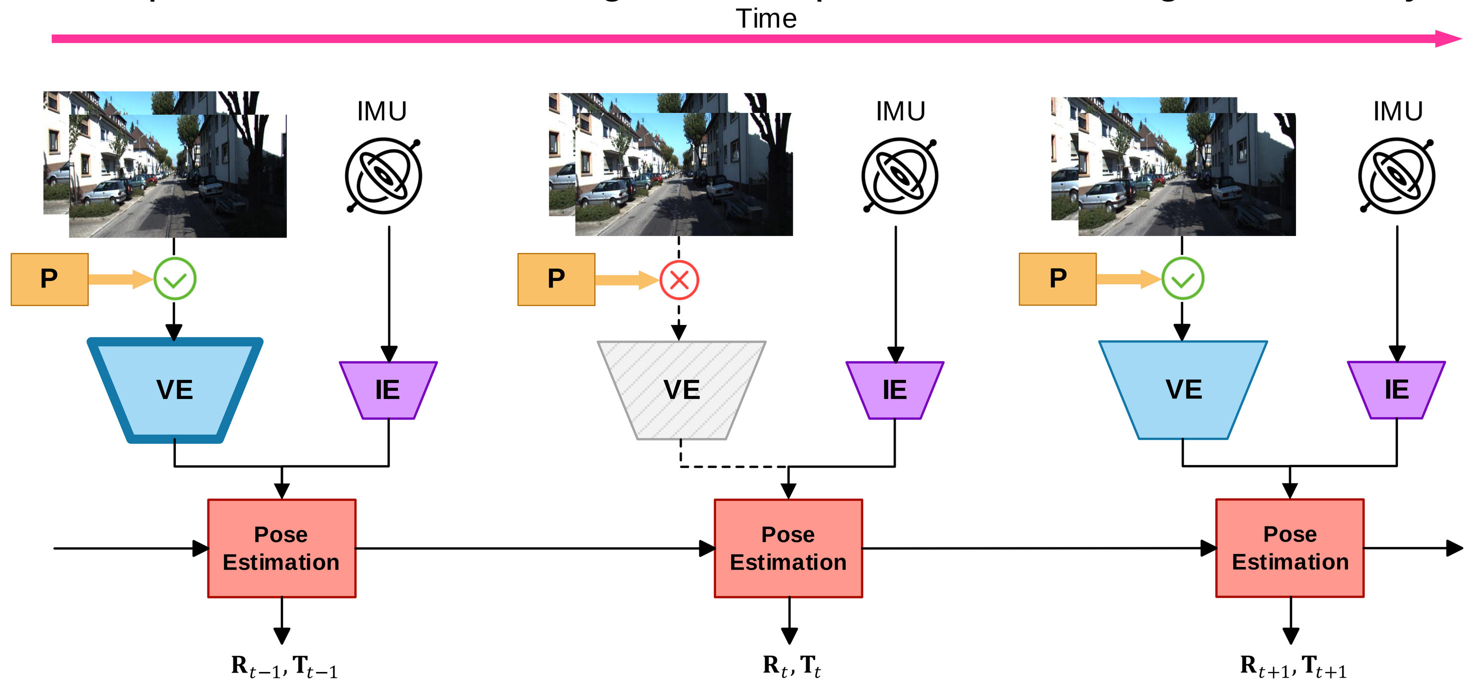
Efficient Deep Visual and Inertial Odometry with Adaptive Visual Modality Selection

Paper ID: 7099

Mingyu Yang, Yu Chen, Hun-Seok Kim
EECS, University of Michigan, Ann Arbor, MI

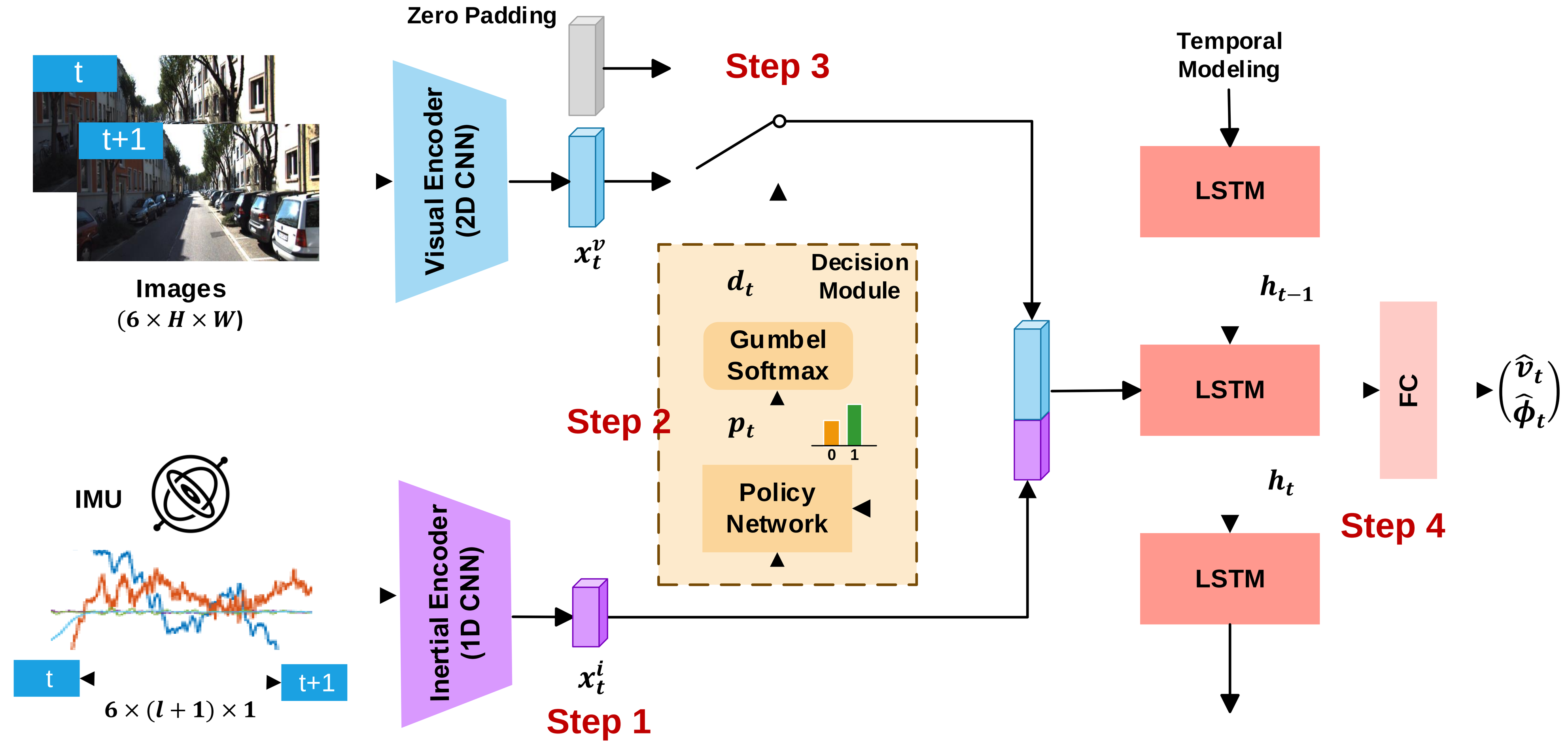
Introduction

- **Visual-inertial odometry (VIO)** estimates the agent's self-motion using information from cameras and inertial measurement unit (IMU)
- **Deep learning-based VIO** has shown competitive performance compared with traditional geometric methods
- Prior works use **both visual and inertial inputs** -- **not affordable** for energy-constraint devices
- We propose to learn a policy to **adaptively disable the visual encoder (VE)** to save computation while remaining a similar performance using full modality



Proposed method with a policy network that dynamically disables the visual encoder (VE), which is much more costly than the inertial encoder (IE)

Deep VIO with Visual Modality Selection



Step 1: At time t, the IMU data between adjacent images is fed to the inertial encoder to extract the inertial feature x_t^i

Step 2: The decision module takes in the inertial feature x_t^i and the hidden state h_{t-1}, and outputs the probability of a Bernoulli dist. p_t, from which decision d_t is sampled. *Gumbel-softmax* is adopted to make the sampling differentiable.

Step 3: If $d_t = 0$, x_t^i is fed to the LSTM along with zeros. Otherwise, images are passed through the visual encoder and generate visual features x_t^v. Then we concat. x_t^v and x_t^i and fed them to the LSTM.

Step 4: The LSTM network produces the pose estimation for time t and the hidden state h_t

Loss Function

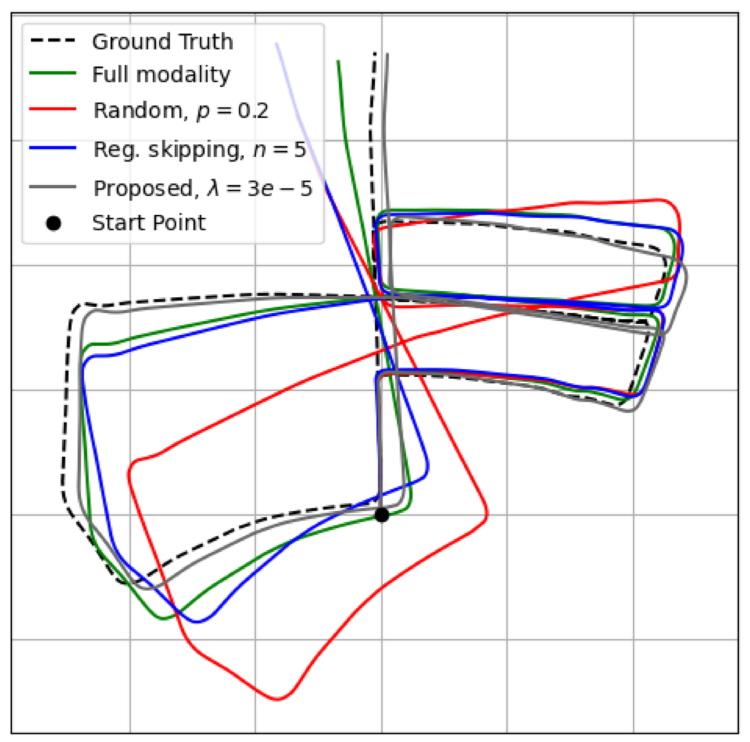
$$L = L_{pose} + L_{penalty} = \frac{1}{3(T-1)}\sum_{t=1}^{T-1}\left(\|v_t - \hat{v}_t\|_2^2 + \alpha\|\phi_t - \hat{\phi}_t\|_2^2\right) + \frac{1}{T-1}\sum_{t=1}^{T-1}\lambda d_t$$

Model Evaluation and Interpretation

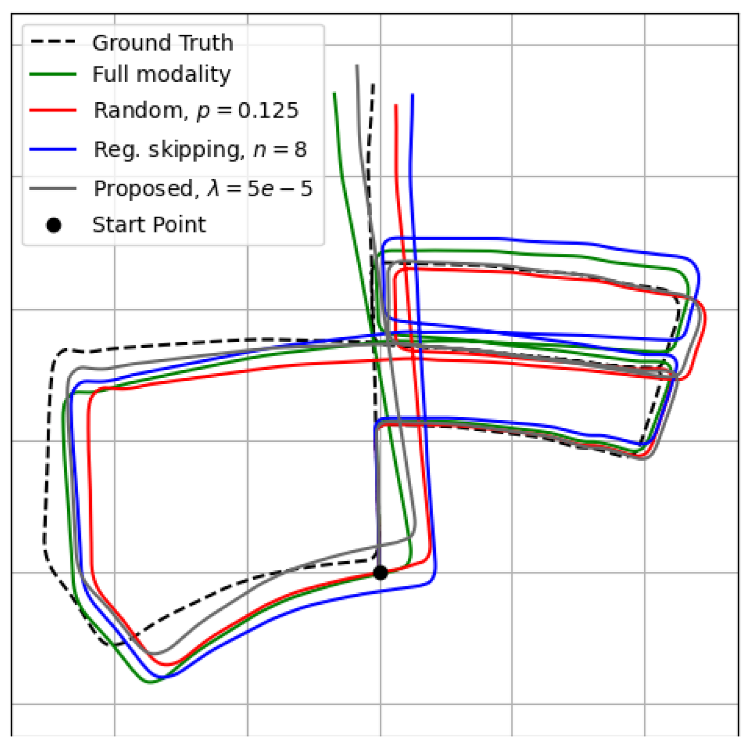
The experiments are conducted on KITTI Odometry dataset. The model is trained on path *00, 01, 02, 04, 06, 08, 09* and tested on path *05, 07, and 10*.

Comparison with Heuristic Sampling Baselines

Baseline #1: random sampling with a probability of p
Baseline #2: regular skipping with n

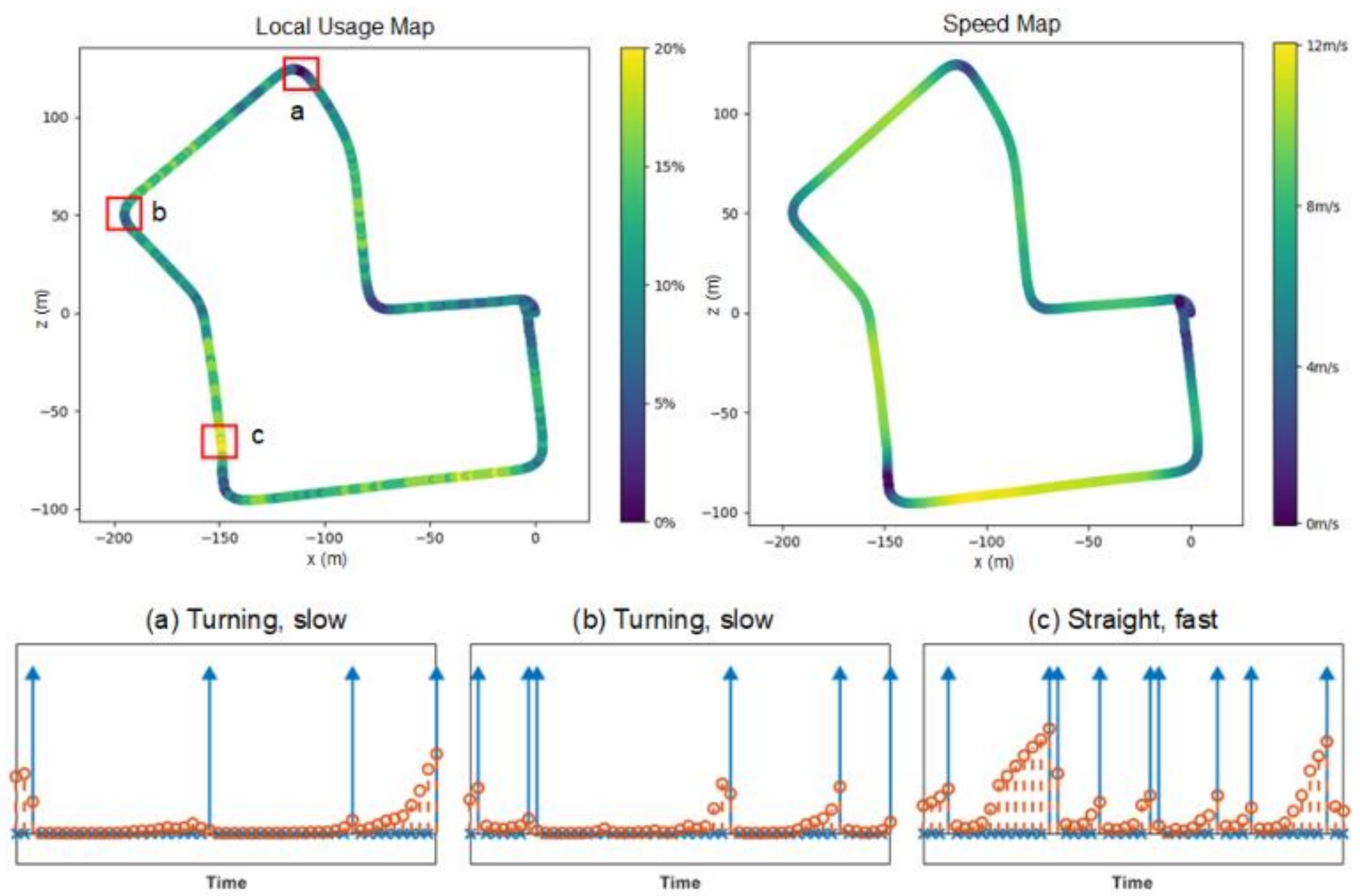


20% image usage



12.5% image usage

Visual Interpretation of Learned Policy



- The decision-making process exhibits an ***Integrate-and-fire*** pattern
- The learned policy shows decreasing visual encoder usage when the vehicle is turning or driving in a low speed

Comparison with SOTA VO/VIO Methods

	Method	Seq.05 $t_{rel}(\%)$	Seq.05 $r_{rel}(^\circ)$	Seq.05 Usage (%)	Seq.07 $t_{rel}(\%)$	Seq.07 $r_{rel}(^\circ)$	Seq.07 Usage (%)	Seq.10 $t_{rel}(\%)$	Seq.10 $r_{rel}(^\circ)$	Seq.10 Usage (%)
Geo	ORB-SLAM2 *	**9.12**	**0.2**	100	10.34	**0.3**	100	**4.04**	**0.3**	100
	VINS-Mono †	11.6	1.26	100	**10.0**	1.72	100	16.5	2.34	100
Self-Sup.	Monodepth2 *	4.66	1.7	100	4.58	2.6	100	7.73	3.4	100
	VIOLearner †	3.00	**1.40**	100	3.60	2.06	100	2.04	1.37	100
	DeepVIO †	**2.86**	2.32	100	**2.71**	**1.66**	100	**0.85**	**1.03**	100
Sup.	GFS-VO *	3.27	1.6	100	3.37	2.2	100	6.32	2.3	100
	BeyondTracking *	2.59	1.2	100	3.07	1.8	100	3.94	1.7	100
	Soft Fusion †	4.44	1.69	100	2.95	1.32	100	3.41	1.41	100
	Hard Fusion †	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

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