Efficient Deep Visual and Inertial Odometry with Modality Selection (Supplementary Material)

Mingyu Yang*, Yu Chen*, and Hun-Seok Kim

University of Michigan, Ann Arbor MI 48109, USA
{mingyuy,unchenyu,hunseok}@umich.edu

1 Network Structure

We provide structure details of the visual encoder, the inertial encoder, the policy network, and the pose estimation network in Table 1, 2, 3, and 4 respectively. The batch size and sequence length are denoted by B and T. For the visual encoder, we adopt the FlowNetS structure proposed in [2], which is trained on the FlyingChairs dataset [2] for optical flow estimation. We adopt the pre-trained checkpoint '*flownets_bn_EPE2.459.pth.tar*' from `https://github.com/ClementPinard/FlowNetPytorch` as our initialization.

Table 1: Detailed structure of the visual encoder.

Input	$B \times 6 \times 512 \times 256$
Layer 1	Conv2d, Chan. 64, ker. 7^2, pad. 3^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 2	Conv2d, Chan. 128, ker. 5^2, pad. 2^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 3	Conv2d, Chan. 256, ker. 5^2, pad. 2^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 4	Conv2d, Chan. 256, ker. 3^2, pad. 1^2, stride 1, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 5	Conv2d, Chan. 512, ker. 3^2, pad. 1^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 6	Conv2d, Chan. 512, ker. 3^2, pad. 1^2, stride 1, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 7	Conv2d, Chan. 512, ker. 3^2, pad. 1^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 8	Conv2d, Chan. 512, ker. 3^2, pad. 1^2, stride 1, batchnorm, LeakyReLU 0.1, drop. 0.2
Layer 9	Conv2d, Chan. 1024, ker. 3^2, pad. 1^2, stride 2, batchnorm, LeakyReLU 0.1, drop. 0.5
Layer 10	FC, 32768×512
Output	$B \times 512$

Table 2: Detailed structure of the inertial encoder.

Input	$B \times 6 \times 11$
Layer 1	Conv1d, Chan. 64, ker. 3, pad. 1, stride 1, batchnorm, LeakyReLU 0.1
Layer 2	Conv1d, Chan. 128, ker. 3, pad. 1, stride 1, batchnorm, LeakyReLU 0.1
Layer 3	Conv1d, Chan. 256, ker. 3, pad. 1, stride 1, batchnorm, LeakyReLU 0.1
Layer 4	FC, 2816×256
Output	$B \times 256$

* Equally contributed first co-authors

Table 3: Detailed structure of the policy network

Input	$B \times 1280$
Layer 1	FC, 768×256, LeakyReLU 0.1, batchnorm
Layer 2	FC, 256×32, LeakyReLU 0.1, batchnorm
Layer 3	FC, 32×2
Layer 4	Gumbel-Softmax
Output	$B \times 1$

Table 4: Detailed structure of the pose estimation network

Input	$B \times T \times 768$
Layer 1	LSTM, layer 2, hidden size 1024, dropout 0.2
Layer 2	dropout 0.2
Layer 3	FC, 1024×128, LeakyReLU 0.1
Layer 4	FC, 128×6
Output	$B \times T \times 6$

2 Additional Data Processing and Training Details

The KITTI Odometry dataset does not come with IMU data. Thus, we extract the IMU data from the KITTI raw dataset and associate them with the KITTI Odometry dataset as shown in Table 5. The IMU data and image frames are not synchronized. Therefore, we apply linear interpolation to the raw IMU data to obtain data synchronized to the image time index. The training sub-sequences are extracted from the original long sequences with an overlap of 1 frame between sub-sequences. During training, we apply horizontal flipping to images with 50% probability and adjust the ground truth poses and IMU data accordingly. A weight decay of 5×10^{-6} is applied to the visual encoder, inertial encoder, and the pose network training to avoid overfitting.

Table 5: Correspondence between KITTI Odometry dataset and KITTI raw dataset

Nr.	Sequence name	Frames
00	2011_10_03_drive_0027	0000-4540
01	2011_10_03_drive_0042	0000-1100
02	2011_10_03_drive_0034	0000-4660
04	2011_09_30_drive_0016	0000-0270
05	2011_09_30_drive_0018	0000-2760
06	2011_09_30_drive_0020	0000-1100
07	2011_09_30_drive_0027	0000-1100
08	2011_09_30_drive_0028	1100-5170
09	2011_09_30_drive_0033	0000-1590
10	2011_09_30_drive_0034	0000-1200

3 FLOPS Computation

We list the FLOPS counting formula for each type of layers in Table 6. The total number of FLOPS of each individual network is shown in Table 7 where computations of non-linear activation functions are ignored. It is observed the visual encoder exhibits a 3700× higher FLOPS than that of the inertial encoder. The total number of FLOPS of a test sequence is calculated using multiple sampled policies and is then averaged over the total time (in seconds).

Table 6: FLOPS counting for each layer type.

Layer	Total number of FLOPS
2D Convolution	$C_o \times C_i \times k \times k \times \frac{H}{s} \times \frac{W}{s}$
1D Convolution	$C_o \times C_i \times k \times L$
2D BatchNorm	$C_o \times H \times W$
1D BatchNorm	$C_o \times L$
LSTM (each layer)	$4 \times (L + L_h + 1) \times L_h + 4 \times L_h$
Fully connected	$L_h \times L$

C_o, C_i: output & input channel,
k: kernel size, s: stride,
H, W: 2D input height & width,
L: 1D input length, L_h: hidden state size

Table 7: FLOPS counting for different networks.

Network	MFLOPS
Visual Encoder	7768.54
Inertial Encoder	2.09
Policy Network	0.336
Pose Regression LSTM	15.75

4 Complete Results for KITTI

In this section, we show the complete results for all KITTI test sequences (Sequence *05*, *07*, and *10*) in Table. 8, 9, and 10. For stochastic methods, we show both the mean and standard deviation.

5 Additional Qualitative Analysis on KITTI

We present additional visualizations of test paths. Figure 1 shows the full-modality baseline, two sub-optimal strategies (random sampling and regular

Table 8: Complete result for Sequence *05*

Method		$t_{rel}(\%)$	$r_{rel}(^\circ)$	Trans. RMSE (m)	Rot. RMSE (°)	Usage (%)
Regular Skipping	$n = 1$	2.61	1.06	0.0268	0.0578	100
	$n = 5$	3.23	1.20	0.0553	0.0608	20
	$n = 8$	5.13	1.87	0.0766	0.0755	12.5
Random Policy	$p = 0.2$	4.09 ± 0.11	1.49 ± 0.05	0.0487 ± 0.0028	0.0521 ± 0.0003	20.25
	$p = 0.125$	4.09 ± 0.20	0.99 ± 0.04	0.0764 ± 0.0050	0.0527 ± 0.0005	12.62
Proposed	$\lambda = 1 \times 10^{-5}$	2.15 ± 0.04	0.78 ± 0.01	0.0249 ± 0.0002	0.0483 ± 0.0002	60.3
	$\lambda = 3 \times 10^{-5}$	2.01 ± 0.05	0.75 ± 0.02	0.0294 ± 0.0007	0.0478 ± 0.0002	20.6
	$\lambda = 5 \times 10^{-5}$	2.71 ± 0.15	1.03 ± 0.06	0.0352 ± 0.0009	0.0509 ± 0.0007	11.34
	$\lambda = 7 \times 10^{-5}$	3.00 ± 0.21	1.20 ± 0.07	0.0443 ± 0.0010	0.0563 ± 0.0007	6.83

Table 9: Complete result for Sequence *07*

Method		$t_{rel}(\%)$	$r_{rel}(^\circ)$	Trans. RMSE (m)	Rot. RMSE (°)	Usage (%)
Regular Skipping	$n = 1$	1.83	1.35	0.0359	0.0587	100
	$n = 5$	2.84	0.67	0.0675	0.0696	20
	$n = 8$	4.15	2.15	0.0797	0.0861	12.5
Random Policy	$p = 0.2$	2.53 ± 0.25	1.16 ± 0.17	0.0527 ± 0.0033	0.0520 ± 0.0014	20.96
	$p = 0.125$	4.33 ± 0.68	1.33 ± 0.08	0.0792 ± 0.0100	0.0527 ± 0.0014	13
Proposed	$\lambda = 1 \times 10^{-5}$	2.25 ± 0.10	1.19 ± 0.04	0.0398 ± 0.0005	0.0457 ± 0.0003	63.35
	$\lambda = 3 \times 10^{-5}$	1.79 ± 0.11	0.76 ± 0.06	0.0424 ± 0.0017	0.0435 ± 0.0003	19.79
	$\lambda = 5 \times 10^{-5}$	2.22 ± 0.20	1.14 ± 0.09	0.0430 ± 0.0017	0.0478 ± 0.0005	10.57
	$\lambda = 7 \times 10^{-5}$	2.48 ± 0.15	1.60 ± 0.15	0.0528 ± 0.0029	0.0561 ± 0.0014	6.03

skipping), and our proposed method on Sequence *10*. Figure 2 - 6 contain the visual interpretation of our method on Sequence *05* and *10* with $\lambda = 5 \times 10^{-5}$, and Sequence *05*, *07*, and *10* with $\lambda = 3 \times 10^{-5}$. The visualization on Sequence *07* with $\lambda = 5 \times 10^{-5}$ is shown in the paper.

6 Test Results on EuRoC

We present the results of our method tested on the EuRoC dataset [1] for indoor scenarios. The EuRoC dataset conains 11 tightly-synchronized stereo videos and IMU meansurements collected by an Asctec Firefly hex-rotor helicopter. The ground truth poses are collected by a Leica MS50 laser tracker or Vicon 6D motion capture system. We select the sequence *MH_04_difficult* for testing and use the remaining sequences for training. The images and IMU data are sampled at 20Hz and 200Hz respectively. We use the monocular images from camera 1 of the EuRoC dataset. The system parameters and training procedure are the same as those used for KITTI, except for α which is set to 10 for EuRoC. Table 11 shows the results of the full-modality baseline and our proposed method with different λ's. Similar with KITTI, disabling the visual modality does not lead to significant performance degradation. It is observed that the best translational RMSE is achieved with $\lambda = 1 \times 10^{-6}$ and the best rotational RMSE with $\lambda = 3 \times 10^{-6}$.

Table 10: Complete result for Sequence *10*

Method		$t_{rel}(\%)$	$r_{rel}(^\circ)$	Trans. RMSE (m)	Rot. RMSE (°)	Usage (%)
Regular Skipping	$n = 1$	3.11	1.12	0.0438	0.0779	100
	$n = 5$	4.12	0.97	0.0746	0.0776	20
	$n = 8$	6.88	2.43	0.0937	0.0975	12.5
Random Policy	$p = 0.2$	2.70 ± 0.17	0.83 ± 0.07	0.0679 ± 0.0029	0.0611 ± 0.0003	20.02
	$p = 0.125$	4.83 ± 0.17	1.28 ± 0.03	0.0963 ± 0.0055	0.0603 ± 0.0007	12.45
Proposed	$\lambda = 1 \times 10^{-5}$	3.30 ± 0.11	0.94 ± 0.05	0.0443 ± 0.0006	0.0573 ± 0.0003	65.01
	$\lambda = 3 \times 10^{-5}$	3.41 ± 0.15	1.08 ± 0.03	0.0500 ± 0.0014	0.0573 ± 0.0003	22.68
	$\lambda = 5 \times 10^{-5}$	3.59 ± 0.15	1.20 ± 0.04	0.0651 ± 0.0019	0.0601 ± 0.0010	12.2
	$\lambda = 7 \times 10^{-5}$	3.67 ± 0.19	1.57 ± 0.03	0.0854 ± 0.0036	0.0651 ± 0.0009	7.68

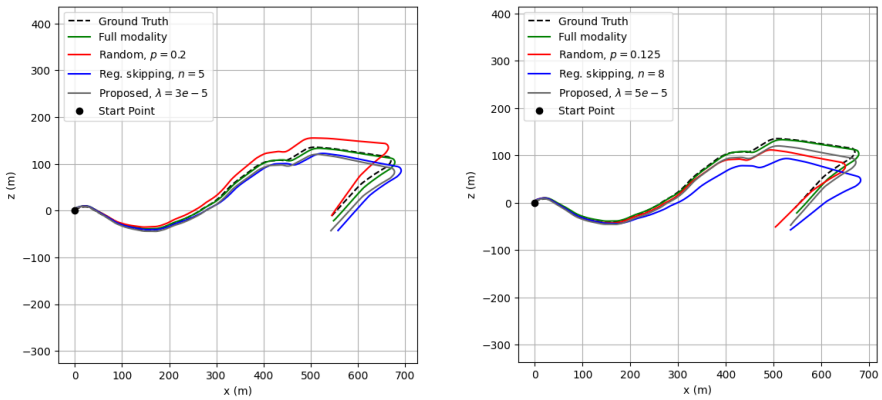

Fig. 1: Trajectories of ground truth, full modality baseline, random and regular skipping, and proposed method on KITTI Sequence *10*.

Table 11: Evaluation of the full-modality baseline and our proposed method with various penalty factor λ's on the EuRoC dataset. Due to the stochastic nature of our policy, we test our model with 10 different random seeds and show the average performance.

Method	Trans. RMSE (m)	Rot. RMSE (°)	Visual encoder usage	GFLOPS
Full Modality	0.0178	0.0906	100%	155.73
$\lambda = 1 \times 10^{-6}$	**0.0168**	0.0909	71.14%	110.89
$\lambda = 3 \times 10^{-6}$	0.0178	**0.0894**	41.67%	65.11
$\lambda = 5 \times 10^{-6}$	0.0187	0.0897	21.74%	34.14
$\lambda = 7 \times 10^{-6}$	0.0204	0.0895	**12.53%**	**19.83**

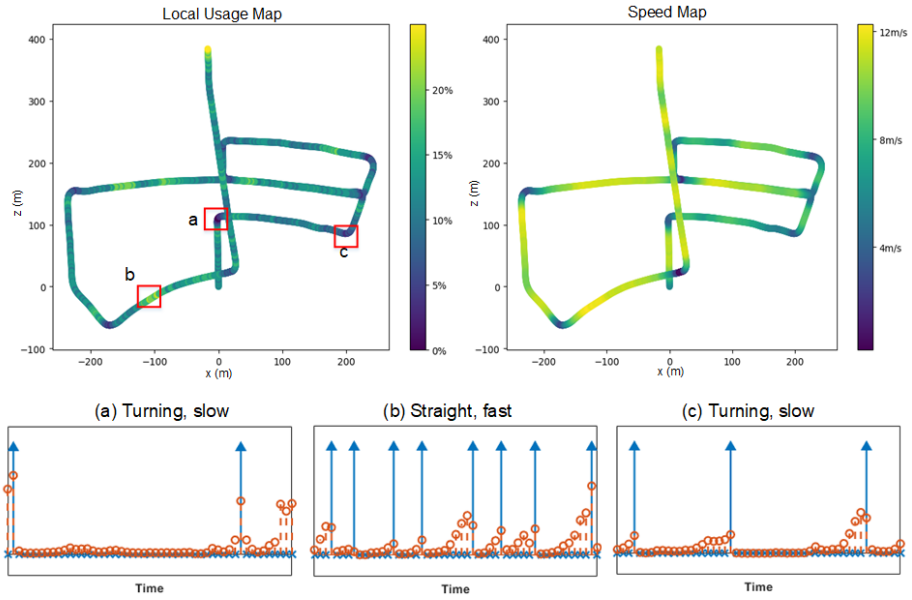

Fig. 2: Visual interpretation of the learned policy on Sequence *05* with $\lambda = 5 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed map is shown on the top right. We selected three short segments from the path to visualize the policy network's behavior by showing the decisions d_t (blue pulses) and probabilities p_t (orange circles) on the bottom for different scenarios.

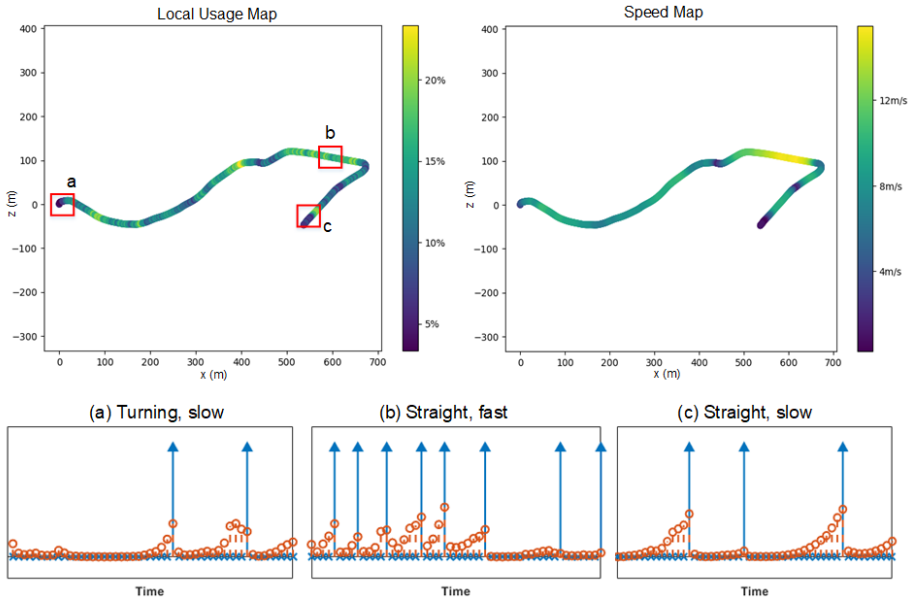

Fig. 3: Visual interpretation of the learned policy on Sequence *10* with $\lambda = 5 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed map is shown on the top right. We selected three short segments from the path to visualize the policy network's behavior by showing the decisions d_t (blue pulses) and probabilities p_t (orange circles) on the bottom for different scenarios.

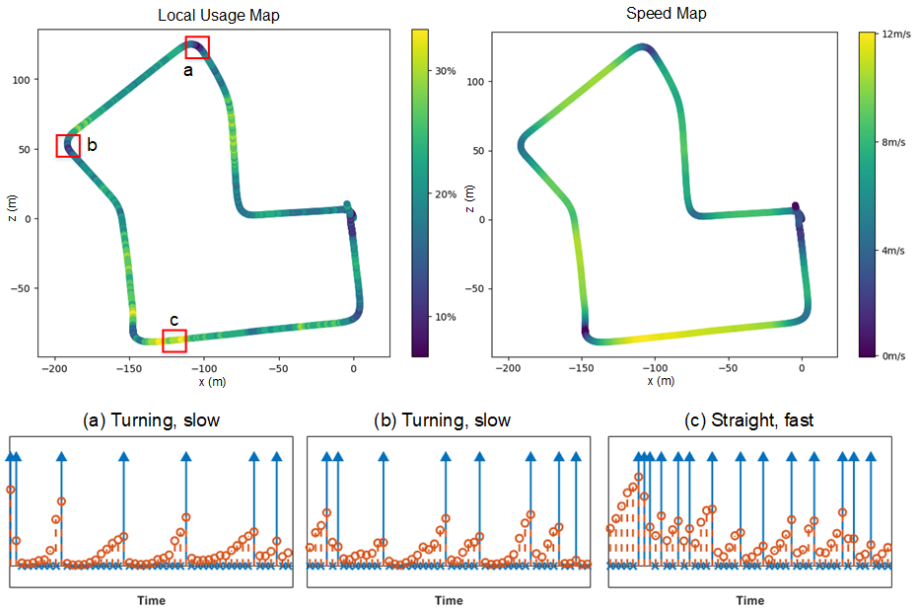

Fig. 4: Visual interpretation of the learned policy on Sequence *07* with $\lambda = 3 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed map is shown on the top right. We selected three short segments from the path to visualize the policy network's behavior by showing the decisions d_t (blue pulses) and probabilities p_t (orange circles) on the bottom for different scenarios.

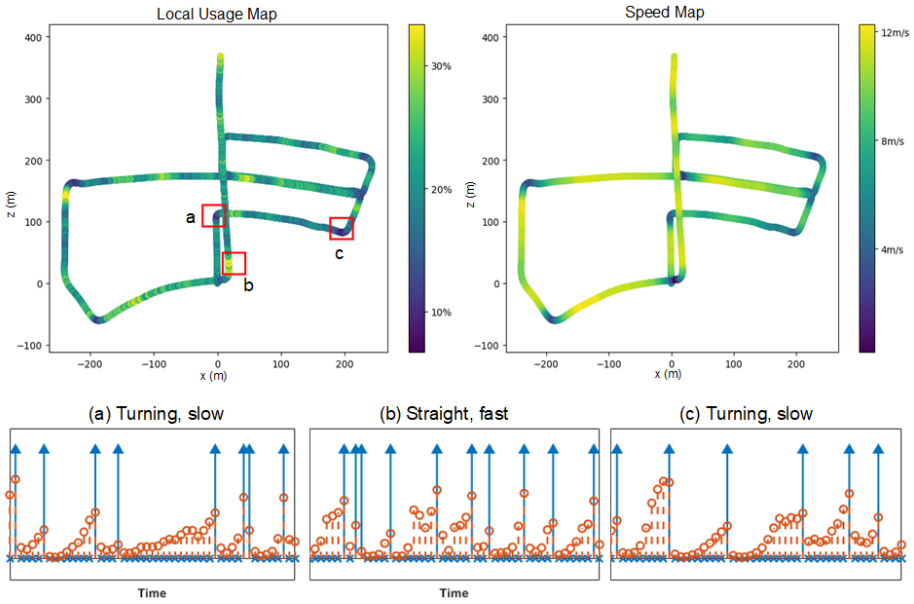

Fig. 5: Visual interpretation of the learned policy on Sequence *05* with $\lambda = 3 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed map is shown on the top right. We selected three short segments from the path to visualize the policy network's behavior by showing the decisions d_t (blue pulses) and probabilities p_t (orange circles) on the bottom for different scenarios.

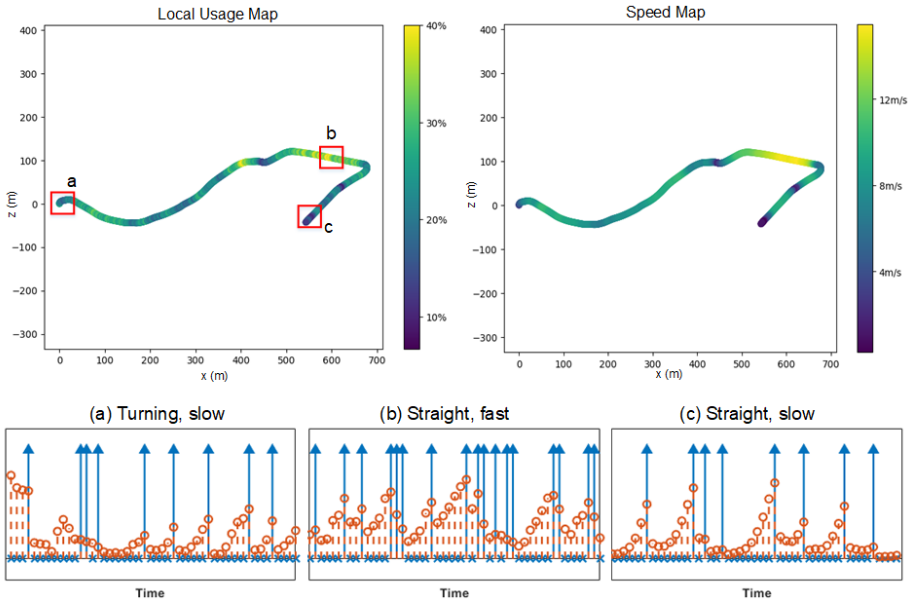

Fig. 6: Visual interpretation of the learned policy on Sequence *10* with $\lambda = 3 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed map is shown on the top right. We selected three short segments from the path to visualize the policy network's behavior by showing the decisions d_t (blue pulses) and probabilities p_t (orange circles) on the bottom for different scenarios.

References

1. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The euroc micro aerial vehicle datasets. The International Journal of Robotics Research **35**(10), 1157–1163 (2016)
2. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)