Efficient Deep Visual and Inertial Odometry with Adaptive Visual Modality Selection

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

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

Abstract. In recent years, deep learning-based approaches for visual-inertial odometry (VIO) have shown remarkable performance outperforming traditional geometric methods. Yet, all existing methods use both the visual and inertial measurements for every pose estimation incurring potential computational redundancy. While visual data processing is much more expensive than that for the inertial measurement unit (IMU), it may not always contribute to improving the pose estimation accuracy. In this paper, we propose an adaptive deep-learning based VIO method that reduces computational redundancy by opportunistically disabling the visual modality. Specifically, we train a policy network that learns to deactivate the visual feature extractor on the fly based on the current motion state and IMU readings. A Gumbel-Softmax trick is adopted to train the policy network to make the decision process differentiable for end-to-end system training. The learned strategy is interpretable, and it shows scenario-dependent decision patterns for adaptive complexity reduction. Experiment results show that our method achieves a similar or even better performance than the full-modality baseline with up to 78.8% computational complexity reduction for KITTI dataset evaluation. The code is available at `https://github.com/mingyuyng/Visual-Selective-VIO`.

Keywords: visual-inertial odometry, deep neural networks, long short-term memory, gumbel-softmax, adaptive learning

1 Introduction

Visual-inertial odometry (VIO) estimates the agent's self-motion using information collected from cameras and inertial measurement unit (IMU) sensors. With its wide applications in navigation and autonomous driving, VIO became one of the most important problems in the field of robotics and computer vision. Compared with visual odometry (VO) methods [3,9,10,30], VIO systems [24,34] incorporate additional IMU measurements and thus achieve more robust performance in texture-less environments and/or in extreme lightning conditions. However, classical VIO methods (not based on deep learning) rely heavily on manual interventions for system initialization and careful parameter tuning (e.g.,

* Equally contributed first co-authors

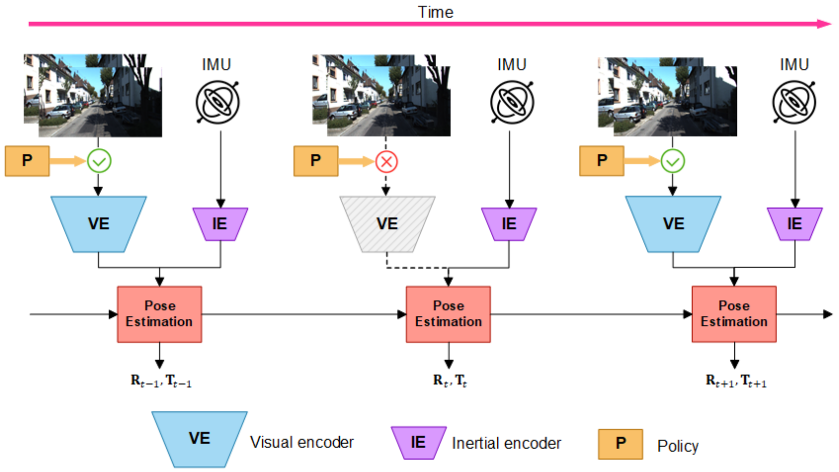

Fig. 1: An overview of our approach. For deep learning-based VIO methods, the computational cost of the visual encoder is much higher than that of the inertial encoder due to the difference in data dimension. Thus, rather than using images for every pose estimation, our method learns a policy that controls the usage of the visual encoder to avoid unnecessary image processing while maintaining a reasonable accuracy.

number of features per frame, threshold of feature matching, and keyframe selection) for each test environment. Besides, there are still significant challenges to deploying such systems with rapid calibration for fast-moving scenarios [53].

With the tremendous success of deep learning in various computer vision tasks [22,35,41], data-driven VIO methods [1,6,7,15,25,40] have drawn significant attention to the community, and they achieve competitive performance in both accuracy and robustness in challenging scenarios. Compared with classical geometric-based methods, these learning-based VIO solutions extract better features using deep neural networks (DNN). In addition, they can learn a better fusion mechanism between visual and inertial features to filter out abnormal sensor data while training on large-scale datasets. However, such learning-based methods typically have significant overhead in computation and power consumption, which is not affordable to energy-constrained mobile platforms operating with low-cost, energy-efficient cameras and IMU sensors.

Motivated by recent works that apply temporal adaptive inference to realize efficient action recognition [27,28,33,49] and fast text classification [4,16,39], we propose a new adaptive policy-based method to alleviate the high computational cost of deep learning-based VIO methods. The trained policy network opportunistically disables the visual (image) modality, as illustrated in Figure 1, to reduce the computational overhead when the visual features do not contribute significantly to the overall pose estimation accuracy. We choose to dynamically disable the visual modality while keeping IMU always available because the image encoder is much more computationally demanding than the inertial encoder due to their modality dimension difference. Thus, skipping the image processing significantly reduces the overall computational complexity. Besides, visual information is not always necessary for an accurate pose estimation, especially when the motion state does not vary much over time. Thus, occasionally skipping unimportant image inputs does not necessarily degrade the odometry accuracy. For our method, the proposed policy uses sampling from a Bernoulli distribution parameterized by the output of a light-weight policy network. We adopt the Gumbel-Softmax trick [20] to make the decision process differentiable. The model is trained to strike a balance between accuracy and efficiency with a joint loss. Our experiments demonstrate that our method significantly reduces computation (up to 78.8%) without compromising VIO accuracy. Thus, the proposed framework is suitable for mobile platforms with limited computation resources and energy budgets. Also, our method is modal-agnostic and can be applied to any visual and inertial encoders with different structures. Moreover, the learned policy is interpretable and yields scenario-dependent decision patterns in various test sequences.

Overall, our contributions are summarized as follows:

- We propose a novel method that adaptively disables the visual modality on the fly for efficient deep learning-based VIO. To the best of our knowledge, we are the first to demonstrate such a system reducing the complexity and energy consumption of deep learning-based VIO.
- A novel policy network is jointly trained with a pose estimation network to learn a visual modality selection strategy to enable or disable a visual feature extractor based on the motion state and IMU measurements. We adopt a Gumbel-Softmax trick to make the end-to-end system differentiable.
- The proposed method is tested extensively on the KITTI Odometry dataset. Experiments show that our approach achieves up to 78.8% computation reduction without noticeable performance degradation. Furthermore, we show that the learned policy exhibits an interpretable behavior that depends on motion states and patterns.

2 Related Works

2.1 Visual-inertial odometry

Visual odometry (VO) is a process to estimate ego-motion from sequential camera images [32], and it is extended to visual inertial odometry (VIO) including an IMU as an additional input. The datapath of conventional schemes typically consists of the following steps: feature detection, feature matching and tracking, motion estimation, and local optimization [38]. The VO/VIO system can be integrated into a simultaneous localization and mapping (SLAM) system [30,31,34] by performing additional steps of 3D environment mapping, global optimization, and loop closure. The performance of conventional VIO/SLAM systems is largely affected by visual feature matching and tracking accuracy, and the sensor fusion strategy. Hence, identifying superior handcrafted feature descriptors [26,37], adaptive filtering [24] or nonlinear optimization [17,23] based sensor fusion schemes are key challenges of such methods.

In recent years, deep learning-based methods have achieved remarkable successes on various computer vision applications, including VIO. VINet [7] is the first end-to-end trainable deep learning-based VIO where a DNN learns pose regression from the sequence of images and IMU measurements in a supervised manner. A long short-term memory (LSTM) network is introduced in VINet to model the temporal motion correlation. Later, Chen et al. [6] propose two different masking techniques that selectively fuse the visual and inertial features. ATVIO [25] introduces an attention-based fusion function and uses an adaptive loss for pose regression. Some recent works also propose to learn the 6-DoF ego-motion through a self-supervised learning framework that does not require ground-truth annotations during training. Shamwell et al. [40] introduce VIOLearner that estimates the poses through a view-synthesis approach with multi-level error correction. DeepVIO [15] improves VIO poses by additional self-supervision of optical flow, and similarly Almalioglu et al. [1] demonstrate a self-supervised VIO based on depth estimation [13, 55].

Whereas these prior works always rely on both the visual and inertial modality for each pose estimation, we propose a new framework to save the computation and power consumption overhead by opportunistically disabling the visual modality based on a learned strategy.

2.2 Adaptive inference

An adaptive inference scheme dynamically allocates computing resources based on each task input instance to minimize the redundant computation for relatively ‘easy’ task inputs. Several techniques for adaptive inference have been proposed including early exiting [2,19,42], layer skipping [14,43,45], and dynamic channel pruning [18,52,54]. Recently, the idea of adaptive inference has been extend to sequential data (e.g., text [4,16,39] and videos [27,28,33,49]) that are processed by recurrent neural networks (RNNs). Our technique is closely related to adaptive video recognition first proposed in [49], which introduces a memory-augmented LSTM to select only the relevant frames for efficient action recognition by training with a policy gradient method. Similarly, AR-Net [27] learns a policy that dynamically selects more relevant image frames and also adjusts their resolutions. The training in AR-Net is simplified using the Gumbel-Softmax trick. Later, this idea was extended to adaptively selecting a proper modality [33] or patches [46]. Our approach is motivated by these prior works to apply a similar framework to adaptive computation on deep learn-based VIO for the first time. We formulate it as a discrete-time pose regression problem that produces a pose estimation for every time step.

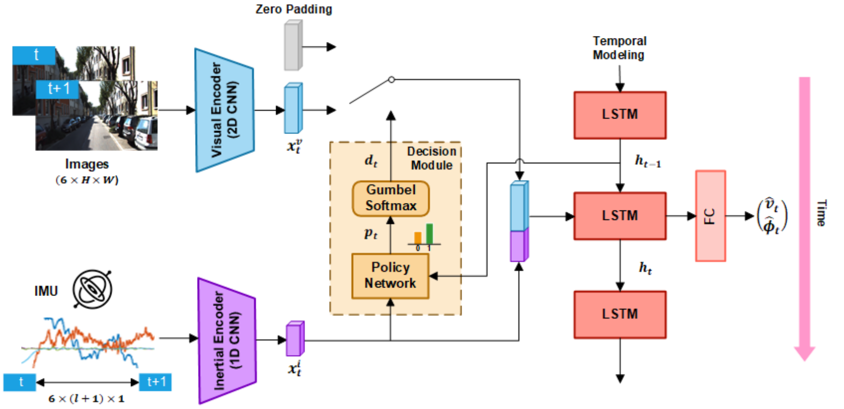

Fig. 2: Illustration of the proposed framework. At each time step, the policy network takes the inertial features and the previous hidden state vector to decide whether to use the visual modality or not. Once the policy network decides to use the visual modality, the current image is passed through the visual encoder, and the corresponding visual features are fed to the pose regression LSTM together with the inertial features for pose estimation. Otherwise, the visual encoder is disabled to save computations and LSTM input is zero padded. The decisions are sampled from a Gumbel-Softmax distribution during training to make the system end-to-end differentiable. During inference, the decision is sampled from a Bernoulli distribution controlled by the policy network.

3 Method

The inputs for VIO are the monocular video frames $\{V_i\}_{i=1}^{N}$, IMU measurements $\{I_i\}_{i=1}^{Nl}$ captured with a sampling frequency l times higher than the video frame rate, and the initial camera pose P_1. The goal of VIO is to estimate the camera poses $\{P_i\}_{i=2}^{N}$ for the entire path where $V_i \in \mathbb{R}^{3\times H\times W}$, $I_i \in \mathbb{R}^6$, and $P_i \in \mathbf{SE}(3)$. One typical way to perform VIO is to estimate the 6-DoF relative pose $T_{t\to t+1}$ that satisfies $P_t T_{t\to t+1} = P_{t+1}$ using two consecutive images $V_{t\to t+1} = \{V_t, V_{t+1}\}$ and a set of IMU measurements $I_{t\to t+1} = \{I_{tl}, \dots, I_{(t+1)l}\}$ for the time index $t = 1, 2, \dots, N-1$. The relative pose $T_{t\to t+1}$ can be further decomposed into a rotational vector $\phi_t \in \mathbb{R}^3$ containing Eular angles and a translational vector $v_t \in \mathbb{R}^3$. Our method learns a selection strategy that opportunistically skips the visual information $V_{t\to t+1}$ to reduce the computational complexity while maintaining the relative pose estimation accuracy.

3.1 End-to-end neural visual-inertial odometry

End-to-end neural VIO methods [6, 7, 25] consist of a visual feature encoder E_{visual} and an inertial feature encoder $E_{inertial}$ that extracts learned features from the input images and IMU measurements as follows:

$$x_t^v = E_{visual}(V_{t\to t+1}), \quad x_t^i = E_{inertial}(I_{t\to t+1}). \tag{1}$$

Typically, the visual feature encoder is much larger than the inertial feature encoder as the image dimension is much larger than that of the IMU measurement.

Visual feature x_t^v and inertial feature x_t^i are combined as z_t through concatenation [7] or attention modules [6, 25]. For accurate pose estimation, estimated motions and states of previous frames are used together with the newly extracted features of the current frame. Because of this temporally sequential nature of the problem, an RNN is typically employed to learn the correlation within the sequence of motions. The RNN employs fully connected layers as the last step for the final 6-DoF pose regression as in:

$$(h_t, \hat{\phi}_t, \hat{v}_t) = \mathrm{RNN}(z_t, h_{t-1}), \tag{2}$$

where h_{t-1} and h_t are the hidden latent vectors of the RNN at time t and $t-1$. $\hat{\phi}_t$ and $\hat{v}_t$ denotes the estimated rotational vector and translational vector, respectively.

3.2 Deep VIO with visual modality selection

The overview of our proposed method is illustrated in Figure 2. As an adaptive method, we aim to learn a binary decision d_t to determine whether the visual modality is not necessary and can be disabled without significant pose estimation accuracy degradation. We introduce a decision module where the decision d_t is sampled from a Bernoulli distribution whose probability p_t is generated by a light-weight policy network Φ. The policy network takes the current IMU features x_t^i and the last hidden latent vector h_{t-1} that contains the history information as the input. Thus, we have

$$p_t = \Phi(h_{t-1}, x_i^t), \tag{3}$$

where $p_t \in \mathbb{R}^2$ denotes the probability of the Bernoulli distribution. To make the system end-to-end trainable, we sample the binary decision $d_t \in \{0, 1\}$ via the Gumbel-Softmax operation,

$$d_t \sim \mathrm{GUMBEL}(p_t). \tag{4}$$

The detail of training with Gumbel-Softmax is discussed in the next section. When $d_t = 1$, visual features are enabled and the combined feature is obtained by concatenation of visual features and inertial features. On the other hand, when $d_t = 0$, visual features are disabled thus we apply zero padding to replace visual features to keep the same input dimension for the following RNN. This can be expressed as:

$$z_t = \begin{cases} x_t^v \oplus x_t^i & \text{if } d_t = 1 \\ \mathbf{0} \oplus x_t^i & \text{otherwise} \end{cases}, \tag{5}$$

where $\oplus$ denotes the concatenation operation. The combined feature z_t is then fed to the RNN that produces the estimated pose outputs ($\hat{\phi}_t$ and $\hat{v}_t$) via regression as in equation (2). In this paper, we adopt a two-layer LSTM for the pose estimation RNN.

3.3 Training with Gumbel-Softmax

Sampled d_t that follows a Bernoulli distribution is discrete in nature and it makes the network non-differentiable. Thus, it is not trivial to train the policy network through back-propagation. One common choice is to use a score function estimator (e.g., REINFORCE [12,47]) to estimate the gradient through the 'log-derivative trick'. However, that approach often has issues with slow convergence and high variance [48] for many applications. As an alternative, we adopt the Gumbel-Softmax scheme [20] to resolve non-differentiability by sampling from a corresponding Gumbel-Softmax distribution, which is essentially a reparametrization trick for categorical distributions [21,29,36]. Though reparameterization tricks may be less general than score function estimators, they usually exhibit several advantages such as lower variance and easier implementation.

Consider a categorical distribution where the probability for the k_{th} category is p_k for $k = 1, ..., K$. Then, following the Gumbel-Max trick [20], a discrete sample $\hat{P}$ that follows the target distribution can be drawn by:

$$\hat{P} = \underset{k}{\operatorname{argmax}}(\log p_k + g_k), \quad k \in [1, 2, ..., K], \tag{6}$$

where $g_k = -\log(-\log U_k)$ is a standard Gumbel distribution with a random variable U_k sampled from a uniform distribution $U(0, 1)$. Later, the softmax function is applied to relax the argmax operation to obtain a real-valued vector $\tilde{P} \in \mathbb{R}^K$ by a differentiable function as in

$$\tilde{P}_k = \frac{\exp((\log p_k + g_k)/\tau)}{\sum_{j=1}^{K} \exp((\log p_j + g_j)/\tau)}, \quad k = 1, 2, ..., K, \tag{7}$$

where τ is a temperature parameter that controls the 'discreteness' of $\tilde{P}$. When τ goes to infinity, $\tilde{P}$ tends to be a uniformly distributed vector, whereas $\tau \approx 0$ makes $\tilde{P}$ close to a one-hot vector and indistinguishable from the discrete distribution. In our case, we only have two categories $K = 2$ since we are dealing with a binary decision. During training, we sample the policy from the target Bernoulli distribution through (6) for the forward pass whereas the continuous relaxation (7) is used for the backward pass to approximate the gradient.

3.4 Loss function

During training, we apply the mean squared error (MSE) loss to reduce the pose estimation error given by:

$$\mathcal{L}_{pose} = \frac{1}{T-1} \sum_{t=1}^{T-1} (\|\hat{v}_t - v_t\|_2^2 + \alpha \|\hat{\phi}_t - \phi_t\|_2^2), \tag{8}$$

where T is the sequence length of training. v_t and ϕ_t denote the ground-truth translational and rotational vectors. α is a weight to balance the translational loss and rotational loss. We set $\alpha = 100$ as in the setting in prior supervised learning VO/VIO methods [6, 7, 25, 44, 51].

Besides, we apply an additional penalty factor λ to every visual encoder usage to encourage disabling visual feature computations. During the training, we calculate the averaged penalty and denote it as the efficiency loss defined by:

$$\mathcal{L}_{eff} = \frac{1}{T-1} \sum_{t=1}^{T-1} \lambda d_t. \tag{9}$$

Finally, the end-to-end system is trained with the summation of the pose estimation loss and efficiency loss (10) to strike a balance between good accuracy and computational efficiency.

$$\mathcal{L} = \mathcal{L}_{pose} + \mathcal{L}_{eff} \tag{10}$$

4 Experiments

In this section, we conduct an ablation study on the penalty factor to compare the proposed adaptive scheme with the full-modality baseline that always uses visual features. Results in this section will show that our proposed visual modality selection strategy can significantly reduces computational overhead while maintaining a similar or better accuracy compared to the full-modality baseline.

4.1 Experiment Setup

Dataset We evaluate our approach on KITTI Odometry dataset [11], which is one of the most influential VO/VIO benchmarks. The KITTI Odometry dataset consists of 22 sequences of stereo videos, where Sequence *00-10* contain the ground-truth trajectory and Sequence *11-22* exclude the ground-truth for evaluation. Following the procedure in [6], we train our model with Sequence *00, 01, 02, 04, 06, 08, 09* and test with Sequence *05, 07*, and *10*. We exclude Sequence *03* because of the lack of the raw IMU data. The images and ground-truth poses are recorded at 10 Hz and the IMU data is recorded at 100 Hz. The IMU data and images are not strictly synchronized. Thus, we interpolate the raw IMU data to time-synchronize it with the images and ground-truth poses. We use the monocular images from the left camera of KITTI Odometry dataset.

Implementation Details During training, we resize all images to 512 × 256 and set the training subsequence length to 11. We have 11 IMU measurements between every two consecutive images and thus the dimension of the input IMU data is 6 × 11. For the visual encoder, we adopt the FlowNet-S network [8] (except for the last layer) pretrained on the FlyingChairs dataset [8] for optical flow estimation. A fully connected layer is attached at the end of the network to produce a visual feature of length 512. The inertial encoder contains three 1D-convolutional layers and a fully connected layer to generate the inertial feature

Table 1: Evaluation of the full-modality baseline and our proposed method with various penalty factors λ on the KITTI 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	**0.0355**	0.0648	100%	77.87
$\lambda = 1 \times 10^{-5}$	0.0364	0.0505	62.89%	49.04
$\lambda = 3 \times 10^{-5}$	0.0406	**0.0495**	21.02%	16.51
$\lambda = 5 \times 10^{-5}$	0.0477	0.0529	11.37%	9.02
$\lambda = 7 \times 10^{-5}$	0.0609	0.0592	**6.85%**	**5.50**

Table 2: The relative translational & rotational error, and visual encoder usage of the baseline model and our proposed method with different penalty factors λ on Sequence 05, 07, and 10. The results are averaged over 10 tests with different seeds. The last column also shows the standard deviation to quantify the stability.

Method	Seq. 05			Seq. 07			Seq. 10			Average		
	t_{rel}	r_{rel}	Usage	t_{rel}	r_{rel}	Usage	t_{rel}	r_{rel}	Usage	t_{rel}	r_{rel}	Usage
Full Modality	2.61	1.06	100%	1.83	1.35	100%	**3.11**	1.12	100%	2.52	1.18	100%
$\lambda = 1 \times 10^{-5}$	2.15	0.78	60.30%	2.25	1.19	63.35%	3.30	**0.94**	65.01%	2.57 ± 0.052	0.97 ± 0.018	62.89%
$\lambda = 3 \times 10^{-5}$	**2.01**	**0.75**	20.60%	**1.79**	**0.76**	19.79%	3.41	1.08	22.68%	**2.40 ± 0.064**	**0.86 ± 0.018**	21.02%
$\lambda = 5 \times 10^{-5}$	2.71	1.03	11.34%	2.22	1.14	10.57%	3.59	1.20	12.20%	2.84 ± 0.102	1.13 ± 0.045	11.37%
$\lambda = 7 \times 10^{-5}$	3.00	1.20	**6.83%**	2.48	1.60	**6.03%**	3.67	1.57	**7.68%**	3.05 ± 0.086	1.46 ± 0.046	**6.85%**

■ t_{rel} and r_{rel} are the average translational error (%) and average rotational error (°/100m) obtained from various segment lengths of 100m – 800m.

of size 256. The pose estimation network contains a two-layer LSTM each with 1024 hidden units. At each time step, the hidden state of the last LSTM layer is passed through a two-layer multi-layer perceptron (MLP) to estimate the 6-DoF pose. The policy network is designed with a light-weight three-layer MLP.

The training process consists of two stages: warm-up stage and joint-training stage. In the warm-up stage, we train the visual encoder, inertial encoder, and the pose estimation network for 40 epochs with a random policy where we have a 50% chance to use the visual encoder at each time step. The learning rate is set to 5×10^{-4} in this stage. Next, in the joint-training stage, we train all end-to-end components including the policy network for 40 epochs with a learning rate of 5×10^{-5}, and then decrease the learning rate to 1×10^{-6} for additional 20 epochs. We set the initial temperature of Gumbel-Softmax to 5 and apply exponential decaying for each epoch with a factor of -0.05. We use Adam optimization with $\alpha = 0.9$ and $\beta = 0.999$, and the batch size is set to 16. During training, we always use the visual modality for the first frame to guarantee a qualified initial pose estimation. Similarly, during inference, we always enable the visual modality for the first pose estimation before we run the policy network without intervention for the rest of the path. Although the sequence length for training is set to 11, our method can run on any length of inputs for the inference.

Metric We calculate the root mean square error (RMSE) for the estimated translational vectors $\{\hat{v}_t\}_{t=1}^{N-1}$ and rotational vectors $\{\hat{\phi}_t\}_{t=1}^{N-1}$ of the entire path (i.e., $\sqrt{\frac{1}{N-1}\sum_{t=1}^{N-1}\|\hat{v}_t - v_t\|_2^2}$ and $\sqrt{\frac{1}{N-1}\sum_{t=1}^{N-1}\|\hat{\phi}_t - \phi_t\|_2^2}$). We also evaluate the relative translation/rotation error denoted by t_{rel} and r_{rel} for various subse-

Table 3: Comparison with two sub-optimal policies (regular skipping and random sampling) that use a similar visual encoder usage.

Method	params	$t_{rel}(\%)$	$r_{rel}(^\circ)$	Visual encoder usage
Policy network	$\lambda = 3 \times 10^{-5}$	2.40 ± 0.064	0.86 ± 0.018	21.02%
	$\lambda = 5 \times 10^{-5}$	2.84 ± 0.102	1.13 ± 0.045	11.37%
Regular skipping	$n = 5$	3.40	0.95	20%
	$n = 8$	5.39	2.15	12.5%
Random policy	$p = 0.2$	3.11 ± 0.11	1.16 ± 0.073	20.41%
	$p = 0.125$	4.42 ± 0.239	1.2 ± 0.027	12.69%

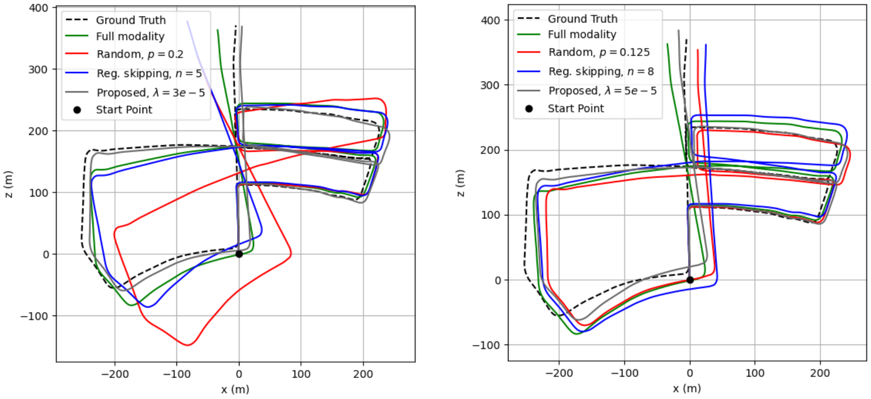

Fig. 3: Trajectories of ground-truth, full modality baseline, random and regular skipping, and proposed method on KITTI Sequence *05*.

quence path lengths such as 100, 200, ..., 800 meters as in [11]. To evaluate our policy network, we calculate the average usage rate of the visual modality and GFLOPS (giga floating-point operations per second).

4.2 Main results

Ablation study on the penalty factor We first test our method on KITTI using four different penalty factors: $1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}$and 7×10^{-5} to compare with the full modality baseline. For a fair comparison, we train the proposed and baseline full modality models with the same optimizer and common hyperparameters including the number of epochs and learning rate. Since our method is non-deterministic with a random sampling process, we test our model with 10 different random seeds and show the average performance. In Table 1, we present the average usage rate of the visual encoder, average GFLOPS, and average translational and rotational RMSE. It is observed that, as we gradually increase the penalty factor λ, both the usage of the visual encoder and system GFLOPS decrease as expected. In the meantime, as the visual encoder usage (and GFLOPS) drops, the translational RMSE becomes monotonically worse while the rotational RMSE does not show a monotonic behavior. This indicates that visual features do not necessarily always contribute to improving rotation

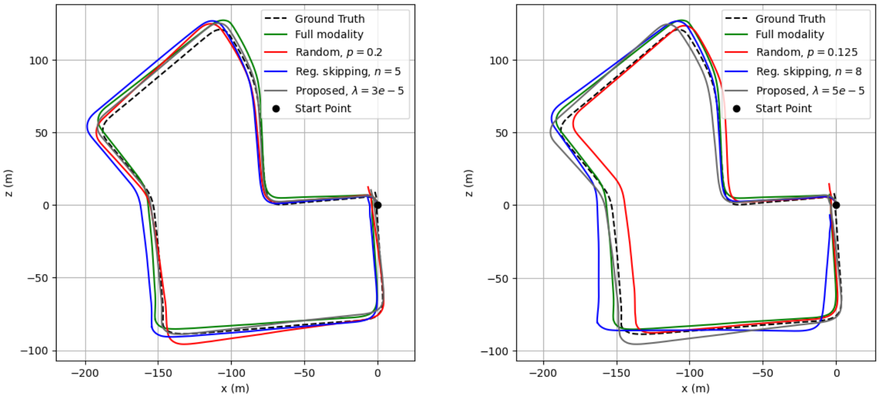

Fig. 4: Trajectories of ground-truth, full modality baseline, random and regular skipping, and proposed method on KITTI Sequence *07*.

estimation accuracy. A particular setting of $\lambda = 3 \times 10^{-5}$ provides 78.8% reduction in GFLOPS at the cost of a relatively small 14.3% loss in translational RMSE while improving rotational RMSE by 23.6%. We also conduct evaluations of t_{rel} and r_{rel} obtained from various subsequent path lengths and report the results in Table 2. Similarly, our method achieves comparable accuracy to the fully modality baseline with $\lambda = 1 \times 10^{-5}$ and achieves an even better result with $\lambda = 3 \times 10^{-5}$ which results in 78.8% lower GFLOPS. Very aggressive policy network settings at $\lambda = 5 \times 10^{-5}$ and $\lambda = 7 \times 10^{-5}$ experience mild performance degradation. Note that the standard deviation shown in the last column remains quite small, demonstrating the stability of our proposed method.

Comparison with sub-optimal selection strategies In this section, we compare our proposed method with two sub-optimal visual modality selection strategies: regular skipping and random sampling. For regular skipping, we train the model with a fixed selection pattern where the visual encoder is enabled every n time indices. For random sampling, the visual modality is enabled with probability of p for each time index. These two methods are trained with the same number of epochs, optimizer, learning rate decaying strategy, and the other hyperparameters as in our proposed method. For each penalty factor λ applied to our method, we carefully choose a corresponding skipping rate parameter n and probability p such that all methods share a similar visual encoder usage. Table 3 shows our method significantly outperforms those two sub-optimal policies especially for t_{rel}. We also plot the path trajectories based on estimated poses from all methods on Sequence *05* in Figure 3 and Sequence *07* in Figure 4 for comparison. The proposed method exhibits the most reliable trajectory among all evaluated policies.

Table 4: Comparison with prior VO/VIO works in translational & rotational error and image usage. The best performance in each block is marked in **bold**. Loop closure is excluded for ORB-SLAM2 and VINS-Mono.

	Method	Seq. 05			Seq. 07			Seq. 10		
		t_{rel}(%)	r_{rel}(°)	Usage	t_{rel}(%)	r_{rel}(°)	Usage	t_{rel}(%)	r_{rel}(°)	Usage
Geo.	ORB-SLAM2*[31]	**9.12**	**0.2**	100%	10.34	**0.3**	100%	**4.04**	**0.3**	100%
	VINS-Mono†[34]	11.6	1.26	100%	**10.0**	1.72	100%	16.5	2.34	100%
Self-Sup.	Monodepth2 *[13]	4.66	1.7	100%	4.58	2.6	100%	7.73	3.4	100%
	Zou et.al.*[56]	**2.63**	**0.5**	100%	6.43	2.1	100%	5.81	1.8	100%
	VIOLearner†[40]	3.00	1.40	100%	3.60	2.06	100%	2.04	1.37	100%
	DeepVIO†[15]	2.86	2.32	100%	**2.71**	**1.66**	100%	**0.85**	**1.03**	100%
Sup.	GFS-VO*[50]	3.27	1.6	100%	3.37	2.2	100%	6.32	2.3	100%
	BeyondTracking*[51]	2.59	1.2	100%	3.07	1.8	100%	3.94	1.7	100%
	ATVIO†[25]	4.93	2.4	100%	3.78	2.59	100%	5.71	2.96	100%
	Soft Fusion†[5]	4.44	1.69	100%	2.95	1.32	100%	3.41	1.41	100%
	Hard Fusion†[5]	4.11	1.49	100%	3.44	1.86	100%	**1.51**	**0.91**	100%
	(Ours) baseline†	2.61	1.06	100%	1.83	1.35	100%	3.11	1.12	100%
	(Ours) $\lambda = 3 \times 10^{-5}$†	**2.01**	**0.75**	20.6%	**1.79**	**0.76**	19.79%	3.41	1.08	22.68%
	(Ours) $\lambda = 5 \times 10^{-5}$†	2.71	1.03	**11.34%**	2.22	1.14	**10.57%**	3.59	1.20	**12.2%**

*: Visual Odometry, †: Visual-Inertial Odometry

Comparison with other VO/VIO baselines Now, we compare our method with geometric (non-learning-based) methods such as ORB-SLAM2 [31] and VINS-Mono [34] without loop closure, and also with state-of-the-art deep learning-based VO/VIO methods. Among those, deep learning-based self-supervised methods are [13, 15, 40, 56], and supervised learning methods are [5, 25, 50, 51]. All self-supervised methods are trained on Sequence *00-08* and tested on *09-10*. Among supervised methods, [50] and [51] are trained on Sequence *00, 02, 08, 09*. The other methods use the same training set as ours. It can be seen that although our main goal is not necessarily maximizing the odometry accuracy, our method still achieves the best performance among all the supervised methods. Compared with the state-of-the-art self-supervised methods [15, 40] (which are known to outperform supervised methods in general), our method achieves a competitive performance especially for Sequence *05* and *07* that belong to their training set. This demonstrates the robustness of our policy network and also the effectiveness of our network structure and training strategy.

Interpretation of the learned policy In Figure 5, we present the visual interpretation of our learned policy evaluated on Sequence *07* with $\lambda = 5 \times 10^{-5}$. On the top left, we plot the local visual encoder usage with a color coding that represents the visual modality usage rate for a local window of 31 frames. Darker (lighter) colors represent lower (higher) usages. On the top right, we show the speed of the agent (a vehicle) at each time step where darker colors represent lower speed. An obvious correlation is observed between the visual modality usage and the speed which is also correlated with the turning angle. When the agent is moving slowly or making a turn, the policy network utilizes the visual modality less frequently. When the agent moves straight and fast, the visual encoder is activated more frequently.

One explanation for this behavior is based on the inherent IMU's property that directly measures the angular velocity. Unlike visual feature based estima-

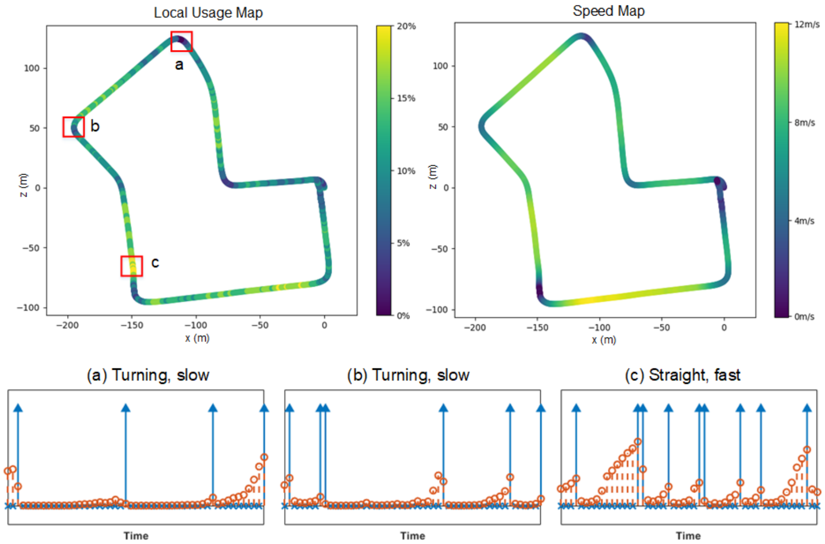

Fig. 5: Visual interpretation of the learned policy on Sequence *07* with $\lambda = 5 \times 10^{-5}$. Top left is the usage map that shows the local usage rate at each time step calculated by averaging the activation rate of the visual encoder during a local window of 31 frames. The agent vehicle speed 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. Seg. a and b show low speed with turning scenarios whereas Seg. c is a high speed straight movement scenario. The policy network tends to activate the visual encoder more frequently when the agent is moving fast in straight, and decrease the usage of the visual encoder when the agent is moving slowly and making a turn.

tion, it is relatively easy to estimate the turning angle using IMU because it is obtained by simple first-order integration. However, estimating translation requires additional process with IMU measurements because it only measures the acceleration which is the second-order differential of translation, requiring a qualified initialization of the velocity. Thus, when the agent is moving fast, IMU-only estimation tends to make large translation errors and hence the policy network enables the visual modality more frequently to reduce the errors.

To provide more insights on the behavior of the policy network, we selected three short segments from the path (marked with red squares in Figure 5 top left) to show the decisions d_t and corresponding probabilities to enable the visual modality (p_t, generated by the policy network) on the bottom of Figure 5. We mark d_t using blue pulses and p_t using orange circles. The policy network exhibits a clear 'integrate-and-fire' pattern where it immediately resets the probability to ≈ 0 after the visual encoder is activated, and it keeps increasing the probability until the visual modality is enabled again. The slope of increasing p_t varies along the path. When the agent is making a sharp turn and moving slowly, p_t tends

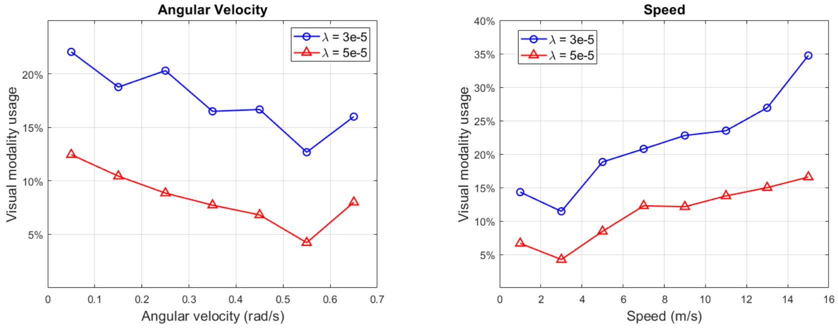

Fig. 6: The average usage rate of the visual modality for different angular velocities (left) and speeds (right) with two different penalty factors λ over the entire test set (Sequence *05, 07, 10*). The learned policy tends to use more images with lower angular velocity and higher speed.

to increase slower and thus the gaps between two visual modality usages are relatively large (segments a and b). When the agent moves fast and straight, p_t surges much faster leading to smaller gaps to enable the visual encoder.

To show the general trend, we also plot the visual modality usage versus angular velocity and speed over all test paths for two different λ's in Figure 6. We calculate the averaged visual encoder usage for the intervals of $[0, 0.1)$, $[0.1, 0.2)$, $\ldots$, $[0.6, 0.7)$ rad$/s$ for the angular velocity and the intervals of $[0, 2)$, $[2, 4)$, $\ldots$, $[14, 16)$ m/s for the speed. It is observed that the usage is closely related to the angular velocity and speed. The usage in general tends to decrease with higher angular velocity and lower speed, although there can be occasional spots where this observation does not necessarily hold since our method is stochastic in nature.

5 Conclusion

In this paper, we propose a novel deep learning-based VIO system that reduces computation overhead and power consumption by opportunistically disabling the visual modality when the visual information is not critical to maintain the accuracy of pose estimation. To learn the selection strategy, we introduce a decision module to the neural VIO structure and end-to-end train it with the Gumbel-Softmax trick. Our experiments show that our approach provides up to 78.8% computation reduction without obvious performance degradation. Our learned strategy significantly outperform simple sub-optimal strategies. Furthermore, the learned policy is interpretable and shows scenario-dependent adaptive behaviours. Our adaptive learning strategy is model-agnostic and can be easily adopted to other deep VIO systems.

Acknowledgements This work was supported in part by Meta Platforms, Inc. We also acknowledge Google LLC for providing GCP computing resources.

References

1. Almalioglu, Y., Turan, M., Sari, A.E., Saputra, M.R.U., de Gusmão, P.P., Markham, A., Trigoni, N.: Selfvio: Self-supervised deep monocular visual-inertial odometry and depth estimation. arXiv preprint arXiv:1911.09968 (2019)
2. Bolukbasi, T., Wang, J., Dekel, O., Saligrama, V.: Adaptive neural networks for efficient inference. In: International Conference on Machine Learning. pp. 527–536. PMLR (2017)
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics **32**(6), 1309–1332 (2016)
4. Campos, V., Jou, B., Giró-i Nieto, X., Torres, J., Chang, S.F.: Skip rnn: Learning to skip state updates in recurrent neural networks. arXiv preprint arXiv:1708.06834 (2017)
5. Chen, C., Rosa, S., Lu, C.X., Trigoni, N., Markham, A.: Selectfusion: A generic framework to selectively learn multisensory fusion. arXiv preprint arXiv:1912.13077 (2019)
6. Chen, C., Rosa, S., Miao, Y., Lu, C.X., Wu, W., Markham, A., Trigoni, N.: Selective sensor fusion for neural visual-inertial odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10542–10551 (2019)
7. Clark, R., Wang, S., Wen, H., Markham, A., Trigoni, N.: Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
8. 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)
9. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE transactions on pattern analysis and machine intelligence **40**(3), 611–625 (2017)
10. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 15–22. IEEE (2014)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
12. Glynn, P.W.: Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM **33**(10), 75–84 (1990)
13. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3828–3838 (2019)
14. Graves, A.: Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983 (2016)
15. Han, L., Lin, Y., Du, G., Lian, S.: Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6906–6913. IEEE (2019)
16. Hansen, C., Hansen, C., Alstrup, S., Simonsen, J.G., Lioma, C.: Neural speed reading with structural-jump-lstm. arXiv preprint arXiv:1904.00761 (2019)

17. Hong, E., Lim, J.: Visual inertial odometry using coupled nonlinear optimization. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6879–6885. IEEE (2017)
18. Hua, W., Zhou, Y., De Sa, C.M., Zhang, Z., Suh, G.E.: Channel gating neural networks. Advances in Neural Information Processing Systems **32** (2019)
19. Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., Weinberger, K.Q.: Multi-scale dense networks for resource efficient image classification. arXiv preprint arXiv:1703.09844 (2017)
20. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
21. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
23. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual–inertial odometry using nonlinear optimization. The International Journal of Robotics Research **34**(3), 314–334 (2015)
24. Li, M., Mourikis, A.I.: High-precision, consistent ekf-based visual-inertial odometry. The International Journal of Robotics Research **32**(6), 690–711 (2013)
25. Liu, L., Li, G., Li, T.H.: Atvio: Attention guided visual-inertial odometry. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4125–4129. IEEE (2021)
26. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1150–1157. Ieee (1999)
27. Meng, Y., Lin, C.C., Panda, R., Sattigeri, P., Karlinsky, L., Oliva, A., Saenko, K., Feris, R.: Ar-net: Adaptive frame resolution for efficient action recognition. In: European Conference on Computer Vision. pp. 86–104. Springer (2020)
28. Meng, Y., Panda, R., Lin, C.C., Sattigeri, P., Karlinsky, L., Saenko, K., Oliva, A., Feris, R.: Adafuse: Adaptive temporal fusion network for efficient action recognition. arXiv preprint arXiv:2102.05775 (2021)
29. Mohamed, S., Rosca, M., Figurnov, M., Mnih, A.: Monte carlo gradient estimation in machine learning. J. Mach. Learn. Res. **21**(132), 1–62 (2020)
30. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
31. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017)
32. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. Ieee (2004)
33. Panda, R., Chen, C.F.R., Fan, Q., Sun, X., Saenko, K., Oliva, A., Feris, R.: Adamml: Adaptive multi-modal learning for efficient video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7576–7585 (2021)
34. Qin, T., Li, P., Shen, S.: Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics **34**(4), 1004–1020 (2018)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

36. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: International conference on machine learning. pp. 1278–1286. PMLR (2014)
37. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011)
38. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE robotics & automation magazine **18**(4), 80–92 (2011)
39. Seo, M., Min, S., Farhadi, A., Hajishirzi, H.: Neural speed reading via skim-rnn. arXiv preprint arXiv:1711.02085 (2017)
40. Shamwell, E.J., Leung, S., Nothwang, W.D.: Vision-aided absolute trajectory estimation using an unsupervised deep network with online error correction. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2524–2531. IEEE (2018)
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
42. Teerapittayanon, S., McDanel, B., Kung, H.T.: Branchynet: Fast inference via early exiting from deep neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 2464–2469. IEEE (2016)
43. Veit, A., Belongie, S.: Convolutional networks with adaptive inference graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–18 (2018)
44. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 2043–2050. IEEE (2017)
45. Wang, X., Yu, F., Dou, Z.Y., Darrell, T., Gonzalez, J.E.: Skipnet: Learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 409–424 (2018)
46. Wang, Y., Chen, Z., Jiang, H., Song, S., Han, Y., Huang, G.: Adaptive focus for efficient video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16249–16258 (2021)
47. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3), 229–256 (1992)
48. Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L.S., Grauman, K., Feris, R.: Blockdrop: Dynamic inference paths in residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8817–8826 (2018)
49. Wu, Z., Xiong, C., Ma, C.Y., Socher, R., Davis, L.S.: Adaframe: Adaptive frame selection for fast video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1278–1287 (2019)
50. Xue, F., Wang, Q., Wang, X., Dong, W., Wang, J., Zha, H.: Guided feature selection for deep visual odometry. In: Asian Conference on Computer Vision. pp. 293–308. Springer (2018)
51. Xue, F., Wang, X., Li, S., Wang, Q., Wang, J., Zha, H.: Beyond tracking: Selecting memory and refining poses for deep visual odometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8575–8583 (2019)
52. Yang, M., Kim, H.S.: Deep joint source-channel coding for wireless image transmission with adaptive rate control. arXiv preprint arXiv:2110.04456 (2021)

53. Yang, N., Wang, R., Gao, X., Cremers, D.: Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. IEEE Robotics and Automation Letters **3**(4), 2878–2885 (2018)
54. Yuan, Z., Wu, B., Sun, G., Liang, Z., Zhao, S., Bi, W.: S2dnas: Transforming static cnn model for dynamic inference via neural architecture search. In: European Conference on Computer Vision. pp. 175–192. Springer (2020)
55. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1851–1858 (2017)
56. Zou, Y., Ji, P., Tran, Q.H., Huang, J.B., Chandraker, M.: Learning monocular visual odometry via self-supervised long-term modeling. In: European Conference on Computer Vision. pp. 710–727. Springer (2020)