Learning Selective Sensor Fusion for State Estimation

Learning Selective Sensor Fusion for States Estimation

Changhao Chen*, Stefano Rosa, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham

***Abstract*—Autonomous vehicles and mobile robotic systems are typically equipped with multiple sensors to provide redundancy. By integrating the observations from different sensors, these mobile agents are able to perceive the environment and estimate system states, e.g. locations and orientations. Although deep learning approaches for multimodal odometry estimation and localization have gained traction, they rarely focus on the issue of robust sensor fusion - a necessary consideration to deal with noisy or incomplete sensor observations in the real world. Moreover, current deep odometry models suffer from a lack of interpretability. To this extent, we propose SelectFusion, an end-to-end selective sensor fusion module which can be applied to useful pairs of sensor modalities such as monocular images and inertial measurements, depth images and LIDAR point clouds. Our model is a uniform framework that is not restricted to specific modality or task. During prediction, the network is able to assess the reliability of the latent features from different sensor modalities and estimate trajectory both at scale and global pose. In particular, we propose two fusion modules - a deterministic soft fusion and a stochastic hard fusion, and offer a comprehensive study of the new strategies compared to trivial direct fusion. We extensively evaluate all fusion strategies in both public datasets and on progressively degraded datasets that present synthetic occlusions, noisy and missing data and time misalignment between sensors, and we investigate the effectiveness of the different fusion strategies in attending the most reliable features, which in itself, provides insights into the operation of the various models.**

***Index Terms*—Sensor Fusion, Localization, Feature Selection, Deep Neural Networks, Multimodal Learning, Visual-Inertial Odometry, Point Cloud Odometry, Robot Navigation**

I. Introduction

Mobile agents are often outfitted with multiple sensors. For example, a self-driving vehicle is equipped with a combination of GPS, IMUs, cameras, and LIDAR. Making such mobile agents fully autonomous and intelligent requires the ability of sensor fusion, a method that can effectively exploit the individual strengths of distinct sensors and coherently estimate the system states. Multimodal sensor fusion has long been a central problem in robotics and computer vision [41], with applications to perception, planning and control. Despite different application scenarios, the rationale for sensor fusion is more or less the same: many system state variables are not always fully observable by a single sensor modality, and combining different sensors that are complementary to each other can reduce the overall uncertainty, improving accuracy and/or robustness. Conventional sensor fusion methods resort to handcrafted design that heavily relies on human experience and domain knowledge. Consequently, the developed fusion methods are often modality-specific and/or task-specific.

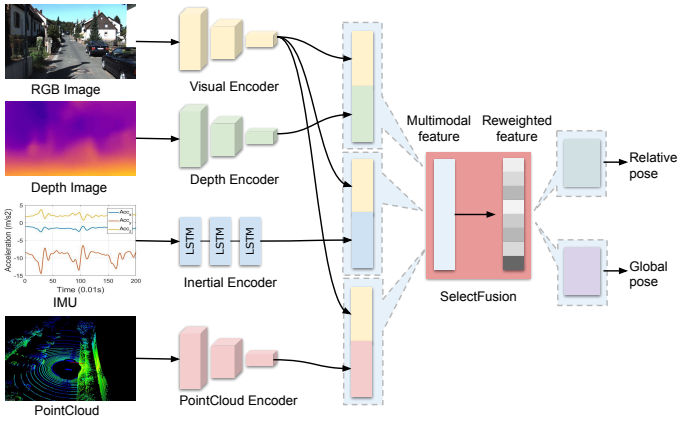


Fig. 1: An overview of the general framework to learn system states from multiple sensor modalities. Our framework can selectively utilize the suitable features for solving problems to improve both the accuracy and robustness.

Recently, there are growing interests in applying deep neural networks (DNNs) for *learning to estimate system states* in an end-to-end manner, for example, solving Visual Odometry (VO) [20], [45], [53], Visual-Inertial Odometry (VIO) [7], [35] or camera relocalization [6], [22]. Instead of building analytical models by hand, they are achieved by learning complex mapping functions directly from raw sensory data to target values. These end-to-end approaches are appealing due to the capability of deep networks to automatically extract features from high-dimensional raw data. However, despite the long history of classical sensor fusion techniques, there is a lack of effective fusion strategies working on the deep feature space, especially for the tasks of localization and odometry estimation. These previous learning-based methods are not explicitly modelling error sources in real-world usage. Without considering possible sensor errors, all features are directly fed into other modules for further pose regression in [1], [6], [22], or simply concatenated as in [7]. These factors can possibly

Changhao Chen is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, 410073, China
Stefano Rosa is with Istituto Italiano di Tecnologia (IIT), Genoa, Italy
Chris Xiaoxuan Lu is with the School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, United Kingdom
Bing Wang, Niki Trigoni and Andrew Markham are with the Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom
*Corresponding author: Changhao Chen (changhao.chen@cs.ox.ac.uk)
This work was supported by EPSRC Program "ACE-OPS: From Autonomy to Cognitive assistance in Emergency OPerationS" (Grant number: EP/S030832/1) and NFSC (Grant number: 62103427, 62073331)
The code of this work is available at https://github.com/changhao-chen/selective_sensor_fusion

cause troubles to the accuracy and safety of neural systems, when input data are corrupted or missing. Moreover, the features from different modalities are considered equally important in these methods, although the complementary property of different modalities requires systems to utilise deep features with regard to observation uncertainties or self/environmental dynamics.

For this reason, we present a generic framework that models feature selection for robust sensor fusion, as illustrated in Figure 1. This work mainly considers the problem of using a pair of sensor modalities, although it can be extended naturally to three or more modalities. As a case study, two tasks - learning global localization and ego-motion estimation, are chosen to demonstrate the effectiveness of our proposed selective sensor fusion. Our system is not restricted to specific modality, performing feature selection from four different sensor data, i.e. RGB-images, inertial measurements, LIDAR point clouds and depth images. The selection process is conditioned on the measurement reliability and the dynamics of both self-motion and environment. Two alternative feature weighting strategies are presented: soft fusion, implemented in a deterministic fashion; and hard fusion, which introduces stochastic noise and intuitively learns to keep the most relevant feature representations, while discarding useless or misleading information.

By explicitly modelling the selection process, we are able to demonstrate the strong correlation between selected features and environmental/measurement dynamics by visualizing the sensor fusion masks, as illustrated in Figure 9. In the case of estimating visual-inertial odometry, our results show that features extracted from different modalities (i.e., vision and inertial motion) are complementary in various conditions: inertial features contribute more in presence of fast rotation, while visual features are preferred during large translations (Figure 11). Thus, the selective sensor fusion provides insights into the underlying strengths of each sensor modality, guiding future multimodal system design. We demonstrate how incorporating selective sensor fusion makes neural models robust to data corruption typically encountered in real-world scenarios.

This paper builds on the work published in [4], and presents a generic framework for selective sensor fusion in multimodal deep pose estimation. This work extends the fusion strategies from visual-inertial odometry to the problem of learning LIDAR-visual odometry and RGB-depth relocalization. To summarise, the novel contributions of this work are as follows:

- We present SelectFusion, a novel generic framework to learn selective sensor fusion enabling more robust and accurate odometry and localization in real-world scenarios.
- We show how our selective sensor fusion can be incorporated into a uniform framework, not restricted by specific modality or task, by learning odometry estimation or relocalization on fusing a pair of modalities from vision, depth, inertial and LIDAR data.
- Our SelectFusion masks can be visualized and interpreted, providing deeper insights into the relative strengths of each stream, and guiding future system design.
- We create challenging datasets on top of current public datasets by considering seven different sources of sensor degradation, and conduct a new and complete study on the accuracy and robustness of deep sensor fusion in presence of corrupted data.

The reminder of the paper is organized as follows: Section II contains a survey of related work; Section III presents a generic framework for multimodal sensor fusion; Section IV introduces our proposed selective sensor fusion mechanism; Section V evaluates SelectFusion applied to three multimodal models for relocalization and trajectory estimation through extensive experiments; Section VI finally draws conclusions.

II. Background and Related Work

A. Learning-based Pose Estimation

Visual-inertial Odometry: Recent work shows how it is possible to learn to estimate odometry from inertial data using recurrent neural networks [3], making deep visual-inertial odometry estimation possible. VINet [7] uses neural network to learn visual-inertial odometry, by directly concatenating visual and inertial features. VIOLearner [35] presents an online error correction module for deep visual-inertial odometry that estimates the trajectory by fusing RGB-D images with inertial data. DeepVIO [15] recently proposes a fusion network to fuse visual and inertial features. This network is trained with a dedicated loss. However, this way of learning sensor fusion does not expose the behaviour of the fusion module, while we propose the use of an interpretable mask, that offers insight into the usefulness of the input at any time. We observe that previous methods do not properly address the problem of learning a meaningful sensor fusion strategy, but simply concatenate visual and inertial features in latent space.

LIDAR Odometry: Learning LIDAR odometry has been explored by LO-Net [26], which exploits geometric consistency for scan-to-scan motion estimation, while also learning pose correction similarly to deep SLAM approaches, and can achieve accuracy comparable to traditional approaches [29]. Fusion of LIDAR and visual information has been investigated in [13], which proposes to fuse LIDAR and visual information, but in their work the learning is limited to training a model for removal of moving objects rather than localization.

Camera Relocalization Deep approaches have also been devoted to visual localization. Posenet [22] is the first work to use Convolutional Neural Networks (CNNs) for 6-DoF pose regression from monocular images. PoseNet has been further improved by combining CNNs and LSTMs for feature correlation [43], introducing temporal information [6], incorporating spatial constraints [1] or by adding additional covisibility constraints based on local maps and the estimated odometry [49]. MS-Transformer [36] is a recent relocalization work based on transformer architecture, achieving the state-of-the-art results.

B. Multimodal Learning

Multimodal learning aims to solve machine learning problems involving multiple data modalities. They are generally

categorized into aggregation-based and alignment-based fusion methods. The success of multimodal learning has been demonstrated in a wide range of applications, e.g. video captioning [39], medical image retrieval [14], face recognition [8], manipulation [23], autonomous navigation [27] and body-sensor-networks [44]. Recently, [46] designs a Channel-Exchanging-Network (CEN) that fuses multiple modalities by dynamically exchanging the channels of sub-networks. [25] proposes a residual fusion network (RFN) based framework that automatically extracts features and fuses multi-scale features. In [38], [40], a shared layer is designed to transfer cross-modal features, in which an inner product function of extracted features from two modalities is adopted to combine them for domain transferring. This inner product method is similar to our soft fusion but for different purposes. However, there is a lack of systematic study into the sensor fusion for deep state estimation, especially in learning based localization and pose estimation, as discussed in Section II-A.

C. Attention Mechanism

Our proposed selective sensor fusion is particularly related to attention mechanisms, that have been widely applied in neural machine translation [10], image caption generation [48], visual question answering [28] and video description [17]. Limited by the fixed-length vector in embedding space, these attention mechanisms compute a focus map to help the decoder, when generating a sequence of words. This is different from our design intention that the features selection works to fuse multimodal sensor fusion for deep pose estimation, and cope with more complex error resources, and self-motion dynamics.

III. Learning Multimodal Representations

This section presents a uniform framework to learn multimodal representation for state estimation, which lays the foundation for selective sensor fusion. Figure 2, 3 and 4, show a modular overview of the architecture, consisting of feature encoders, feature fusion, temporal modelling and task solver.

A. Feature Encoders

1) Visual Feature Encoders: As visual feature encoders are used in both global relocalization and odometry estimation, they are designed with respect to the property of each task for better feature extraction and utilization.

For a relative pose (odometry) estimation, latent representations are extracted from a set of two consecutive monocular images $\mathbf{x}_V$. Ideally, we want our visual encoder f_{vision} to learn geometrically meaningful features rather than features overfitted with appearance or context. For this reason, instead of using a PoseNet model [22], as commonly found in other DL-based VO approaches [50], [51], [53], we use a FlowNet-style architecture, i.e. FlowNetSimple [9] as our feature encoder. Flownet provides features that are suited for optical flow prediction, which highly contributes to the ego-motion detection. The network consists of nine convolutional layers. The size of the receptive fields gradually reduces from 7×7 to 5×5 and finally 3×3, with stride two for the first six layers. Each layer is followed by a ReLU nonlinearity except for the last one, and we use the features from the last convolutional layer $\mathbf{a}_V$ as our visual feature. We initialize the visual encoder with the weights of a model that was pre-trained on the FlyingChairs dataset[1], since training from scratch would require a larger amount of data compared with our dataset size.

For a global relocalization task, we instead use Residual Neural Network (ResNet) [16] to extract features from a set of single images. Both structure and appearance features contribute to the retrieval of absolute poses in the 3D scene that has been visited before. Hence, visual features should capture the entire scene. We adopt ResNet18, consisting of 18 layers convolutional layers with skip connections, and modify it by introducing an average pooling layer and a full-connected layer at the end, that transform the features after ResNet18 to a d dimension visual feature $\mathbf{a}_V$.

In summary, given a set of images $\mathbf{x}_V$, we are able to extract visual features $\mathbf{a}_V \in \mathbb{R}^d$ suited to the task via the Visual Encoder (FlowNet) or (ResNet) f_{vision}:

$$\mathbf{a}_V = f_{\text{vision}}(\mathbf{x}_V). \tag{1}$$

2) Inertial Feature Encoder: Inertial data streams have a strong temporal component, and are generally available at higher frequency (∼100 Hz) than images (∼10 Hz). In order to model the temporal dependencies of consecutive inertial measurements, we use a two-layer Bi-directional LSTM with 128 hidden states as the Inertial Feature Encoder f_{inertial}. In the deep VIO model, as shown in Figure 4, a window of inertial measurements $\mathbf{x}_I$ between each two images is fed to the inertial feature encoder in order to extract the d dimensional feature vector $\mathbf{a}_I \in \mathbb{R}^d$:

$$\mathbf{a}_I = f_{\text{inertial}}(\mathbf{x}_I). \tag{2}$$

3) Depth Feature Encoder: In our work, depth image is exploited to solve the task of vision-depth based relocalization, as shown in Figure 2. Similar to the visual encoder designed for relocalization, we also use ResNet18 as depth feature encoder, but replace the first layer of ResNet model with a 1-channel convolutional network, considering that depth image is 1-channel rather than 3-channels. Hence, the input is a set of 1-channel depth images $\mathbf{x}_D$, and transformed into a d dimensional features vector $\mathbf{a}_D \in \mathbb{R}^d$ via the depth encoder f_{depth}:

$$\mathbf{a}_D = f_{\text{depth}}(\mathbf{x}_D). \tag{3}$$

4) Pointcloud Feature Encoder: The point clouds are a set of data in Cartesian coordinates, representing 3D structure in space. They are produced normally by LIDAR devices. The sparse structure and irregular format of point cloud data make them hard to be processed directly by neural networks. To allow convolutional neural networks (CNNs) to effectively process point cloud data, we convert them into a regular point cloud matrix via the cylindrical projection [5], [26]:

$$\alpha = \arctan(y/x)/\Delta\alpha \tag{4}$$

[1]https://lmb.informatik.uni-freiburg.de/resources/datasets/FlyingChairs.en.html

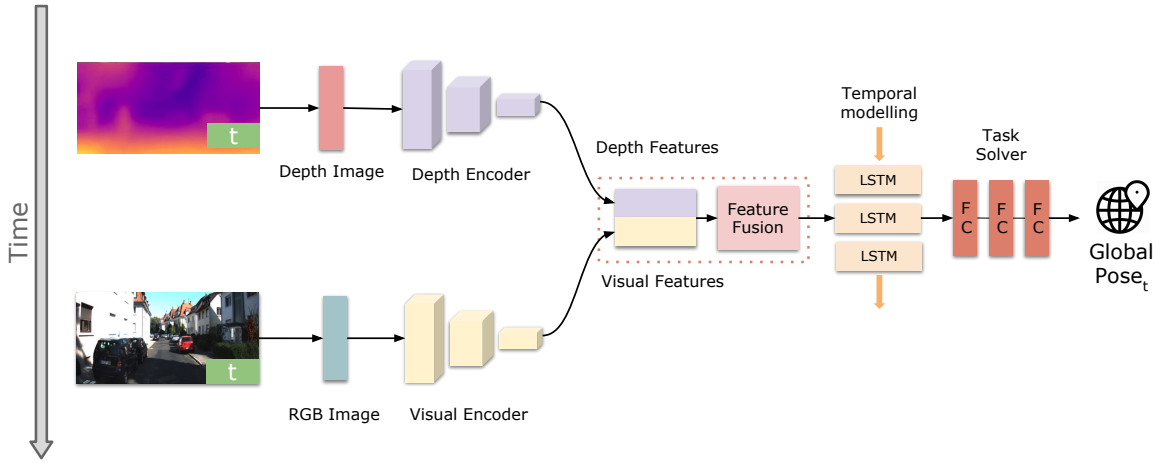


Fig. 2: An overview of our depth-vision relocalization (**Task 1**) architecture with proposed selective sensor fusion, consisting of depth and visual encoders, feature fusion, temporal modelling and task solver (global pose estimation).

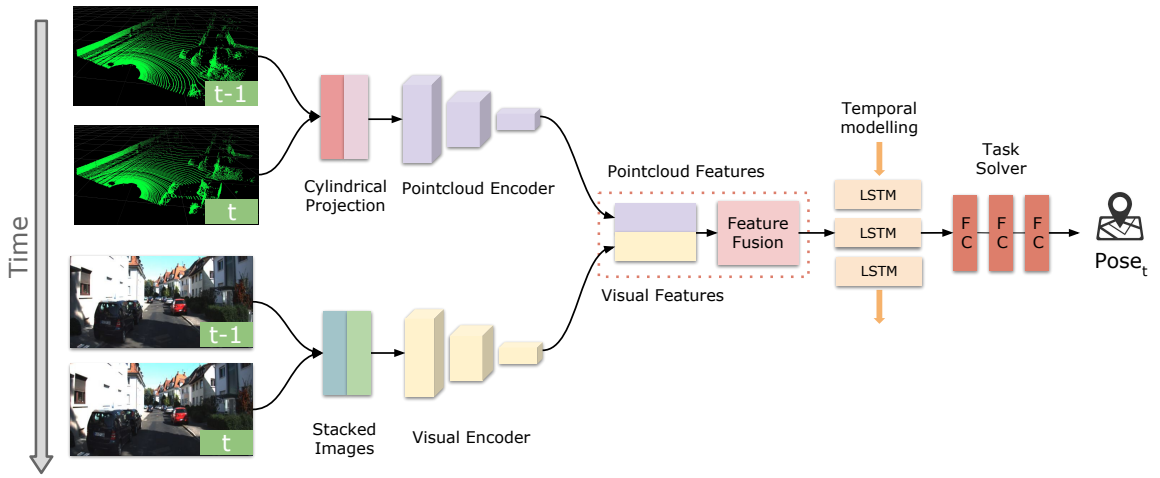


Fig. 3: An overview of our neural LIDAR-visual odometry (**Task 2**) architecture with proposed selective sensor fusion, consisting of visual and LIDAR encoders, feature fusion, temporal modelling and task solver (relative pose regression).

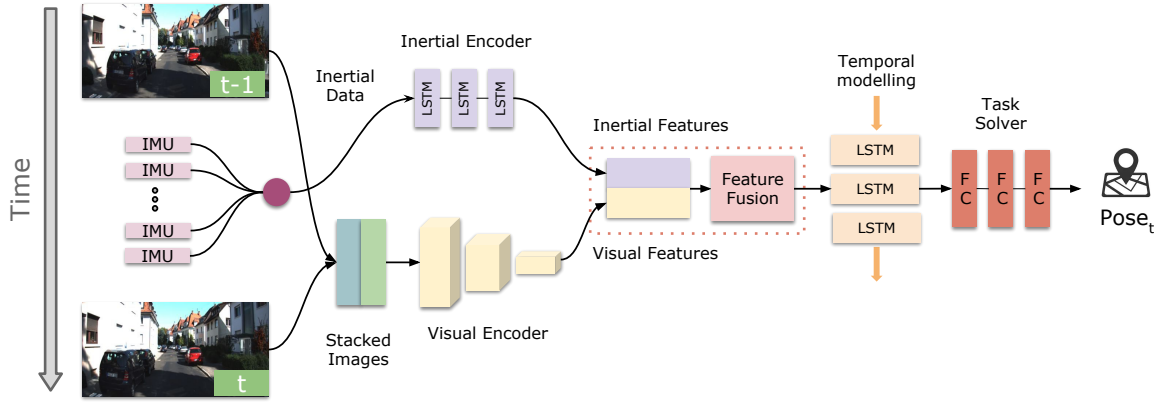


Fig. 4: An overview of our neural visual-inertial odometry (**Task 3**) architecture with proposed selective sensor fusion, consisting of visual and inertial encoders, feature fusion, temporal modelling and task solver (relative pose regression).

$$\beta = \arcsin(z/\sqrt{x^2+y^2+z^2})/\Delta\beta \tag{5}$$

where (x, y, z) are original coordinates in LIDAR coordinate system, and (α, β) are new coordinates in the point cloud matrix. The new point cloud matrix is with a size of $H \times W \times C$. The position (α, β) of matrix is filled with the range value $r = \sqrt{x^2+y^2+z^2}$ from the position (x, y, z) of original point cloud.

In this work, point cloud data are used to learn vision-LIDAR odometry, as shown in Figure 3 and hence we also use the FlowNet visual encoder to transform the input matrix $\mathbf{x}_P$ into a d dimensional point cloud feature $\mathbf{a}_P \in \mathbb{R}^d$:

$$\mathbf{a}_P = f_{\text{pointcloud}}(\mathbf{x}_P). \tag{6}$$

B. Fusion Function

We now combine the high-level representation produced by each feature encoder from raw data sequences, with a fusion function g that combines information from a pair of sensor modalities to extract the useful combined feature $\mathbf{z}$ for a regression task:

$$\mathbf{z} = g(\mathbf{a}_1, \mathbf{a}_2), \tag{7}$$

where $(\mathbf{a}_1, \mathbf{a}_2)$ is any pair of sensor modality features from visual $\mathbf{a}_V$, inertial $\mathbf{a}_I$, depth $\mathbf{a}_D$, and point cloud $\mathbf{a}_P$ channels. In this work, we specifically investigate the problem of fusing two sensor modalities for better demonstration on existing datasets, although our framework can extend naturally to exploit three or more modalities.

There are several different ways to implement this fusion function. The current approach is to directly concatenate the two features together into one feature space (we call this method direct fusion g_{direct}). However, in order to learn a robust sensor fusion model, we propose two fusion schemes – deterministic soft fusion g_{soft} and stochastic hard fusion g_{hard}, which explicitly model the feature selection process according to the current environment dynamics and the reliability of the data input. The fusion network is another deep neural network module. Details will be discussed in Section IV.

C. Temporal Modelling and Task Solvers

The fundamental tenet of state estimation requires modelling temporal dependencies to derive accurate system states, e.g. relative poses. In the past, a state-space-model (SSM) describes this temporal relation and evolution of system states. Similarly, in our learning model, a recurrent neural network, i.e. Long Short-Term Memory (LSTM) network takes in the input combined feature representation $\mathbf{z}_t$ at time step t and its previous hidden states $\mathbf{h}_{t-1}$ and models the dynamics and connections between a sequence of features. The hidden states $\mathbf{h}_t$ contain the history of the features relevant to the task. After the recurrent network, a fully-connected layer serves as the regressor, transforming the features to a system state $\mathbf{y}_t$, i.e. pose transformation or global pose, representing the motion transformation over a time window or a global location/orientation.

Hence, the relation between the final system states $\mathbf{y}_t$ and the input features $\mathbf{z}_t$ can be described via a recurrent neural network and previous hidden states $\mathbf{h}_{t-1}$:

$$\mathbf{y}_t = \mathbf{RNN}(\mathbf{z}_t, \mathbf{h}_{t-1}). \tag{8}$$

We implemented three tasks based on this multimodal representation learning framework to estimate key system states from pairs of raw sensory data.

1) Task 1: Learning Vision-Depth Relocalization: The first task is to exploit monocular RGB images and depth images to perform global relocalization in the scenarios that have been visited before. As illustrated in Figure 2, depth and RGB images are encoded into features by the Depth Encoder and Visual Encoder (ResNet), fused as new features through Feature Fusion modules, and converted into global poses via temporal modelling and task regression modules. The global pose $\mathbf{y} = [\mathbf{p}, \mathbf{q}]$ is composed by a 3-D position vector $\mathbf{p} \in \mathbb{R}^3$ and a quaternion $\mathbf{q} \in \mathbb{R}^4$ for orientation. The objective is to minimize the L1 distance between the groundtruth values $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$ and predicted values $[\mathbf{p}, \mathbf{q}]$ with the loss function:

$$L(\theta)_1 = |\hat{\mathbf{p}} - \mathbf{p}| + \lambda_1 \left| \hat{\mathbf{q}} - \frac{\mathbf{q}}{||\mathbf{q}||} \right|, \tag{9}$$

where λ_1 is a balance factor, which we choose as $\lambda_1 = 10$ in our experiment. Here, L1 loss is chosen rather than L2 loss, because L1 loss performs better and more stable [21].

2) Task 2: Learning Lidar-Vision Odometry: The second task is to learn LIDAR-vision odometry. Different from global relocalization, odometry estimation produces relative poses between two frames of images, which can adapt to new scenarios. Global pose is achieved by integrating pose transformations. As shown in Figure 3, the framework consists of Point Cloud Encoder and Visual Encoder (FlowNet) that extract features from LIDAR point cloud data and RGB images, Feature Fusion that combines LIDAR and visual features as a new feature vector, and Temporal Modelling and Task Solver modules to transform features as system states. The network outputs relative poses $\mathbf{y} = [\mathbf{p}, \mathbf{r}]$, consisting of a 3-dimensional translation vector $\mathbf{p} \in \mathbb{R}^3$, and a 3-dimensional Euler rotation vector $\mathbf{r} \in \mathbb{R}^3$. The objective is to minimize the Mean Square Error (MSE) of the relative poses to recover optimal neural networks parameters θ:

$$L(\theta)_2 = ||\hat{\mathbf{p}} - \mathbf{p}||_2 + \lambda_2 ||\hat{\mathbf{r}} - \mathbf{r}||_2, \tag{10}$$

where $[\hat{\mathbf{p}}, \hat{\mathbf{r}}]$ are groundtruth values, and λ_2 is a scale factor to balance between translational error and rotational error. λ_2 is chosen as 100 in our experiment.

3) Task 3: Learning Visual-Inertial Odometry: The third task is to learn visual-inertial odometry, providing accurate pose estimation by using visual and inertial sensors, which are widely deployed in mobile robotics, self-driving vehicles and drones. Similar to LIDAR-vision odometry, our model outputs the relative poses between two frames of images. Figure 4 shows that visual and inertial features are extracted from consecutive monocular images, and a sequence of inertial data between two frames of images by FlowNet based Visual Encoder and LSTM based Inertial Encoder. The features are combined as new features via Feature Fusion, and converted

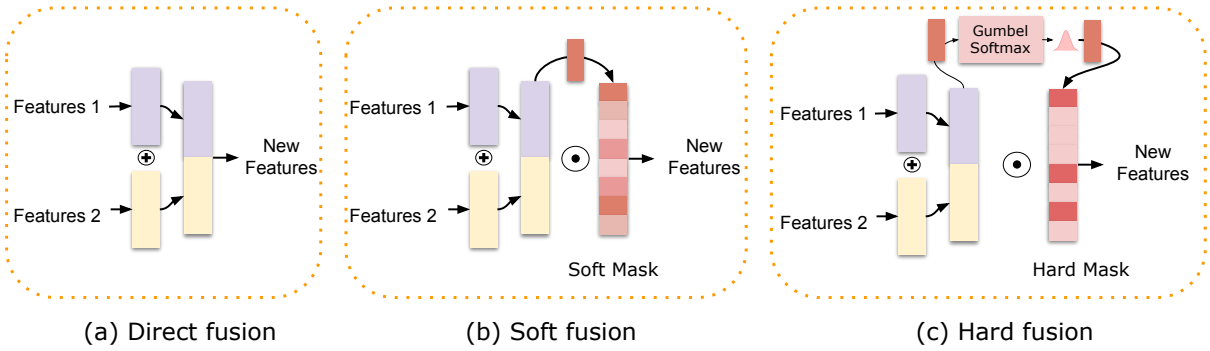


Fig. 5: An overview of three fusion methods: (a) direct fusion, (b) soft fusion and (c) hard fusion.

into system states through Temporal Modelling and Task Regressor. The network produces pose transformation $\mathbf{y} = [\mathbf{p}, \mathbf{r}]$ with a 3-dimensional translation vector $\mathbf{p} \in \mathbb{R}^3$, and a 3-dimensional rotation vector $\mathbf{r} \in \mathbb{R}^3$ (the rotation vector is represented by 3-dimensional Euler angles). By minimizing the MSE of the predicted relative poses, the optimal parameters θ are recovered via:

$$L(\theta)_3 = ||\hat{\mathbf{p}} - \mathbf{p}||_2 + \lambda_3 ||\hat{\mathbf{r}} - \hat{\mathbf{r}}||_2, \tag{11}$$

where $[\hat{\mathbf{p}}, \hat{\mathbf{r}}]$ are true relative poses, $[\mathbf{p}, \mathbf{r}]$ are predicted values, and λ_3 is a scale factor to balance between translational error and rotational error. In our case, we choose λ_3 as 100.

IV. Selective Sensor Fusion

In this section, we propose SelectFusion, a generic framework to selectively learn multisensory representation from raw data. Intuitively, the features from each modality offer different strengths for the task of state estimation. Our perspective is that simply considering all features as that they are equally important and correct, without any consideration of degradation and self/environmental dynamics, is unwise and will lead to unrecoverable drifts and errors. Therefore, we propose two different selective sensor fusion schemes for explicitly learning the feature selection process: soft (deterministic) fusion, and hard (stochastic) fusion, as illustrated in Figure 6. In addition, we also present a straightforward sensor fusion scheme – direct fusion – as a baseline model for comparison.

A. Direct Fusion

A straightforward approach for implementing sensor fusion consists in the use of Multi-Layer Perceptrons (MLPs) to combine the features from the two sensor modality channels. Ideally, the system learns to discriminate relevant features for prediction in an end-to-end fashion. Hence, direct fusion is modelled as:

$$g_{\text{direct}}(\mathbf{a}_1, \mathbf{a}_2) = [\mathbf{a}_1; \mathbf{a}_2] \tag{12}$$

where $[\mathbf{a}_1; \mathbf{a}_2]$ denotes an operation function that concatenates features $\mathbf{a}_1$ and $\mathbf{a}_2$, which are extracted from the Modality One and Two respectively.

B. Soft Fusion (Deterministic)

We now propose a soft fusion scheme that explicitly and deterministically models feature selection. Similar to the widely applied attention mechanism [17], [42], [48], this function re-weights each feature by conditioning on both sensor modality channels, allowing the feature selection process to be jointly trained with other modules. The function is deterministic and differentiable.

Here, a pair of continuous masks $\mathbf{s_1}$ and $\mathbf{s_2}$ are introduced to implement soft selection of the extracted feature representations, before these features are passed to temporal modelling and task solver:

$$\mathbf{s}_1 = \text{Sigmoid}(\text{MLP}_1([\mathbf{a}_1; \mathbf{a}_2])) \tag{13}$$

$$\mathbf{s}_2 = \text{Sigmoid}(\text{MLP}_2([\mathbf{a}_1; \mathbf{a}_2])) \tag{14}$$

where $[\mathbf{a}_1; \mathbf{a}_2]$ denotes an operation function that concatenates features $\mathbf{a}_1$ and $\mathbf{a}_2$. MLP is multilayer perceptron, a feedforward neural network that transforms features to fusion mask space. The Sigmoid function makes sure that each of the features will be re-weighted in the range $[0, 1]$. This process is deterministically parameterised by the neural networks, conditioned on both the features $\mathbf{a}_1$ and features $\mathbf{a}_2$. $\mathbf{s_1}$ and $\mathbf{s_2}$ represent soft masks applied to the features extracted from Modality One and Modality Two respectively.

Then, the visual and inertial features are element-wise multiplied with their corresponding soft masks as the new re-weighted vectors. The selective soft fusion function is modelled as

$$g_{\text{soft}}(\mathbf{a}_1, \mathbf{a}_2) = [\mathbf{a}_1 \odot \mathbf{s}_1; \mathbf{a}_2 \odot \mathbf{s}_2]. \tag{15}$$

C. Hard Fusion (Stochastic)

In addition to the soft fusion introduced above, we propose a variant of the fusion scheme – hard fusion. Instead of re-weighting each feature with a continuous value, hard fusion learns a stochastic function that generates a binary mask that either propagates the feature or blocks it. This mechanism can be viewed as a switcher for each component of the feature map, which is a stochastic layer implemented by a parametrised Bernoulli distributions.

However, the stochastic layer cannot be trained directly by back-propagation, as gradients will not propagate through

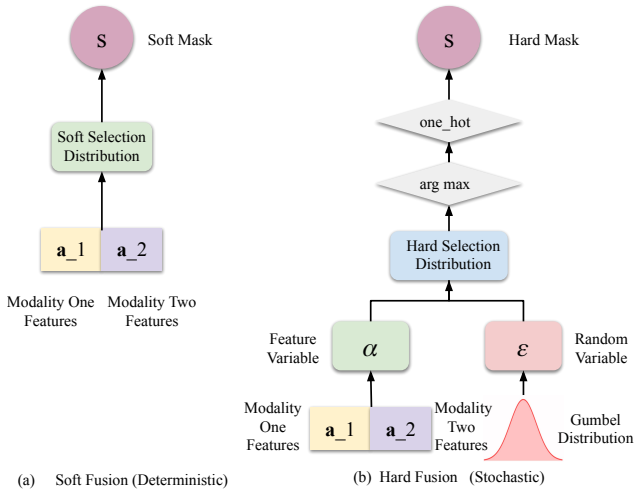


Fig. 6: An illustration of our proposed soft (deterministic) and hard (stochastic) feature selection process.

discrete latent variables. To tackle this, the REINFORCE algorithm [32], [47] is generally used to construct the gradient estimator. In our case, we propose to employ a more lightweight method – Gumbel-Softmax resampling [19], [30] to infer the stochastic layer of hard fusion, so that our hard fusion module can be trained in an end-to-end fashion as well.

Before training the model, the distribution of hard mask is unknown. As each element of this mask $\mathbf{s}^{(i)}$ is a single stochastic variable, we assume it to be under Bernoulli distribution, which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$. As the entire mask $\mathbf{s}$ consists of n elements, it is under the binomial distribution with parameters n, which is the discrete probability distribution of the number of successes in a sequence of n independent experiments. Therefore, instead of learning masks deterministically from features, hard masks $\mathbf{s}_1$ and $\mathbf{s}_2$, representing the binary mask for the features from two modalities, are re-sampled from a discrete Binomial distribution. This discrete distribution is parameterized by $\boldsymbol{\alpha}$, which is learned by deep neural networks and conditioned on features but with the addition of stochastic noise:

$$\mathbf{s}_1 \sim p(\mathbf{s}_1|\mathbf{a}_1, \mathbf{a}_2) = \text{Binomial}(\boldsymbol{\alpha}) \tag{16}$$

$$\mathbf{s}_2 \sim p(\mathbf{s}_2|\mathbf{a}_1, \mathbf{a}_2) = \text{Binomial}(\boldsymbol{\alpha}), \tag{17}$$

where each mask $\mathbf{s} = [\mathbf{s}^{(1)}, ..., \mathbf{s}^{(n)}]$ is a n-dimensional binary vector $\mathbf{s}^{(i)}$. Each element of hard mask $\mathbf{s}^{(i)}$ is a 2-dimensional categorical variable, deciding whether to select the ith feature or not. The total number of features is n. The element $\mathbf{s}^{(i)}$ can be viewed as resampling from a Bernoulli distribution:

$$\mathbf{s}^{(i)} \sim \text{Bernoulli}(\boldsymbol{\alpha}^{(i)}). \tag{18}$$

Similar to soft fusion, the features from two modalities are element-wise multiplied with their corresponding hard masks as new reweighted vectors. The stochastic hard fusion function is modelled as

$$g_{\text{hard}}(\mathbf{a}_1, \mathbf{a}_2) = [\mathbf{a}_1 \odot \mathbf{s}_1; \mathbf{a}_2 \odot \mathbf{s}_2]. \tag{19}$$

Now we come to solve the problem of inferring this discrete distribution in order to generate hard mask $\mathbf{s}$. We apply the so-called Gumbel-Softmax trick to convert the non-continuous function into a continuous approximation by using the fact that the distribution of a discrete random variable $P(x = k)$ can be reparameterized by a random variable π_k and a Gumbel random variable ϵ_k via

$$x = \arg\max_k(\log \pi_k + \epsilon_k). \tag{20}$$

In practical, it is simple to implement this reparameterization trick into our model. Figure 6 (b) shows the detailed workflow of our proposed Gumbel-Softmax resampling based hard fusion. The Gumbel-max trick [31] allows us to efficiently draw a hard mask $\mathbf{s}^{(\mathbf{i})}$ from a categorical distribution given the class vector $\pi_k^{(i)}$ and a Gumbel random variable $\epsilon_k^{(i)}$, and then an one-hot encoding performs 'binarization' of the category:

$$\mathbf{s}^{(i)} = \text{one\_hot}(\arg\max_k[\epsilon_k^{(i)} + \log \pi_k^{(i)}]), \tag{21}$$

where $i \in [1, .., n]$ is the index of feature, $k \in [1, 2]$ is the index of class vector for each feature. In this case, there are only two categories, indicating whether to select a particular feature or not. This can be viewed as a process of adding independent Gumbel perturbations to the discrete class variable. In practice, the random variable $\boldsymbol{\epsilon}$ is sampled from a Gumbel distribution, which is a continuous distribution on the simplex that can approximate categorical samples:

$$\epsilon = -\log(-\log(u)), u \sim \text{Uniform}(0, 1). \tag{22}$$

In Equation 21 the argmax operation is not differentiable, so softmax function is used as an approximation:

$$h^{(i)} = \frac{\exp((\log(\pi_k^{(i)} + \epsilon_k^{(i)})/\tau)}{\sum_{j=1}^{2} \exp((\log(\pi_k^{(i)}) + \epsilon_k^{(i)})/\tau)}, k = 1, 2 \tag{23}$$

where $\tau > 0$ is the temperature that modulates the re-sampling process. Finally, $h^{(i)}$ is transformed into a binary mask $\mathbf{s}^{(i)}$ through the one_hot function.

The $\pi_k^{(i)}$ is jointly learned by deep neural networks in our models, and formulated as the parameters $\boldsymbol{\alpha} = (\pi_k^i)|_{k=1,2}^{i=1..n}$, conditioned on the concatenated feature vectors $[\mathbf{a_1}; \mathbf{a_2}]$ from two modalities:

$$\boldsymbol{\alpha} = \text{ReLU}(\text{FC}([\mathbf{a_1}; \mathbf{a_2}])), \tag{24}$$

where FC is full-connected layer, to map concatenated features into $k * 2$ dimensional class vectors. ReLU is to impose nonlinearity and ensures the class vectors to be nonnegative.

In our approach, we find that modulating the temperature with respect to the training procedure can enable better performance in selective sensor fusion. This is because the temperature determines the samples and gradients: when the temperature is high, the variance of the gradients is small, while the samples are more smooth; at low temperatures, the variance of the gradients is high, while the samples are more discrete, which means it will fit well into the discrete distribution of the fusion mask. Thus we start the temperature from a higher value, i.e. 1 in our case, and gradually decrease it towards 0.5 over each epoch of the training process.

V. EXPERIMENTS

We conducted extensive experiments above four well-known public datasets to learn from different pairs of sensor modalities: the 7-Scenes dataset [37] for vision-depth based relocalization (Task 1), the KITTI odometry dataset [11] for vision-LIDAR odometry estimation (Task 2), the KITTI raw dataset [11] and the EuRoC MAV dataset [2] for visual-inertial odometry (Task 3).

A. Experimental Setups

Our frameworks were implemented with PyTorch and trained on a NVIDIA Titan X GPU. As the main focus of this work is a study of the general multimodal fusion problem, we want to investigate the performance of the two SelectFusion strategies compared to pre-existing models. In each task, we always choose a deep vision-only model and a deep multimodal model with direct fusion as the *common baselines*. Additionally, specific representative works were chosen as *task baselines*, according to each specific task. All of our networks including common baselines were trained with a batch size of 16 using the Adam optimizer, with a learning rate $\text{lr} = 1e^{-4}$, for a fair comparison. All model were trained for 100 epochs, and the sequence length is chosen as 5.

1) Common Baselines: Common baselines share the same basic framework as our proposed SelectFusion framework. For a fair comparison, the hyper-parameters of proposed network and common baselines are identical, including batch size, learning rate, and the dimension of hidden states. The vision-only model is composed of the same visual encoder, temporal modelling and task solver modules as our framework. The multimodal model with direct fusion uses the same structure as our proposed framework, except for the fusion component, which is a simple concatenation of the multimodal features. The single modality model and multimodal model with direct fusion can be viewed as ablated variants of our proposed approach. In addition, we also compare with a recent multimodal fusion work, i.e. Residual Fusion Network (RFN) [25], which is based on the nest connection incorporated into a residual neural network. In the task of depth-vision relocalization and LIDAR-vision odometry, RFN is employed into a framework with the same feature extractors as our select fusion for a fair comparison, but fusion module is replaced with the residual network that aggregates and fuses the features from each extractor. RFN is not employed in visual-inertial odometry, as inertial data are processed with an LSTM, which is not suitable to this CNN based RFN.

2) Vision-Depth Relocalization: **7-Scenes Dataset (vision+depth):** The 7-Scenes dataset [37] contains RGB images and depth data captured by a handheld Microsoft Kinect camera from seven indoor scenarios. Each scene provides several sequences, and each sequence is with 500-1000 frames of colour and depth images. We follow the official data split to train and test our models above this dataset.

Task Baselines: Our SelectFusion model is built as an end-to-end relocalization model, and thus we compare with LSTM-Pose [43], VidLoc [6], and MS-Transformer [36] which are representative within this category of learning techniques.

3) LIDAR-Vision Odometry: **KITTI Odometry Dataset (vision+LIDAR)** The KITTI Odometry dataset [11] provides 11 sequences (00-10) with visual images, LIDAR point cloud and groundtruth. It has been extensively adopted as VO/SLAM benchmark. We use this dataset to fuse the visual and point cloud data to estimate relative pose (odometry) and reconstruct trajectory. Sequences *00, 01, 02, 03, 04, 06, 08, 09* are used for training DNN models, while the rest Sequences *05*, *07*, and *10* are relatively long and used for evaluation. All images are resized to 512×256.

Task Baselines: We use three representative works that are evaluated and widely adopted on the KITTI odometry benchmark, as our task baselines, i.e. VISO2_M [12], ORB-SLAM [33], and ELO [52]. VISO2_M is a monocular VO algorithm, in which a fixed camera height, (i.e. a predefined 1.7 meters in the KITTI dataset) is given to recover the absolute scale of generated trajectories. We also adopt ELO [52], a recent LIDAR odometry work.

4) Visual-Inertial Odometry: **KITTI RAW dataset (visual+inertial)** The KITTI Raw dataset [11] contains the raw data collection from car-driving scenarios. High-frequency inertial data (100 Hz) are only available in the raw unsynchronized data package. We manually synchronize inertial data and images according to their timestamps, in order to exploit the visual and inertial data to learn odometry estimation. We use Sequences *00, 01, 02, 04, 06, 08, 09* for training and tested the network on Sequences *05*, *07*, and *10*, excluding sequence 03 as the corresponding raw file is unavailable. The images and ground-truth provided by GPS are collected at 10 Hz, while the IMU data are at 100 Hz.

EuRoC MAV dataset (visual+inertial) The EuRoC dataset [2] contains tightly synchronized video streams from a Micro Aerial Vehicle (MAV), carrying a stereo camera and an IMU, and is composed by 11 flight trajectories in two environments, exhibiting complex motion. We used Sequence *MH_05_difficult* for testing, and left the other sequences for training. We downsample the images and IMUs to 10 Hz and 100 Hz respectively.

Task Baselines: We choose four representative VIO pipelines, i.e. MSCKF [18], OKVIS [24], mono-VINS [34] and VIO-Learner [35] as task baselines to compare with our deep VIOs: MSCKF [18] is an Extended Kalman Filter based solution; OKVIS [24] is a keyframe based VIO with sliding window nonlinear optimization; mono-VINS [34] uses sliding window nonlinear optimization and IMU preintegration. VIOLearner [35] is a learning based VIO with online error correction.

B. Runtime and Parameters Sensitivity

This section analyzes the runtime and parameters sensitivity of our proposed fusion mechanisms.

We first test direct fusion, soft fusion and hard fusion in the task of learning visual-inertial odometry, to collect their prediction time on a NVIDIA RTX 3080Ti GPU and an Intel Xeon 2.4GHz CPU. Fig 7 reports the averaged results over the per-frame testing time on the Sequence 5 of the KITTI dataset. It is clear to see that no matter our soft fusion or hard fusion only increases the computation burden slightly, compared with

TABLE I: Vision-depth relocalization (Task 1) on the 7-Scenes dataset, reported in position error (m) and orientation error (°)

Scene	LSTM-Pose	VidLoc(V+D)	MS-Transformer	RFN	Direct Fusion	Soft (Ours)	Hard (Ours)
Chess	0.24 m, 5.77°	0.16 m, NA	0.11 m, 4.66°	0.17 m, 5.67°	0.16 m, 5.30°	0.15 m, 5.46°	**0.14 m**, **5.02°**
Fire	0.34 m, 11.9°	0.19 m, NA	0.24 m, 9.60°	0.27 m, 10.3°	**0.26 m**, 10.2°	0.28 m, 10.3°	**0.26 m**, **9.80°**
Heads	0.21 m, 13.7°	0.13 m, NA	0.14 m, 12.2°	**0.14 m**, **12.0°**	0.16 m, 12.5°	0.15 m, 12.1°	0.15 m, 12.4°
Office	0.30 m, 8.08°	0.24 m, NA	0.17 m, 5.66°	0.26 m, 7.22°	0.24 m, 6.78°	**0.22 m**, 6.79°	0.23 m, **6.39°**
Pumpkin	0.33 m, 7.00°	0.33 m, NA	0.18 m, 4.44°	0.27 m, 5.81°	0.22 m, 5.10°	**0.21 m**, 4.97°	**0.21 m**, **4.93°**
Red Kitchen	0.37 m, 8.83°	0.28 m, NA	0.17 m, 5.94°	0.31 m, 6.76°	**0.25 m**, 6.41°	0.26 m, **6.36°**	**0.25 m** , 6.76°
Stairs	0.40 m, 13.7°	0.24 m, NA	0.26 m, 8.45°	0.34 m, **10.8°**	0.37 m, 11.8°	0.35 m, 11.9°	**0.30 m**, 11.3°
Average	0.31 m, 9.85°	0.23 m, NA	0.18 m, 7.28°	0.25 m, 8.37°	0.24 m, 8.30°	0.23 m, 8.27°	**0.22 m**, **8.08°**

- For a fair comparison, the bold character highlights the best results among our proposed approaches and common baselines, excluding task baselines.

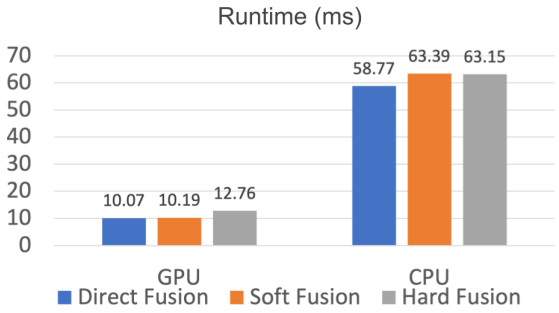


Fig. 7: The runtime of direct fusion, soft fusion and hard fusion model on a GPU (Geforce RTX 3080Ti) and a CPU (Intel Xeon 2.4G Hz) in the task of learning visual-inertial odometry.

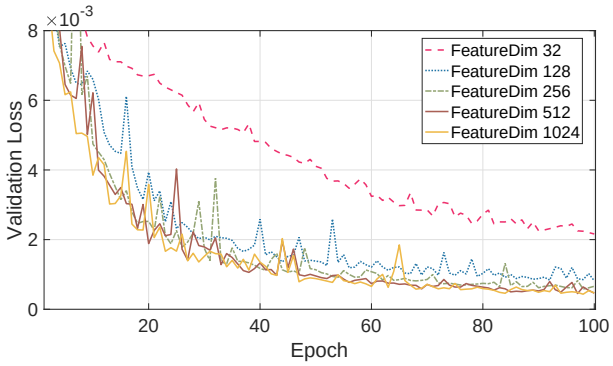


Fig. 8: The validation losses of proposed hard fusion model in terms of training epochs, with different feature dimensions of fusion module.

direct fusion. For hard fusion model, the runtime of each prediction is 12.76 ms and 63.15 ms on a GPU and CPU respectively. Thus, it can achieve 78 frames per second on a GPU and 15 frames per second on a CPU, which would satisfy the real-time requirement of most robotic applications.

One of the main hyper-parameters inside SelectFusion and baseline frameworks is the feature dimension of fusion module. It determines the dimension of extracted features in the feature extractors and the dimension of hidden states in the recurrent neural networks. To study the influence of this hyper-parameter, we test hard fusion model in the task of visual-inertial odometry with five different feature dimensions from 32 to 1024. Fig. 8 shows a comparison of the validation losses in terms of training epochs. Clearly, when increasing the feature dimension from 32 to 128, the validation loss is reduced dramatically. Further, the validation loss decreases slightly, when augmenting the feature dimension to 256, and 512. There is no clear change on validation loss if the feature dim is selected as 1024. Considering both model performance and memory storage, we thus use 512 as the feature dimension of fusion module in the following experiments.

C. Task 1: Global Relocalization using Vision and Depth

We first employ selective sensor fusion to combine visual and depth information for a global localization task in indoor scenarios. The features are extracted from RGB and depth images using the visual and depth feature encoders discussed in Section III. Our models, including the common baseline (direct fusion), are trained and evaluated on the 7-Scenes dataset. For each scene, we follow the official data split to train and test our models, and report for each model the median position and orientation error, according to the convention of prior works [1], [6], [22], [43].

Table I shows the results for the common baseline (Direct Fusion) and our proposed SelectFusion frameworks, i.e. soft fusion (Soft (Ours) and hard fusion (Hard (Ours)). For a fair comparison, the only difference in three models is the feature fusion part. Clearly, our proposed SelectFusion strategies outperform the common baselines, i.e. direct fusion and RFN. In particular hard fusion further improves the performance of the direct fusion with a gain of 8.33% in the position and 2.65% in the orientation. Although RFN performs best in the Heads scene and achieves most accurate orientation prediction in the Stairs scene, other fusion mechanisms still outperform it in the other scenes and averaged results. This shows the effectiveness of SelectFusion in learning multimodal representation for global relocalization.

In addition to the common baselines, we also choose three representative visual localization approaches as task baselines, i.e. LSTM-Pose [43], VidLoc [6] and MS-Transformer [36]. VidLoc can be viewed as a simple direct fusion, but it uses full-size images, and different feature encoders. Our proposed hard fusion model outperforms LSTM-Pose [43] and VidLoc [6], showing that our models can achieve competitive performance over previous works using only the uniform framework and proposed fusion strategies. Our method still can not compete with the state-of-the-art relocalization model, i.e. MS-Transformer with the transformer based architecture. It demonstrates that the performance of our fusion model can

TABLE II: The results of LIDAR-vision odometry (Task 2) on the KITTI Odometry dataset

Seq.	VISO2_M	ELO	Vision Only	LIDAR Only	RFN	Direct Fusion	Soft (Ours)	Hard (Ours)
05	19.2%, 17.6°	0.75%, 0.51°	6.14%, 2.84°	9.55%, 3.60°	5.55%, 2.22°	4.73%, 1.82°	4.65%, 1.83°	**4.25%**, **1.67°**
07	23.6%, 29.1°	0.60%, 0.48°	6.41%, 2.76°	8.63%, 3.75°	**4.19%**, 1.61°	4.31%, 2.34°	4.36%, 2.19°	4.46%, **2.17°**
10	41.6%, 33.0°	2.57%, 0.84°	6.93%, 2.97°	15.6%, 4.77°	10.3%, 2.42°	5.92%, 1.73°	8.35%, 2.01°	**5.81%**, **1.55°**
Ave.	28.1%, 26.7°	1.31%, 0.61°	6.49%, 2.85°	11.3%, 4.04°	6.69%, 2.08°	4.99%, 1.96°	5.78%, 2.01°	**4.84%**, **1.80°**

- $t_{rel}(\%)$ and $r_{rel}(^\circ)$ are the average translational and rotational RMSE drift (%) on lengths of 100m-800m.
- Vision-Only, LIDAR Only, RFN, Direct Fusion, Soft, and Hard models are trained on Sequence 00, 01, 02, 03, 04, 06, 08 and 09
- For a fair comparison, the bold character highlights the best results among our proposed approaches and common baselines, excluding task baselines.

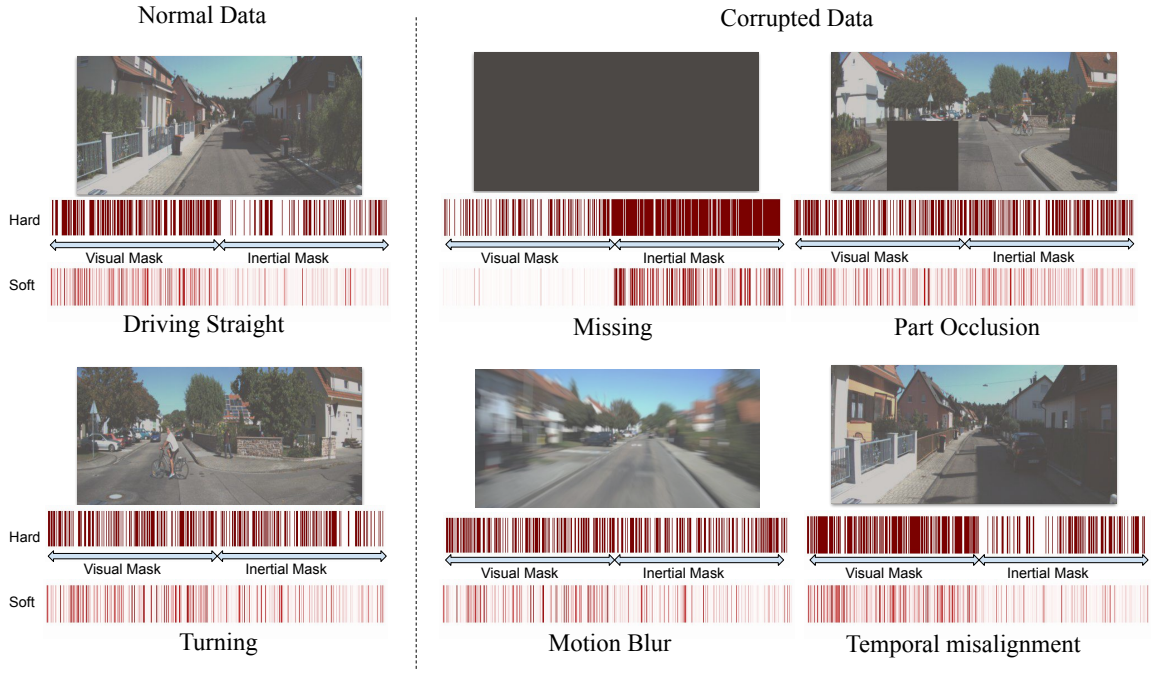


Fig. 9: Visualization of the learned hard and soft fusion masks under different conditions for Task 3 Deep VIO on self-driving scenarios (left: normal data; middle and right: corrupted data). The number (hard) or weights (soft) of selected features in the visual and inertial sides can reflect the self-motion dynamics (increasing importance of inertial features during turning), and data corruption conditions.

be improved with the advances in feature encoders.

D. Task 2: Deep LIDAR-Vision Odometry

We now focus on the problem of learning LIDAR-vision odometry in a car-driving scenario. The models are trained on the KITTI Odometry dataset and tested on three new sequences, i.e. Sequence 05, 07 and 10. Then the relative poses produced by the neural networks are integrated into global trajectories, which are further evaluated according to the official KITTI VO/SLAM evaluation metrics [11]. This metric is calculated by averaging the Root Mean Square Errors (RMSEs) of the translation and rotation for all the sub-sequences of lengths (100,..., 800) meters.

Table II shows the results of our deep LIDAR-vision odometry on the KITTI Odometry dataset. Vision Only and LIDAR Only models represent the model using only vision or LIDAR data to estimate ego-motion. Compared with them, fusing vision and LIDAR features (Direct Fusion) contributes to a large improvement no matter in translation or rotation. Soft (Ours) and Hard (Ours) are our frameworks with soft fusion and hard fusion. Our proposed hard fusion is capable of improving the performance over the naive fusion model (i.e. the direct fusion) about 3.0% in translation and 8.2% in orientation. RFN is able to predict the translation of Sequence 07 accurately, but its overall performance is not as good as other fusion mechanisms in this task. Note that these models are built on the same modules, except the feature fusion part for a fair comparison.

Meanwhile, three classical methods, i.e. VISO_M (Monocular Visual Odometry) [12], and ELO [52] are chosen as task baselines to compare with our data-driven approaches. As we can see, the learning based methods greatly outperform the two monocular visual odometry algorithms, but still have a large performance gap with respect to the traditional LIDAR odometry, i.e. ELO [52]. The model based methods are tailored to the specific visual odometry or LIDAR odometry problem: ELO is built on scene geometry information and quite accurate with good-quality point cloud data; the monocular visual odometry (VISO_M) relies on hand-crafted features and it is quite challenging to perform using high-dimensional raw images directly. In comparison, the data-driven models can automatically extract suitable features, which means that they

TABLE III: The results of visual-inertial odometry (Task 3) on the KITTI Raw dataset (car-driving scenario)

Seq.	MSCKF	VINS	VIOLearner	Vision Only	VIO (Direct)	VIO (Soft)	VIO (Hard)
05	19.0%, 82.5°	11.6%, 1.26°	3.00%, 1.40°	6.14%, 2.84°	4.18%, 1.57°	4.44%, 1.69°	**4.11%**, **1.49°**
07	89.9%, 126°	10.0%, 1.72°	3.60%, 2.06°	6.41%, 2.76°	3.39%, 1.79°	**2.95%**, **1.32°**	3.44%, 1.86°
10	42.0%, 134°	16.5%, 2.34°	2.04%, 1.37°	6.93%, 2.97°	2.80%, 1.69°	2.85%, 1.22°	**1.51%**, **0.91°**
Ave.	50.3%, 114°	12.7%, 1.77°	2.88%, 1.61°	6.49%, 2.85°	3.45%, 1.69°	3.41%, **1.41°**	**3.02%**, 1.42°

- $t_{rel}(\%)$ and $r_{rel}(^\circ)$ are the average translational (%) and rotational (°/100m) RMSE drift on lengths of 100m-800m.
- Vision-Only, VIO Direct, VIO Soft, and VIO Hard models are trained on Sequence 00, 01, 02, 04, 06, 08 and 09
- For a fair comparison, the bold character highlights the best results among our proposed approaches and common baselines, excluding task baselines.

are not restricted to a specific sensor modality or task, hence with the potential to explore an universal framework for deep state estimation.

E. Task 3: Deep VIO on UAV and self-driving scenarios

Finally, we come to evaluate our proposed model on the KITTI raw dataset (self-driving scenario) and EuRoC MAV dataset (UAV scenario) on learning visual-inertial odometry (VIO). These two datasets are challenging, as some real-world sensor degradations are contained in the original data: in the KITTI dataset, some IMU data are missing for a number of timesteps; IMU and camera streams are not tightly time-synchronized, which causes temporal sensor degradation; there are moving vehicles which act to partially occlude the camera; also in the Euroc dataset, there is significant motion blur and camera occlusion. Except the real sensor degradations, we also generate synthetic data degradations above the public datasets to study the robustness of learning models.

1) Synthetic Data Degradation: In order to provide an extensive study of the effects of sensor data degradation and to evaluate the performances of the proposed approach, we generate a degraded dataset, as shown in Figure 9, by adding various types of noise and occlusion to the original data, as described in the following.

1) Vision Degradation.

Occlusions: we overlay a mask of dimensions 128×128 pixels on top of the sample images, at random locations for each sample. Occlusions can happen due to dust or dirt on the sensor or stationary objects close to the sensor.

Motion Blur: we introduce motion blur to represent the camera blur caused by fast ego-motion or fast object movements. This motion blur is generated by estimating the relative motion of the scene, and producing corresponding blur above original images. Motion blur can happen when the camera or the light condition changes substantially.

Missing data: we randomly remove 10% of the input images. This can occur when packets are dropped from the bus due to excess load or temporary sensor disconnection. It can also occur if we pass through an area of very poor illumination e.g. a tunnel or underpass.

2) IMU Degradation.

Noise+bias: on top of the already noisy sensor data we add additive white noise to the accelerometer data and a fixed bias on the gyroscope data. This can occur due to increased sensor temperature and mechanical shocks, causing inevitable thermo-mechanical white noise and random walking noise.

Missing data: we randomly remove windows of inertial samples between two consecutive random visual frames. This can occur when the IMU measuring is unstable or packets are dropped.

3) Cross-Sensor Degradation.

Spatial misalignment: we randomly alter the relative rotation between the camera and the IMU, compared to the initial extrinsic calibration. This can occur due to axis misalignment and the incorrect sensor calibration. We uniformly model up to 10 degrees of misalignment .

Temporal misalignment: we apply a time shift between windows of input images and windows of inertial measurements. This can happen due to relative drifts in clocks between independent sensor subsystems.

TABLE IV: The results (m) of deep visual-inertial odometry (Task 3) on the EuRoC dataset (UAV scenario).

	Original	Vision Degrad.	All Degrad.
MSCKF	0.48	30.37	fail
OKVIS	0.47	1.42	fail
mono-VINS	0.35	fail	fail
Vision Only	2.42	2.44	1.99
VIO Direct	0.99	1.14	1.15
VIO Soft	1.06	1.18	1.21
VIO Hard	**0.84**	**1.04**	**1.12**

- The results (m) are reported the root mean squared error (RMSE) of the absolute translation error (ATE).
- Vision-Only, VIO Direct, VIO Soft, and VIO Hard models are trained on the sequences except MH_05_difficult of EuRoC MAV dataset [2] and tested on Sequence MH_05_difficult.
- For a fair comparison, the bold character highlights the best results among our approaches and common baselines, excluding task baselines.

TABLE V: The results of deep visual-inertial odometry (Task 3) on the KITII dataset (autonomous driving scenario)

	Original	Vision Degrad.	All Degrad.
Vision Only	6.49%, 2.85°	11.8%, 3.53°	8.06%, 3.18°
VIO Direct	3.45%, 1.69°	5.06%, 1.29°	3.62%, 1.28°
VIO Soft	3.41%, **1.41°**	**4.39%**, 1.84°	3.49%, 1.40°
VIO Hard	**3.02%**, 1.42°	4.76%, **1.12°**	**3.27%**, **1.29°**

- $t_{rel}(\%)$ and $r_{rel}(^\circ)$ are the average translational (%) and rotational (°/100m) RMSE drift on lengths of 100m-800m.
- Vision-Only, Direct, Soft, and Hard models are trained on Sequence 00, 01, 02, 04, 06, 08 and 09 of KITTI raw dataset [11] and tested on Sequence 05, 07 and 10.
- For a fair comparison, the bold character highlights the best results among our approaches and common baselines, excluding task baselines.

2) Experiment on the EuRoC Dataset: In the experiment of EuRoC dataset, we report the root mean squared error (RMSE) of the absolute translation error (ATE) of our models and baselines. This evaluation metric is commonly adopted by previous classical VIO works [18], [24], [34], so that our proposed frameworks can be compared with them directly. Table IV reports the performance of learning models (i.e. Vision-

TABLE VI: Effectiveness of different sensor fusion strategies in presence of different kinds of sensor data corruption for deep VIO

Model	Vision Degradation			IMU Degradation		Sensor Degradation	
	Occlusion	Blur	Missing	Noise and bias	Missing	Spatial	Temporal
Vision Only	7.23%, 2.81°	7.76%, 2.59°	27.6%, 9.20°	6.49%, 2.85°	6.49%, 2.85°	6.49%, 2.85°	6.49%, 2.85°
VIO Direct	4.24%, 1.77°	4.28%, 1.85°	5.61%, 1.32°	3.74%, 1.30°	3.59%, 1.74°	4.12%, 2.00°	3.27%, 1.55°
VIO Soft (Ours)	3.85%, **1.59°**	3.82%, 1.48°	6.42%, 2.02°	3.72%, **1.20°**	3.50%, 1.59°	3.45%, **1.46°**	3.43%, 1.72°
VIO Hard (Ours)	**3.77%**, 1.74°	**3.75%**, **1.33°**	**5.45%**, **1.26°**	**3.16%**, 1.64°	**3.18%**, **1.47°**	**3.22%**, 1.57°	**3.20%**, **1.31°**

- $t_{rel}(\%)$ is the average translational RMSE drift (%) on lengths of 100m-800m.
- $r_{rel}(^\circ)$ is the average rotational RMSE drift (°/100m) on lengths of 100m-800m.
- The Vision-Only, VIO Direct, VIO Soft, and VIO Hard models are trained on Sequence 00, 01, 02, 04, 06, 08 and 09 of KITTI raw dataset [11] with same hyperparameters for a fair comparison, and tested on Sequence 05, 07 and 10.

Only (DeepVO), VIO Direct (VINet)), and classical algorithms (i.e. MSCKF [18], OKVIS [24] and mono-VINS [34]) on the trajectory *MH_05_Difficult* in presence of normal data, all combined visual degradation (10% occlusion, 10% motion blur, and 10% missing data) and all combined visual+inertial degradation (5% for each). Note that learning models share the same framework and hyper-parameters, while the only difference is the fusion strategy. Details of data degradations can be found at the Section V-E1.

We can see that hard fusion consistently outperforms other learning models in all three scenarios, demonstrating the effectiveness of our proposed fusion strategy. In the normal set, hard fusion improves the direct fusion (our common baseline) by 15.15% in ATE. Another notable point is that OKVIS only shows a large performance decrease in the vision degradations, and fails on the all degradations, while mono-VINS fails on both degradation scenarios. In contrast, learning models all can work on degradation scenarios. This indicates that learning models are capable of overcoming sensor degradations to perform more robustly. Learning models still can not compete with the classical algorithms in normal set. This is due to two reasons: 1) in the EuRoC dataset, the groundtruth values of motion tracking (from a Vicon system) and sensor data (from a UAV) are collected from two time systems, and hence the training of learning models is limited because of the probable errors on ground-truth labels; 2) this deep VIO framework is still a basic framework, while extensions and constraints relevant to specific properties of visual and inertial sensors can be added onto it to further enhance the performance, e.g. Bundle adjustment.

3) Experiment on the KITTI Dataset: On the KITTI dataset, we use the official KITTI VO/SLAM evaluation metrics. This metric is to calculate the averaged RMSE of the translation and rotation for all the sub-sequences of lengths (100, ..., 800) meters, which can reflect both the global and local drifts of odometry estimation.

Table III reports the performance of our proposed hard fusion and soft fusion on the trajectories 05, 07 and 10 of normal dataset, together with two classical VIO algorithms (i.e. MSCKF and VINS) and three learning models, i.e. vision only (DeepVO), VIO direct (VINet) and VIOlearner. Our proposed selective sensor fusion (hard) further improves the averaged performance of the direct fusion by 12.46% in translation and 15.98% in orientation, while soft fusion shows an improvement of 1.16% and 16.57% in translation and

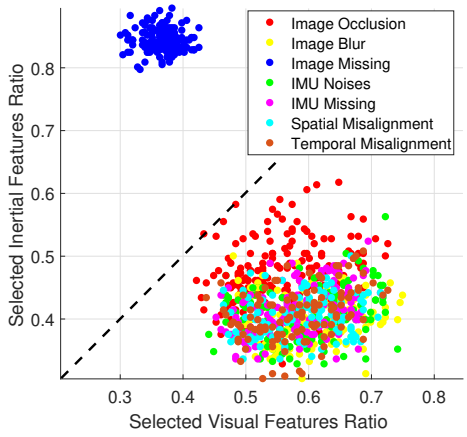
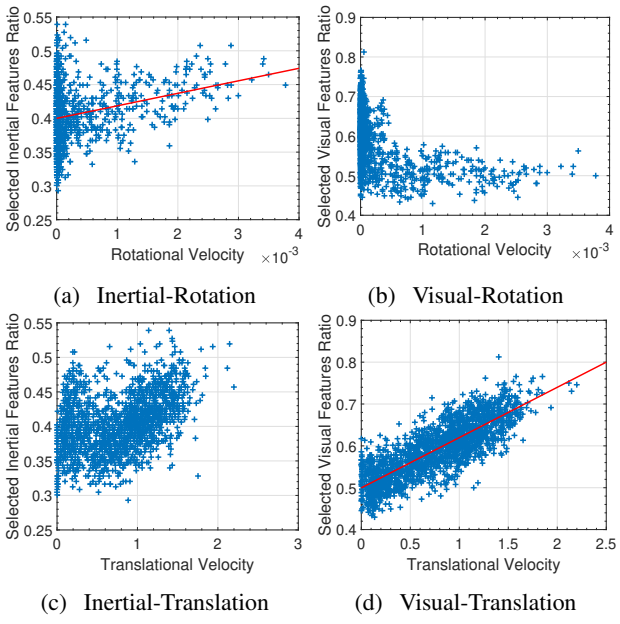


Fig. 10: A comparison of visual and inertial features selection rate in seven data degradation scenarios for Task 3.

rotation. Due to the real issue of bad time synchronization between images and IMUs, OKVIS and mono-VINS failed on the KITTI raw dataset. We then compare with a MSCKF implementation based on [18][2]. This approach, differently from OKVIS or VINS-Mono, is based on a trifocal tensor matching between triplets of successive frames, in order to get an ego-motion estimate, which is then fused with inertial information via a multi-state constraint Kalman filter to refine the estimates of the camera poses for each triplet. This sliding approach arguably makes it more robust to IMU de-synchronisation. It is clear to see that the learning based VIO models, including VIO (direct), VIO (soft) and VIO (hard) outperform MSCKF and VINS by a large margin. This is because MSCKF and VINS are limited by the bad time synchronization of two sensors, whilst the learning models are generally more robust to overcome such data degradations caused by real-world issues (data collection). Note that the original VIOlearner is trained on Sequence 00-08, and tested on Sequence 09 and 10 of the KITTI dataset, and thus it is not fairly to compare with our method directly. But our VIO (hard) still outperforms VIOLearner on Sequence 07 and 10.

Table V and VI show the performance of the proposed data fusion strategies, compared with the common baselines on the KITTI raw dataset. In particular, we compare with a DeepVO [45] (Vision-Only) implementation, and finally with

[2]The code can be found at https://uk.mathworks.com/matlabcentral/fileexchange/43218-visual-inertial-odometry



(a) Inertial-Rotation (b) Visual-Rotation

(c) Inertial-Translation (d) Visual-Translation

Fig. 11: Task 3: Correlations between the number of inertial/visual features and amount of rotation/translation show that the inertial features contribute more with rotation rates, e.g. turning, while more visual features are selected with increasing linear velocity.

an implementation of VINet [7] (VIO Direct), which uses a naïve fusion strategy by concatenating visual and inertial features. In the vision degraded set the input images are randomly degraded by adding occlusion, motion blurring and removing images, with 10% probability for each degradation. In the full degradation set, images and IMU sequences from the dataset are corrupted by all seven types of degradation with a probability of 5% each. Table V reports the results of deep VIO models in the presence of combined visual degradations, and all degradations. As we can see, our proposed selective sensor fusion, especially hard fusion, achieves better performance than the learning models without our proposed fusion module in these sensor degradation scenarios. In each data degradation, as illustrated in Table VI, our proposed selective sensor fusion, especially hard fusion consistently outperforms other baselines. This demonstrates to what extent our proposed SelectFusion can tolerant the perturbations of each data degradation, showing that SelectFusion is more robust to the issues caused by data degradations compared with other baselines.

F. Interpretation of Selective Fusion

Incorporating the hard mask into our framework enables us to quantitatively and qualitatively interpret the fusion process. Firstly, we analyse the contribution of each individual modality in different scenarios for deep visual-inertial odometry (Task 3). Since hard fusion blocks some features according to their reliability, in order to interpret the "feature selection" mechanism we simply compare the ratio of the non-blocked features for each modality. Figure 10 shows that visual features dominate compared with inertial features in most scenarios. Non-blocked visual features are more than 60%, underlining the importance of this modality. We see no obvious change when facing small visual degradation, such as image blur, because the FlowNet extractor can deal with such disturbances. However, when the visual degradation becomes stronger the role of inertial features becomes significant. Notably, the two modalities contribute equally in presence of occlusion. As it would be expected, inertial features dominate (by more than 90%) with missing images.

In Figure 11 we analyze the correlation between amount of linear and angular velocity and the selected features. These results also show how the belief on inertial features is stronger in presence of large rotations, e.g. turning, while visual features are more reliable with increasing linear translations. It is interesting to see that at low translational velocity (0.5m / 0.1s) only 50% to 60% visual features are activated, while at high speed (1.5m / 0.1s) 60 % to 75 % visual features are used.

VI. Conclusion and Future Research

We present a generic multimodal sensor fusion framework for deep states estimation, in support of odometry estimation and global relocalization tasks. Motivated by the need for robust interpretable sensor fusion in real-world applications, we propose two variants of selective fusion modules, i.e. a deterministic soft fusion and a Gumbel-softmax based hard fusion, that can be integrated in different neural network frameworks. It can learn to perform sensor fusion on feature space from pairs of different modalities, e.g. vision-depth, vision-LIDAR and vision-inertial data, conditioned on the input data itself. Extensive experiments illustrate that our proposed models outperform learning based single modality and multimodal model with direct fusion baselines, and also show competitive performance over other classical approaches, though the performance is still inferior to the domain-specific state-of-the-art in some cases. It also demonstrates that learning models are generally more robust than conventional hand-designed algorithms, and our proposed SelectFusion can further improve the performance of basic learning models, and achieve more accurate results than other baselines in these degraded sets. By interpreting the learned fusion masks, we investigate the influence of different modalities with different types of motion and different levels of sensor degradation.

Acknowledgments

The authors would like to thank Yishu Miao from MoIntelligence, Wei Wu from Tencent and Wei Wang from University of Oxford for helpful discussions. This work was done during Changhao Chen's Ph.D. and postdoctoral study at University of Oxford.

References

[1] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.

[2] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[3] C. Chen, C. X. Lu, A. Markham, and N. Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6468–6476, 2018.

[4] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10542–10551, 2019.

[5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[6] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2652–2660, 2017.

[7] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3995–4001, 2017.

[8] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, 2015.

[9] P. Fischer, E. Ilg, H. Philip, C. Hazırbas, P. V. D. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[10] A. Galassi, M. Lippi, and P. Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, 2021.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[12] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 963–968, 2011.

[13] J. Graeter, A. Wilczynski, and M. Lauer. Limo: Lidar-monocular visual odometry. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7872–7879, 2018.

[14] Y. Gu, K. Vyas, M. Shen, J. Yang, and G.-Z. Yang. Deep graph-based multimodal feature embedding for endomicroscopy image retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):481–492, 2021.

[15] L. Han, Y. Lin, G. Du, and S. Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6906–6913, 2019.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] C. Hori, T. Hori, T. Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-Based Multimodal Fusion for Video Description. *Proceedings of the IEEE International Conference on Computer Vision*, pages 4203–4212, 2017.

[18] J. S. Hu and M. Y. Chen. A Sliding-Window Visual-IMU Odometer Based on Tri-focal Tensor Geometry. In *Proceedings of the International Conference on Robotics and Automation*, pages 3963–3968, 2014.

[19] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations*, 2017.

[20] H. J. Kashyap, C. C. Fowlkes, and J. L. Krichmar. Sparse representations for object-and ego-motion estimations in dynamic scenes. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2521–2534, 2021.

[21] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5974–5983, 2017.

[22] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[23] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *Proceedings of the International Conference on Robotics and Automation*, pages 8943–8950, 2019.

[24] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[25] H. Li, X.-J. Wu, and J. Kittler. Rfn-nest: An end-to-end residual fusion network for infrared and visible images. *Information Fusion*, 73:72–86, 2021.

[26] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li. Lo-net: Deep real-time lidar odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8473–8482, 2019.

[27] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. In *Proceedings of the Conference on Robot Learning*, pages 249–261, 2017.

[28] Y. Liu, X. Zhang, F. Huang, L. Cheng, and Z. Li. Adversarial learning with multi-modal attention for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3894–3908, 2021.

[29] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[30] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the International Conference on Learning Representations*, 2017.

[31] C. J. Maddison, D. Tarlow, and T. Minka. A* Sampling. In *Advances in Neural Information Processing Systems*, pages 1–9, 2014.

[32] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *Proceedings of the International Conference on Machine Learning*, volume 32, pages 1791–1799.

[33] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[34] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[35] E. J. Shamwell, K. Lindgren, S. Leung, and W. D. Nothwang. Unsupervised deep visual-inertial odometry with online error correction for rgb-d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2478–2493, 2020.

[36] Y. Shavit, R. Ferens, and Y. Keller. Learning multi-scene absolute pose regression with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2733–2742, 2021.

[37] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[38] X. Shu, G.-J. Qi, J. Tang, and J. Wang. Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In *Proceedings of the ACM International Conference on Multimedia*, pages 35–44, 2015.

[39] J. Song, Y. Guo, L. Gao, X. Li, A. Hanjalic, and H. T. Shen. From deterministic to generative: Multimodal stochastic rnns for video captioning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):3047–3058, 2018.

[40] J. Tang, X. Shu, Z. Li, G. Qi, and J. Wang. Generalized deep transfer networks for knowledge propagation in heterogeneous domains. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12(4s):68:1–68:22, 2016.

[41] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[43] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers. Image-based localization using lstms for structured feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 627–637, 2017.

[44] M. Wang, X. Wang, L. T. Yang, X. Deng, and L. Yi. Multi-sensor fusion based intelligent sensor relocation for health and safety monitoring in bsns. *Information Fusion*, 54:61–71, 2020.

[45] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proceedings of International Conference on Robotics and Automation*, pages 2043–2050, 2017.

[46] Y. Wang, W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang. Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems*, 2020.

[47] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[48] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the International*

Conference on Machine Learning, pages 2048–2057, 2015.

[49] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha. Local supports global: Deep camera relocalization with sequence enhancement. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2841–2850, 2019.

[50] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

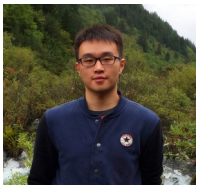
[51] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[52] X. Zheng and J. Zhu. Efficient lidar odometry for autonomous driving. *IEEE Robotics and Automation Letters*, 6(4):8458–8465, 2021.

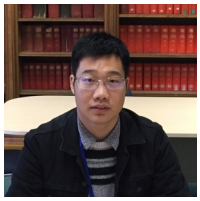
[53] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6612–6619, 2017.

Changhao Chen is a Lecturer at College of Intelligence Science and Technology, National University of Defense Technology. Before that, he obtained his Ph.D. degree at University of Oxford (UK), M.Eng. degree at National University of Defense Technology (China), and B.Eng. degree at Tongji University (China). His research interest lies in robotics, computer vision and cyberphysical systems.

Stefano Rosa is a research fellow at Istituto Italiano di Tecnologia (IIT). He was a postdoctoral researcher at University of Oxford, working on the EPSRC Programme Grant "Mobile Robotics: Enabling a Pervasive Technology of the Future". He achieved his MS degree in Computer Engineering from Politecnico di Torino in 2008, and his Ph.D. degree in Robotics from Istituto Italiano di Tecnologia (IIT). His research interests include localization and mapping for mobile robotics, computer vision applied to robot navigation, and human-robot interaction.



Chris Xiaoxuan Lu is an assistant professor at School of Informatics, University of Edinburgh. Before that, he obtained his Ph.D degree at University of Oxford, and MEng degree at Nanyang Technology University, Singapore. His research interest lies in Cyber Physical Systems, which use networked smart devices to sense and interact with the physical world.



Bing Wang is a Ph.D. student at Department of Computer Science, University of Oxford. Before that, he obtained his BEng Degree at Shenzhen University, China. His research interest lies in camera localization, feature detection, description & matching, and cross-domain representation learning.

Niki Trigoni is a professor at the Department of Computer Science, University of Oxford. She is currently the director of the EPSRC Centre for Doctoral Training on Autonomous Intelligent Machines and Systems, and leads the Cyber Physical Systems Group. Her research interests lie in intelligent and autonomous sensor systems with applications in positioning, healthcare, environmental monitoring and smart cities.

Andrew Markham is an associate professor at the Department of Computer Science, University of Oxford. He obtained his BSc (2004) and PhD (2008) degrees from the University of Cape Town, South Africa. He is the Director of the MSc in Software Engineering. He works on resource-constrained systems, positioning systems, in particular magneto-inductive positioning and machine intelligence.