2]. We follow the MoCov3 [2] on projector designs of $h_{\mathsf{e}}^{\texttt{img}}$ and $h_{\mathsf{e}}^{\texttt{evt}}$, with a two-layer MLP that has hidden dimension of 4096 and output dimension of 256. We set the projector h_{I} to a single linear layer with an output dimension of 256. Our pre-training settings are given in Tab. 1. We use simplified augmentations methods, RandomFlip, RandomResizedCrop, and our patch masking, as we observe no performance degradation compared to additionally using Gaussian Blur and ColorJitter. The learning rate lr is linearly scaled with batch size, i. e., $\text{lr} = \text{base\_lr} \times \text{batch size}/256$ [8]. We initialize our model with checkpoints pre-trained

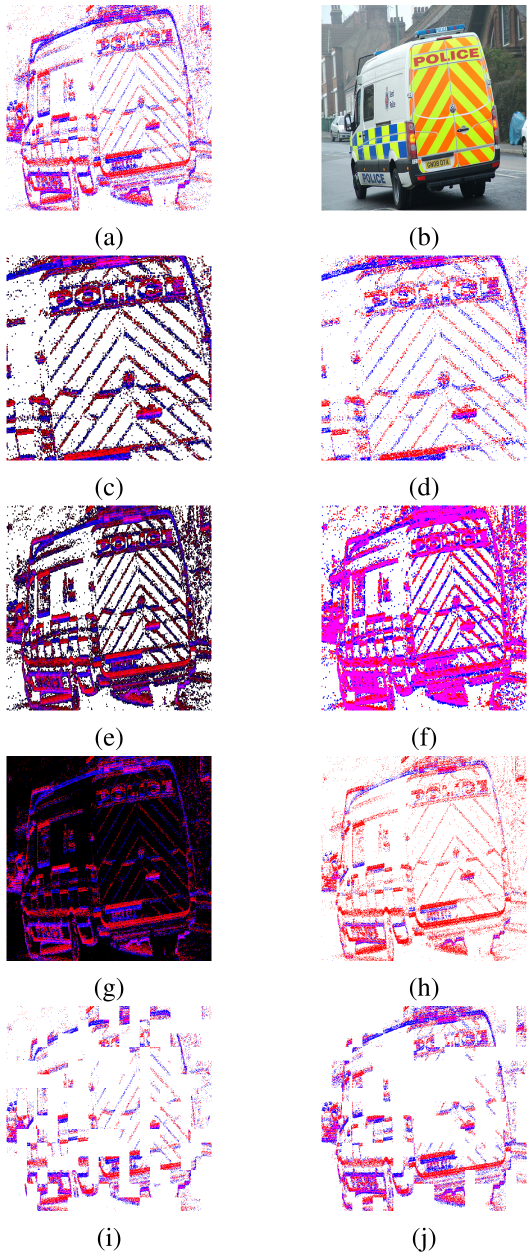


Figure 1: *Comparison of event data augmentations. We compare event images generated by previous augmentation methods and our methods. We use red and blue to indicate positive and negative events, and white for no events. Please note that when negative and positive events both occur at the same position, the mixed color is fuchsia. (a) and (b) are the original event image and corresponding RGB image for assisting visualization. In the remaining rows, we show pairwise augmented event images by previous methods (Left) and our methods (Right): RandomResizedCrop ((c) and (d)), Gaussian Blur ((e) and (f)), ColorJitter ((g) and (h)), and Patch Masking ((i) and (j)).*

by MoCov3, which improves the line probing accuracy by 3.16%.

Table 1: *Hyperparameters of pre-training our method on the N-ImageNet dataset [11]*

Hyperparameters	Value
optimizer	AdamW
base_lr	1.5×10^{-4}
weight decay	3×10^{-2}
layer decay	none
batch size	1024
epochs	300
warmup epochs	40
lr scheduler	cosine
momentum	0.99
λ_1	2
drop path rate	none

Table 2: *Hyperparameters of fine-tuning our method on the N-ImageNet dataset.*

Hyperparameters	Value
optimizer	AdamW
base_lr	1×10^{-4}
weight decay	3×10^{-1}
layer decay	7.5×10^{-1}
batch size	2048
epochs	100
warmup epochs	20
lr scheduler	cosine
gradient clipping	5
drop path rate	1×10^{-1}

2.2. Object Recognition on the N-ImageNet

Our fine-tuning settings are given in Tab. 2. We load the checkpoint of the trained learning probing head before starting fine-tuning, for shortening fine-tuning epochs. One may note iBoT achieves poor fine-tuning performance (Table 1 in the main paper). In our experiments, iBoT easily collapses in fine-tuning, though we have tried our best to find its best learning rate.

2.3. Object Recognition on Other Small Datasets

Our fine-tuning schedules for the N-Cars [14], N-Caltech101 [12], and CIFAR-10-DVS datasets [3] are given in Tab. 3. Though large fine-tuning epochs are used, only 9k, 3.6k, and 11.2k optimization steps are performed on the datasets, respectively.

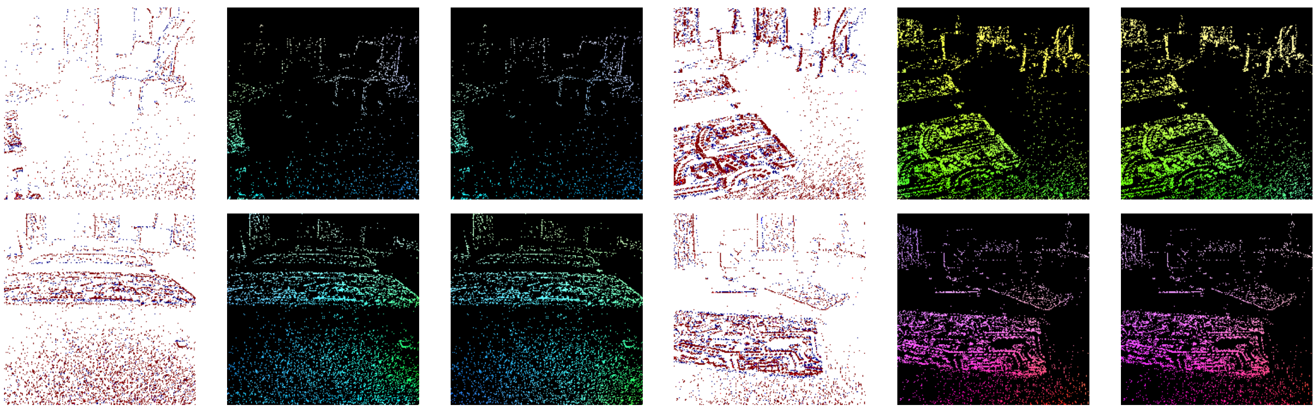
The N-Caltech101 and CIFAR-10-DVS datasets have not provided the train-test splits. We fix a seed (123) to randomly split 80% data for training, and the remaining data is used for testing.

2.4. Flow Estimation

We append a UperNet decoder [9, 1] to our pre-trained network for flow estimation. In addition, we replace the

Table 3: *Hyperparamters of fine-tuning our method on the N-Cars, N-Caltech101, and CIFAR-10-DVS datasets.*

Hyperparameters	N-Cars	N-Caltech101	CIFAR-10-DVS
optimizer	AdamW	AdamW	AdamW
base_lr	1.25×10^{-4}	2.5×10^{-4}	2.5×10^{-4}
weight decay	5×10^{-2}	5×10^{-2}	3×10^{-1}
layer decay	7.5×10^{-1}	7.5×10^{-1}	7.5×10^{-1}
batch size	1024	1024	1024
epochs	600	600	1600
warmup epochs	60	60	80
lr scheduler	cosine	cosine	cosine
gradient clipping	5	5	5
drop path rate	1×10^{-1}	1×10^{-1}	1×10^{-1}



(a) (b) (c) (d) (e) (f)

Figure 2: *Optical flow prediction examples of our method on the MVSEC dataset [18]. (a)/(d) are event images, where red and blue indicate positive and negative events. (b)/(e) are ground-truth optical flows. (c)/(f) are our predicted optical flows.*

Table 4: *Hyperparameters of fine-tuning our method on the MVSEC dataset.*

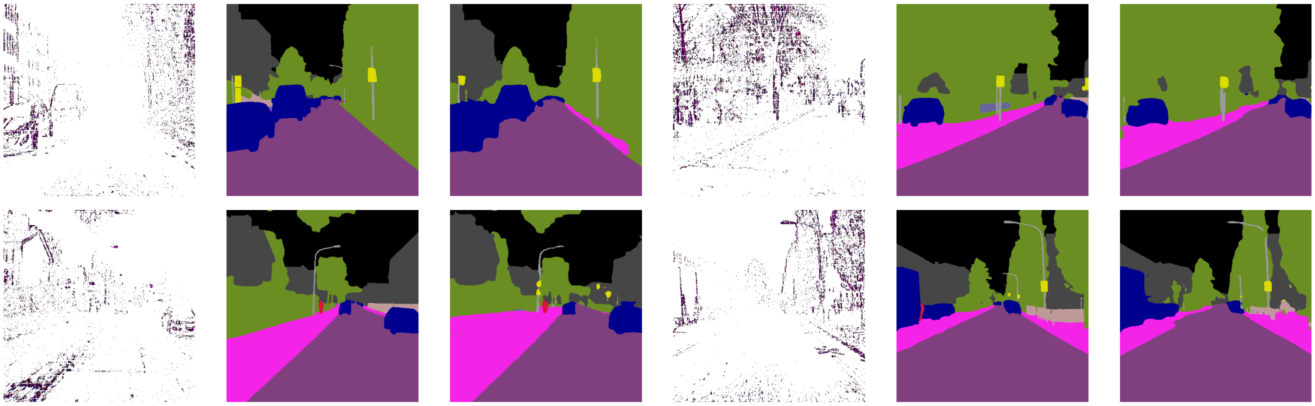
Hyperparameters	Value
optimizer	AdamW
lr	1×10^{-3}
weight decay	1×10^{-4}
layer decay	none
batch size	256
epochs	150
warmup epochs	20
lr scheduler	cosine
gradient clipping	none
drop path rate	1×10^{-1}

Table 5: *Hyperparameters of fine-tuning our method on the DDD17 and DSEC datasets.*

Hyperparameters	DDD17	DSEC
optimizer	AdamW	AdamW
lr	1×10^{-3}	1×10^{-3}
weight decay	5×10^{-2}	5×10^{-2}
layer decay	7.5×10^{-1}	7.5×10^{-1}
batch size	16	12
epochs	100	200
warmup epochs	10	10
lr scheduler	cosine	cosine
gradient clipping	3	3
drop path rate	1×10^{-1}	1×10^{-1}

patch embed layer of ViT with the one used in [17]. For fine-tuning, we use the L1 loss between predicted flow and ground-truth flow as supervision [7]. Our optimization settings are given in Tab. 4.

The MVSEC dataset collects event data from both indoor and outdoor scenes, which is composed of 'indoor_flying1', 'indoor_flying2', 'indoor_flying3', 'outdoor_day1', and 'outdoor_day2'. Our training data is composed of two outdoor scenes and 1% randomly sampled data from 'indoor_flying1', where the seed is fixed. The remaining data is used for testing. The poor-quality data is filtered out [16]. The results are given in Table 3 in the main paper. Please note that EST and DCEIFlow are originally trained with 4×10^4 and 2.2×10^4 samples. Here, we only have around



(a) (b) (c) (d) (e) (f)

Figure 3: *Semantic segmentation predictions examples of our method on the DSEC dataset [18]. (a)/(d) are event images, where red and blue indicate positive and negative events. (b)/(e) are ground-truth segmentation images, and pixel colors denote semantic classes. (c)/(f) are our predicted segmentation images.*

(a) (b) (c) (d) (e) (f)

Figure 4: *Attention maps of our pre-trained model (without any fine-tuning) on sample data from the N-ImageNet dataset [11]. (a)/(d) are event images. Similarly, we use red and blue to indicate positive and negative events. (b)/(e) are corresponding natural RGB images used for visualization assistance. (c)/(f) are our attention maps.*

5×10^3 samples for training. Our method outperforms them under the same setting. Please refer to Fig. 2 for more flow estimation results of our method.

2.5. Semantic Segmentation

We use the UperNet decoder [9, 1] and 3D-expanded patch embed layer [17] for semantic segmentation. The cross-entropy and Dice losses [15] are used for fine-tuning. Our optimization settings are given in Tab. 5. Please refer to Fig. 3 for more semantic segmentation results of our method.

2.6. Attention maps.

Please refer to Fig. 4 for more attention maps estimated by our method. Note that features are extracted from our pre-trained model, and no fine-tuning is performed.

Table 6: *Results of pre-training on the MobileNet_Small [10] backbone. The 'Scratch' and 'Supervised' settings separately denote training from scratch and supervised pre-training on ImageNet-1K [5].*

Method	Linear Probing		Fine-tuning	
	acc@1	acc@5	acc@1	acc@5
Scratch	-	-	40.36	64.22
Supervised	-	-	42.83	66.95
Ours	**41.34**	**64.88**	**45.19**	**69.14**

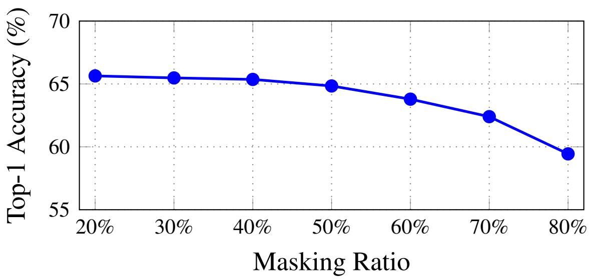


Figure 5: *The performance of our method with respect to different masking ratios.*

Table 7: *Ablation of* h_l.

h_l	Linear Probing		Fine-tuning	
	acc@1	acc@5	acc@1	acc@5
MoCov3 projector [2]	58.17	80.94	63.22	85.42
Linear projector	**59.90**	**82.26**	**64.84**	**86.30**

3. Ablation Studies

In this section, all experiments are conducted on the N-ImageNet dataset.

3.1. Mobile Applications

To demonstrate the effectiveness of our method for mobile and embedded vision applications, we pre-train MobileNet_Small (2.5M parameters) network [10]. The results are given in Tab. 6. Compared with supervised pre-training on the ImageNet-1K dataset, our method achieves better performance.

3.2. Masking ratios

We study the masking ratio (from 20% to 80%) for our conditional masking strategy. Our method is pre-trained with different masking ratios, and fine-tuned on the N-ImageNet (Fig. 5). Balancing performance and computation costs, we use a masking ratio of 50%.

3.3. Architectures of h_l and f_l

The comparisons of using different architectures for h_l and f_l are given in Tab. 7 and Tab. 8, respectively.

Table 8: *Ablation of* f_l.

f_l	Linear Probing		Fine-tuning	
	acc@1	acc@5	acc@1	acc@5
CLIP ViT-B/32 [13]	59.90	**82.26**	**64.84**	**86.30**
CLIP ViT-B/16 [13]	**60.02**	81.98	64.35	86.10
CLIP ViT-L/16 [13]	58.90	81.02	63.75	85.20
CLIP ResNet50 [13]	59.46	82.03	64.50	86.21
MAE ViT-B/16 [9]	55.78	79.26	61.31	84.07

Table 9: *Ablation of losses.*

Objectives	Linear Probing		Fine-tuning	
	acc@1	acc@5	acc@1	acc@5
$\mathcal{L}_{\text{evt}}$	24.23	46.90	55.49	78.56
$\mathcal{L}_{\text{evt}} + \mathcal{L}_{\text{RGB}}$	56.15	79.35	62.38	84.43
$\mathcal{L}_{\text{evt}} + \mathcal{L}_{\text{RGB}} + \mathcal{L}_{\text{kl}}$	**59.90**	**82.26**	**64.84**	**86.30**

In Tab. 7, we show that setting h_l to a linear projector achieves the best performance. In Tab. 8, we show that setting f_l to a CLIP pre-trained ViT-B/32 achieves the best performance.

3.4. Losses

The effectiveness of our losses is given in Tab. 9. The best performance is obtained when $\mathcal{L}_{\text{evt}}$, $\mathcal{L}_{\text{RGB}}$, and $\mathcal{L}_{\text{kl}}$ are used for training.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: BERT pre-training of image transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2, 4

[2] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 1, 5

[3] Wensheng Cheng, Hao Luo, Wen Yang, Lei Yu, and Wei Li. Structure-aware network for lane marker extraction with dynamic vision sensor. *CoRR*, abs/2008.06204, 2020. 2

[4] Gregory Cohen, Saeed Afshar, Garrick Orchard, Jonathan Tapson, Ryad Benosman, and André van Schaik. Spatial and temporal downsampling in event-based visual classification. *IEEE Trans. Neural Networks Learn. Syst.*, 29(10):5030–5044, 2018. 1

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1

[7] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 5632–5642. IEEE, 2019. 3

[8] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. 1

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022. 1, 2, 4, 5

[10] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1314–1324. IEEE, 2019. 5

[11] 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 (ICCV)*, pages 2146–2156, October 2021. 2, 4

[12] Garrick Orchard, Ajinkya Jayawant, Gregory Cohen, and Nitish V. Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *CoRR*, abs/1507.07629, 2015. 2

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 5

[14] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: histograms of averaged time surfaces for robust event-based object classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1731–1740. Computer Vision Foundation / IEEE Computer Society, 2018. 2

[15] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In M. Jorge Cardoso, Tal Arbel, Gustavo Carneiro, Tanveer F. Syeda-Mahmood, João Manuel R. S. Tavares, Mehdi Moradi, Andrew P. Bradley, Hayit Greenspan, João Paulo Papa, Anant Madabhushi, Jacinto C. Nascimento, Jaime S. Cardoso, Vasileios Belagiannis, and Zhi Lu, editors, *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, 2017, Proceedings*, volume 10553 of *Lecture Notes in Computer Science*, pages 240–248. Springer, 2017. 4

[16] Zhexiong Wan, Yuchao Dai, and Yuxin Mao. Learning dense and continuous optical flow from an event camera. *IEEE Trans. Image Process.*, 31:7237–7251, 2022. 3

[17] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip H. S. Torr, Wayne Zhang, and Dahua Lin. Vision transformer with progressive sampling. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 377–386. IEEE, 2021. 3, 4

[18] 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. *CoRR*, abs/1801.10202, 2018. 3, 4