Learning Normal Flow Directly From Events

Supplementary Material

8. List of Flow Prediction Videos

We provide flow prediction videos for each evaluated dataset—MVSEC, EVIMO2, and DSEC. These visualizations showcase predictions from models trained on each of the three datasets. Figure 8 shows some screenshots of the video. The videos are in this link. The enumeration of videos are as followed:

- DSEC_eval_interlaken_00_a.mov
- DSEC_eval_interlaken_00_b.mov
- DSEC_eval_interlaken_01_a.mov
- DSEC_eval_thun_01_a.mov
- DSEC_eval_thun_01_b.mov
- DSEC_eval_zurich_city_13_a.mov
- DSEC_eval_zurich_city_13_a.mov
- DSEC_eval_zurich_city_13_b.mov
- DSEC_eval_zurich_city_14_a.mov
- DSEC_eval_zurich_city_14_b.mov
- DSEC_eval_zurich_city_14_c.mov
- DSEC_eval_zurich_city_15_a.mov
- EVIMO_eval_scene13_dyn_test_00_000000.mov
- EVIMO_eval_scene13_dyn_test_05_000000.mov
- EVIMO_eval_scene14_dyn_test_03_000000.mov
- EVIMO_eval_scene14_dyn_test_04_000000.mov
- EVIMO_eval_scene14_dyn_test_05_000000.mov
- EVIMO_eval_scene15_dyn_test_01_000000.mov
- EVIMO_eval_scene15_dyn_test_02_000000.mov
- EVIMO_eval_scene15_dyn_test_05_000000.mov
- EVIMO_sfm_scene_03_00_000000.mov
- EVIMO_sfm_scene_03_01_000000.mov
- EVIMO_sfm_scene_03_02_000001.mov
- EVIMO_sfm_scene_03_02_000002.mov
- EVIMO_sfm_scene_03_02_000003.mov
- EVIMO_sfm_scene_03_03_000000.mov
- EVIMO_sfm_scene_03_03_000001.mov
- EVIMO_sfm_scene_03_03_000002.mov
- EVIMO_sfm_scene_03_04_000000.mov
- MVSEC_eval_indoor_flying1.mov
- MVSEC_eval_outdoor_day1.mov

In addition, we provide qualitative evaluation on FPV and VECtor, which ground-truth optical flow is not available. The predictions are generated by the model trained on M+D+E to demonstrate the usability of the our estimator. Figure 10 shows some screenshots and the following videos are uploaded:

- VECtor_sofa_normal.mov
- VECtor_sofa_fast.mov
- VECtor_mountain_normal.mov
- VECtor_mountain_fast.mov
- FPV_outdoor_45_2_davis.mov
- FPV_outdoor_forward_6_davis.mov
- FPV_indoor_45_16_davis.mov
- FPV_indoor_forward_11_davis.mov

9. Dataset Preprocessing

In this section, we detail how to preprocess the data to obtain undistorted normalized per-event optical flow on MVSEC, EVIMO2, and DSEC.

MVSEC & EVIMO2 both provide frame-based forward optical flows in the distorted camera coordinates. We first interpolate the flow in the time domain. If an event (t, x, y) lies between t_0 and t_1, the optical flow at this event is computed as:

$$\mathbf{u}(t,x,y) = \frac{t-t_0}{t_1-t_0} flow(t_1,x,y) + \frac{t_1-t}{t_1-t_0} flow(t_0,x,y)$$

After this, we convert the per-event distorted flow in the raw pixel coordinates into undistorted flow in the normalized pixel coordinates using `cv2.undistortPoints`.

$$\begin{aligned} \text{start} &= \texttt{cv2.undistortPoints}(x, y, K, D) \\ \text{end} &= \texttt{cv2.undistortPoints}(x + \mathbf{u}_x, y + \mathbf{u}_y, K, D) \\ \text{out} &= (\text{end} - \text{start})/(t_1 - t_0) \end{aligned}$$

This will transform the flow into undistorted normalized camera coordinates, with unit normalized pixel per second.

DSEC, different from the previous two datasets, provides frame-based forward optical flow and backward optical flow, which can be used to obtain more accurate per-event optical flow. Specifically, we let

$$\begin{aligned} flow(t_1,x,y) = \frac{1}{2}(\\ flow\_forward(t_1,x,y) - flow\_backward(t_1,x,y) \\ ) \end{aligned}$$

The following procedures are the same as the previous two datasets.

10. Implementation Details of Our Model

We transform the event pixels and flows into undistorted, normalized camera coordinates as explained in Appendix 9. The resulting flows are then scaled such that their unit is in pixels per second. After this scaling, the flow norms fall within a range of 0 to 3.

During training, we randomly sample an event, using a uniform distribution over the logarithm of the flow norm,

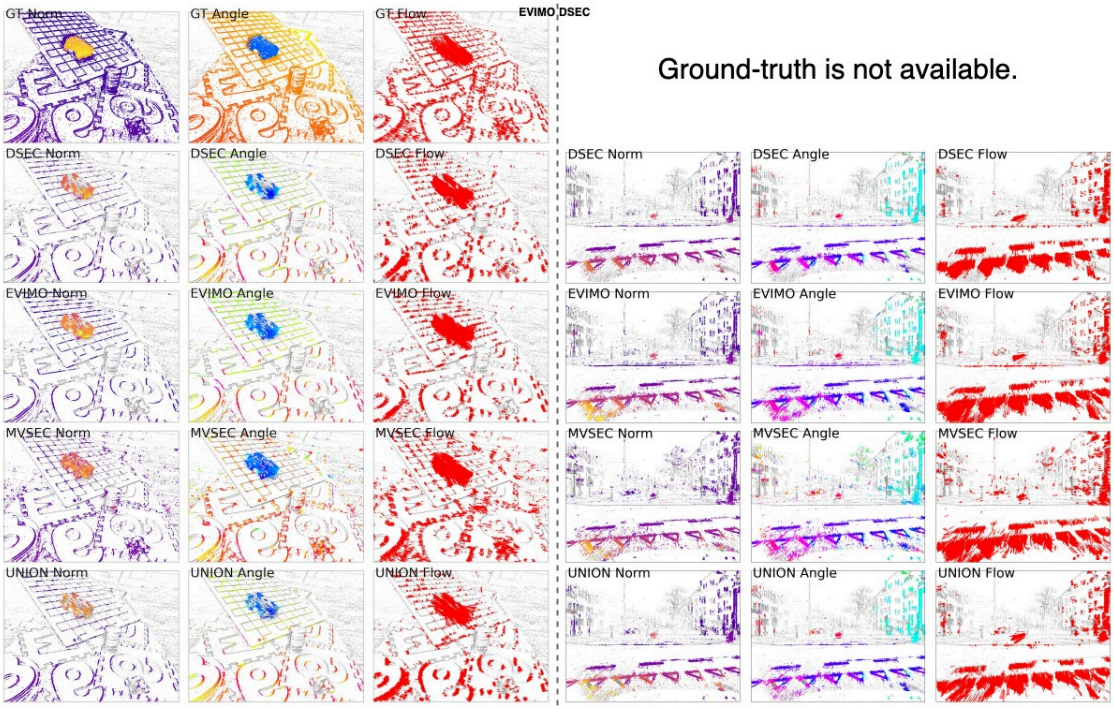

Figure 6. Screenshot of the flow prediction videos. Each row displays the norm, angle, and flow fields of both ground-truth and predicted flows. The first row visualizes the ground-truth optical flow, while subsequent rows show model predictions trained on each dataset. To illustrate the flow field, we sample 5,000 flow points for visualization. If a pixel is gray, it means the flow prediction has a high uncertainty.

	13_00		13_05		14_03		14_04		14_05		15_01		15_02		15_05		Average	
	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos
Norm + Direction Loss	0.972	90.0%	0.912	98.1%	0.749	98.9%	0.762	96.9%	1.150	97.3%	0.559	96.9%	0.610	94.5%	1.274	91.5%	0.87	95.5%
Ours	0.497	96.7%	0.399	99.2%	0.478	99.2%	0.515	98.8%	0.584	98.6%	0.286	98.1%	0.274	96.8%	0.354	95.5%	0.42	97.9%
Difference	↓0.475	↑6.7%	↓0.513	↑1.1%	↓0.271	↑0.3%	↓0.247	↑1.9%	↓0.566	↑1.3%	↓0.273	↑1.2%	↓0.336	↑2.3%	↓0.92	↑4.0%	↓0.45	↑2.4%

Table 5. Comparison between the estimator trained with our motion field loss function and the one trained with the standard norm-plus-direction loss function. Using our motion field function significantly improves the model's performance.

within the range of 0.01 to 3. We then slice the event stream around the sampled event to create the training samples. We apply the data augmentation techniques described in Section 3.4. The pixel radius parameters (δx, δy in Eqn. (3)) are set to 0.02, which correspond to 4.5, 10.4, and 11.1 pixels for the MVSEC, EVIMO2, and DSEC datasets, respectively, measured in terms of raw pixels. The time radius (δt in Eqn. (3)) is 20 ms. The parameter ϵ in Eqn. (5) is set to 0.1. The dimension of the local event encoding is 384. We remove the predictions with circular standard deviation larger than 0.15 (Section 3.5). If the events size within 20 ms is larger than 80,000, we randomly sample 80,000 events from the 20 ms interval. Our model is trained end-to-end in one stage, where the VecKM encoding does not require training. The training converges in 24/24/48/64 hours for M/E/D/M+D+E datasets, using the Adam optimizer with a 1e-3 learning rate.

11. Ablation Studies

We use the EVIMO2-`imo` dataset for our ablation studies because it presents challenging scenarios with independently moving objects. The models are trained on EVIMO2-`imo` training set to better capture the impact of the ablated factors.

11.1. Effect of Motion Field Loss

We show that our estimator benefits from being trained on the novel motion field loss by comparing to an optical flow loss. While variations on average end-point error (AEE) are typically used for supervised training of optical flow estimators, our method, which is designed to estimate normal flow, does not converge when trained with such losses. Thus, we designed the following norm + direction loss to train our estimator to estimate optical flow, defined as follows:

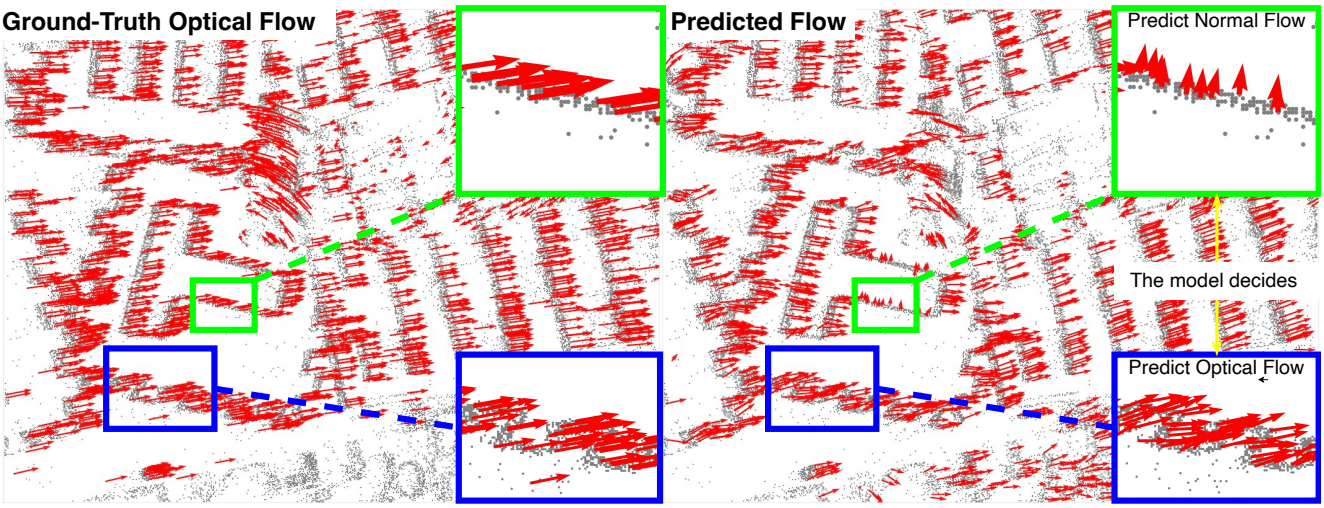

Figure 7. The model can choose between estimating the full optical flow or normal flow depending on the texture of the local region. If the local texture is rich enough (e.g. a corner), the model will estimate full optical flow. If the local texture only contains strong edges, the model will estimate normal flow.

$$\mathcal{L}_1 = \log\left(\frac{\epsilon + ||\mathbf{u}||}{\epsilon + ||\hat{\mathbf{u}}||}\right)^2$$
$$\mathcal{L}_2 = -\frac{\mathbf{u} \cdot \hat{\mathbf{u}}}{||\mathbf{u}|| \cdot ||\hat{\mathbf{u}}||}$$
$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$$

Where $\hat{\mathbf{u}}$ is the output of our method when being trained with this optical flow loss.

As shown in Table 5, our motion field loss function significantly enhances the estimator's performance in terms of PEE and $\%Pos$. The reason is as followed: When optical flow is unambiguous, predicting optical flow minimizes both objectives. However, when optical flow is ambiguous (e.g. the local events correspond to an edge), our normal flow objective guarantees that predicting normal flow will minimize the loss, while optical flow objective does not.

In addition, we analyze the behavior of our estimator qualitatively in Figure 7. After the model is trained using our motion field loss function, the model can choose between estimating full optical flow or normal flow depending on the texture of the local regions. This further justifies the effect of our motion field loss function.

11.2. Effect of Uncertainty Quantification

Figure 9 (Left) provides a comprehensive analysis of the estimator's performance, showing prediction errors alongside the percentage of confident predictions across various ensemble sizes and uncertainty thresholds. The positive correlation observed between prediction errors and uncertainty scores underscores the effectiveness of the uncertainty quantification. Our results indicate that an uncertainty threshold between 0.3 and 0.6 achieves an optimal balance between valid prediction rates and accuracy. Additionally, the table reveals that 3 to 4 ensemble predictions are sufficient for consistent uncertainty estimation, though larger ensembles generally yield improved performance. For scenarios where runtime is not a constraint, employing larger ensembles can enhance prediction accuracy.

We also evaluate the egomotion estimation error when setting different uncertainty threshold, shown at Figure 9 (Right). We found an uncertainty threshold of 0.15 yields the best performance. When the threshold is too low, the number of events may be too few to yield good estimation.

11.3. Runtime and Memory Usage

Table 7 presents the computational cost of our estimator. Unlike frame-based methods, the runtime of our point-based estimator varies with event density. When event density is high, it may run slower than frame-based methods. However, the computation remains generally feasible even on entry-level GPUs.

12. Per-Scene Normal Flow Evaluation on EVIMO2

We present the per-scene normal flow evaluation on EVIMO2-`imo`, as shown in Table 8.

13. Per-Scene Normal Flow Evaluation on DSEC

We present the per-scene normal flow evaluation on DSEC training set holdout, as shown in Table 6. For the model trained on DSEC, it is only trained on

zurich_city_02_c, with a duration of 80 seconds. Our model performs well on day scenes, while its performance degrades when applied to night scenes. As shown in Figure 8, the performance degradation is mainly because many events are continuously triggered by the flickering light. At the same time, our uncertainty quantification module can assign high uncertainty scores to those events.

14. Per-Scene Egomotion Evaluation on EVIMO2

We present the per-scene egomotion evaluation on EVIMO2, as shown in Figure 11.

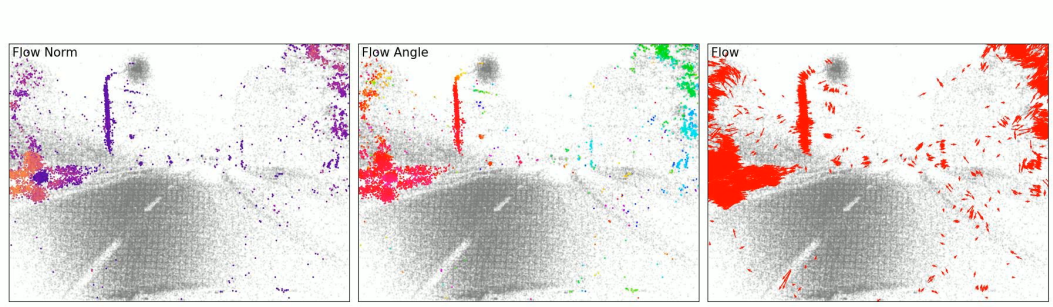

Figure 8. Flow prediction on a night scene `zurich_city_02_e`. The flickering light causes many events to be triggered continuously, which degrades the performance of our model.

	Day Scenes							
training	zurich_city_01_a		zurich_city_02_a		zurich_city_02_d		zurich_city_05_a	
set	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos
MVSEC	1.377	82.5%	1.162	99.6%	1.316	93.3%	2.176	96.3%
EVIMO	1.008	86.6%	0.977	99.8%	0.749	95.1%	1.281	96.0%
DSEC	0.873	87.9%	0.900	99.8%	0.619	95.5%	1.187	95.4%

	Day Scenes							
training	zurich_city_05_b		zurich_city_06_a		zurich_city_07_a		zurich_city_08_a	
set	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos
MVSEC	3.451	73.8%	1.134	84.6%	1.528	82.6%	0.904	99.5%
EVIMO	1.457	81.2%	1.236	82.1%	0.978	85.5%	0.912	99.8%
DSEC	1.485	80.5%	1.493	81.3%	0.972	85.9%	0.820	99.8%

	Night Scenes							
training	zurich_city_02_e		zurich_city_03_a		zurich_city_10_a		zurich_city_10_b	
set	PEE	%Pos	PEE	%Pos	PEE	%Pos	PEE	%Pos
MVSEC	2.952	74.1%	2.390	74.2%	2.088	96.8%	1.149	99.0%
EVIMO	2.141	73.1%	1.060	63.2%	2.145	89.4%	1.047	98.4%
DSEC	1.972	74.2%	0.862	65.3%	1.710	93.3%	0.885	98.4%

Table 6. Per-scene normal flow evaluation on DSEC training set hold out.

% Pos	num_ensemble=2	num_ensemble=4	num_ensemble=6	num_ensemble=10
conf_thres=0.1	97.2%	97.7%	98.4%	98.2%
conf_thres=0.2	96.9%	98.0%	98.3%	98.4%
conf_thres=0.3	96.5%	97.7%	98.0%	98.1%
conf_thres=0.4	96.2%	97.4%	97.7%	97.9%
conf_thres=0.5	95.9%	97.2%	97.5%	97.7%
conf_thres=0.6	95.7%	96.9%	97.3%	97.5%
conf_thres=0.7	95.5%	96.7%	97.1%	97.2%
conf_thres=nfty	92.3%	92.6%	92.8%	92.8%

PEE	num_ensemble=2	num_ensemble=4	num_ensemble=6	num_ensemble=10
conf_thres=0.1	0.467	0.529	0.580	0.686
conf_thres=0.2	0.461	0.442	0.436	0.436
conf_thres=0.3	0.454	0.436	0.423	0.423
conf_thres=0.4	0.454	0.436	0.423	0.423
conf_thres=0.5	0.454	0.436	0.423	0.423
conf_thres=0.6	0.454	0.436	0.423	0.429
conf_thres=0.7	0.454	0.436	0.423	0.423
conf_thres=nfty	0.442	0.423	0.411	0.411

Valid Pct.	num_ensemble=2	num_ensemble=4	num_ensemble=6	num_ensemble=10
conf_thres=0.1	45.0%	17.4%	4.3%	1.6%
conf_thres=0.2	66.8%	53.3%	49.6%	47.9%
conf_thres=0.3	74.9%	66.6%	65.0%	63.3%
conf_thres=0.4	78.8%	72.2%	70.6%	68.8%
conf_thres=0.5	81.1%	75.1%	73.4%	71.5%
conf_thres=0.6	82.7%	77.0%	75.3%	73.6%
conf_thres=0.7	83.9%	78.5%	76.9%	76.8%
conf_thres=nfty	100%	100%	100%	100%

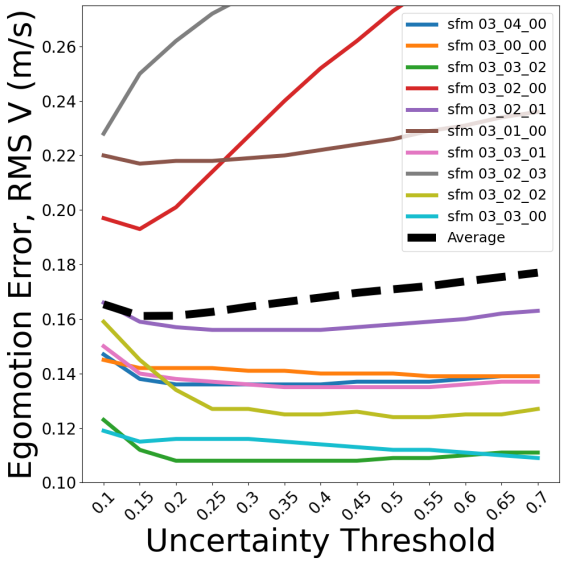

Figure 9. Effectiveness of uncertainty quantification. **Left**: The flow prediction errors are positively correlated with the uncertainty scores. **Right**: The egomotion estimation error is minimized by choosing a suitable threshold. Both findings highlight the effectiveness of the UQ.

	Inference Time (5 ensembles)	Max GPU Memory Allocation
num_events = 10k	0.111 s	0.70 GB
num_events = 20k	0.287 s	1.36 GB
num_events = 40k	0.910 s	2.71 GB
num_events = 80k	3.138 s	5.39 GB

Runtime to process 1 second of events. Tested on RTX3090	MVSEC outdoor_day_1	EVIMO imo-13-00	EVIMO sfm-03-00	DSEC thun_01_a	DSEC interlaken_00_a
median event density (events / 20 ms)	6900	35500	77000	101900	146400
Multi-CM (not parallelizable)	>30 minutes				
E-RAFT	7.5 s	40 s	40 s	40 s	40 s
TCM	6.8 s	35 s	35 s	35 s	35 s
Ours (5 ensembles, max 80k events)	5.9 s	30 s	80 s	150 s	150 s

	#events every 20 ms – quantile				
	min	25%	50%	75%	max
MVSEC – indoor_flying1	85	2396.25	3720.0	5376.0	16177
MVSEC – indoor_flying2	78	3157.0	5297.5	7987.5	23890
MVSEC – indoor_flying3	78	2746.0	4850.0	6845.0	17476
MVSEC – outdoor_day1	58	4412.0	6903.0	10646.75	96327
EVIMO – IMO_13_00	7954	23564.5	35535.0	46824.5	68946
EVIMO – IMO_13_05	10806	55951.0	78963.0	87891.0	120730
EVIMO – SFM_03_00	2962	15766.0	76979.0	88460.5	105177
EVIMO – SFM_03_01	15315	41049.0	80223.0	94096.0	118441
DSEC – interlaken_00_a	123449	133520.0	146397.0	157920.0	165356
DSEC – interlaken_00_b	183286	187122.75	189133.5	196445.75	209048
DSEC – thun_01_a	66476	83569.25	101858.5	117016.75	121001
DSEC – zurich_city_12_a	116746	138822.0	167154.0	196738.5	228827

Table 7. Runtime and memory cost of our estimator. **Upper Left**: runtime and memory cost when processing events of different densities. **Lower Left**: runtime comparison among existing methods. Multi-CM [46] relies on contrast maximization to estimate optical flow, which significantly increases its runtime when event density is high. In contrast, E-RAFT [21] and TCM [35] are frame-based methods, making their runtime largely independent of event density. The runtime of our point-based estimator, however, varies with event density. While it may run slower than frame-based methods at high event densities, it remains generally feasible even on entry-level GPUs. **Right**: Density statistics of scenes from different datasets.

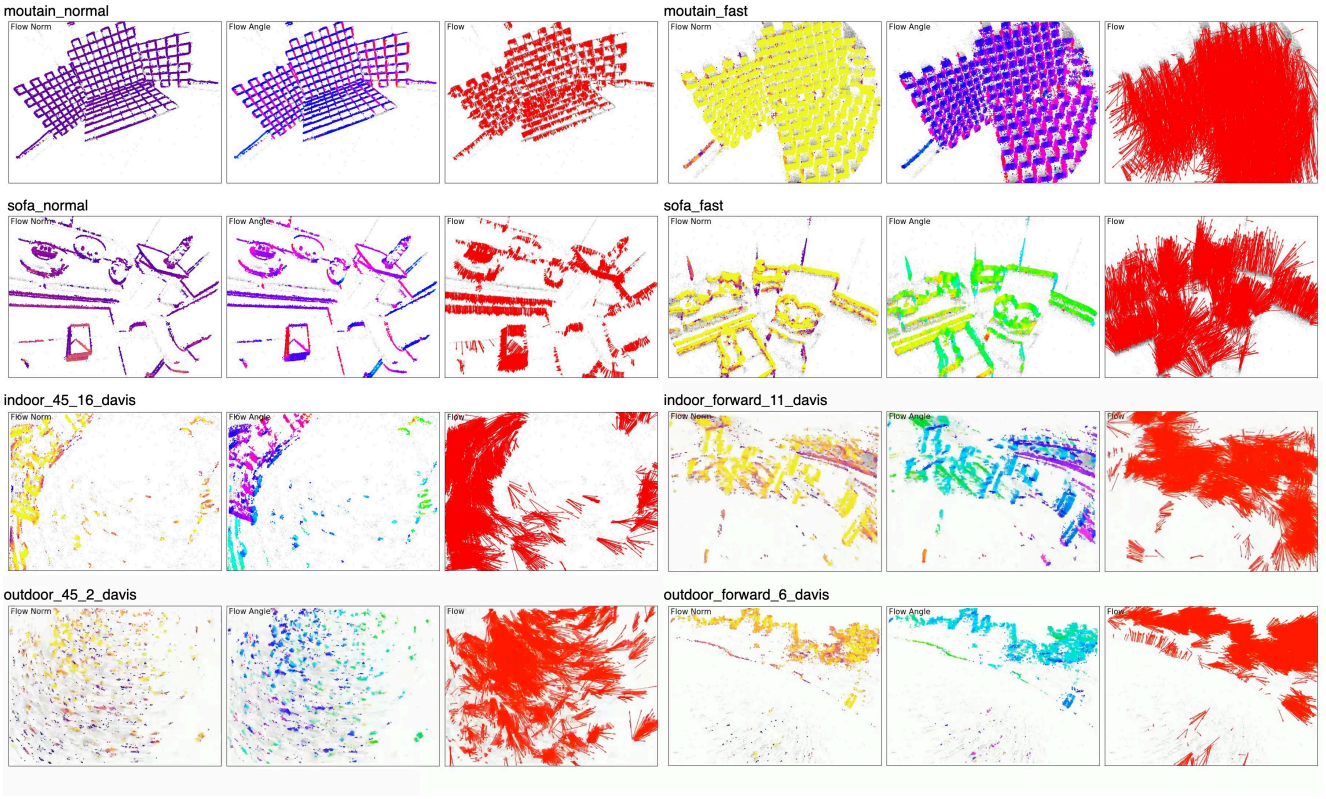

Figure 10. Qualitative evaluation on FPV and VECtor. The predictions are generated using a model trained on M+D+E, demonstrating the estimator's generalizability across diverse datasets. VECtor in general has more confident predictions than FPV since VECtor uses a higher resolution camera and produces denser events.

		Input	Training Set	Scene_13_00		Scene_13_05		Scene_14_03		Scene_14_04	
				PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑
MultiCM	MB	F	-	1.509	53.2%	4.315	75.7%	1.611	79.2%	1.800	73.2%
PCA	MB	P	-	1.573	88.2%	2.035	87.5%	1.580	91.9%	1.784	90.3%
E-RAFT	SL	F	M	1.370	71.9%	2.406	90.6%	1.356	69.5%	1.458	64.6%
			D	0.843	88.9%	1.185	97.5%	0.517	88.1%	0.538	85.9%
TCM	SSL	F	M	0.823	85.6%	3.201	95.3%	1.111	86.3%	1.532	86.0%
			D	0.774	87.3%	2.541	95.1%	0.872	87.8%	1.090	86.5%
PointNet	SL	P	E	1.047	88.1%	0.924	97.7%	0.848	98.3%	0.892	96.2%
Ours	SL	P	M	0.713	95.6%	0.269	99.3%	0.676	98.8%	0.651	98.1%
			D	0.590	96.6%	**0.230**	**99.8%**	0.575	**99.8%**	0.625	**99.5%**
			E	0.497	**96.7%**	0.399	99.2%	**0.478**	99.2%	0.515	98.8%
			M+D+E	**0.465**	96.2%	0.308	99.2%	0.544	99.3%	**0.467**	98.8%

			Training Set	scene_14_05		scene_15_01		scene_15_02		scene_15_05	
				PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑
MultiCM	MB	F	-	2.768	72.9%	0.852	68.0%	0.802	66.2%	0.744	59.8%
PCA	MB	P	-	1.823	89.4%	1.467	92.1%	1.612	78.2%	1.821	84.7%
ERAFT	SL	F	M	2.186	67.1%	0.899	72.7%	0.980	67.1%	1.100	57.9%
			D	0.908	86.3%	0.432	91.3%	0.541	90.9%	0.674	73.9%
TCM	SSL	F	M	2.445	82.2%	0.588	85.4%	0.556	87.7%	0.811	68.0%
			D	1.640	84.1%	0.523	85.5%	0.528	87.9%	0.871	68.1%
PointNet	SL	P	E	1.053	96.6%	0.765	96.1%	0.752	95.1%	1.185	91.5%
Ours	SL	P	M	0.806	98.2%	0.470	96.6%	0.433	95.8%	0.392	94.3%
			D	**0.567**	**99.4%**	0.391	**98.1%**	0.298	**97.1%**	0.424	93.2%
			E	0.584	98.6%	**0.286**	**98.1%**	**0.274**	96.8%	0.354	95.5%
			M+D+E	0.568	98.5%	0.319	97.8%	0.300	97.0%	**0.201**	**95.7%**

Table 8. Per-scene normal flow evaluation on EVIMO2-`imo` split.

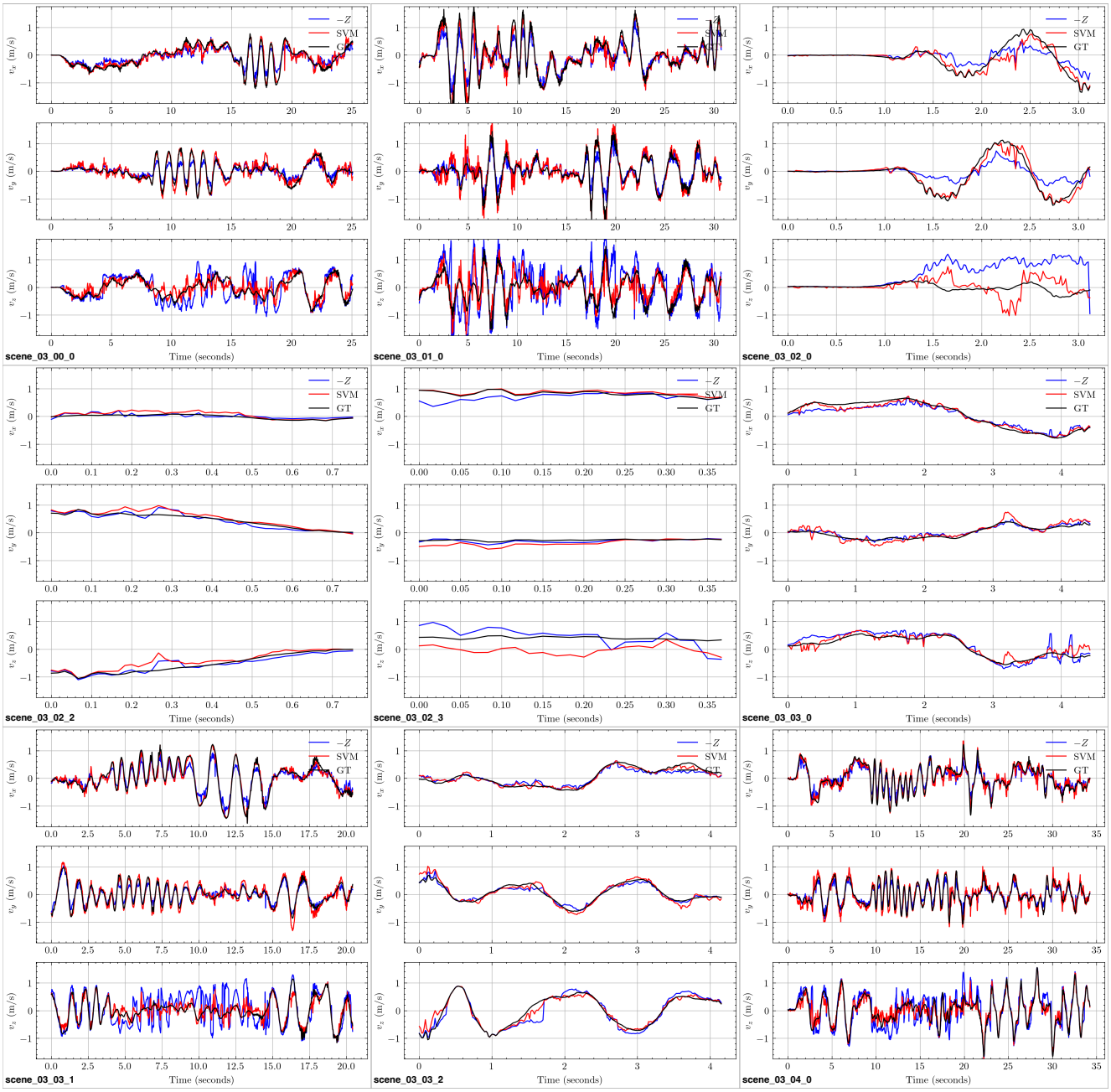

Figure 11. Per-scene egomotion evaluation on EVIMO2 `sfm` split.