

SelectFusion: A Generic Framework to Selectively Learn Multisensory Fusion

Changhao Chen, Stefano Rosa, Chris Xiaoxuan Lu, 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 also 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. During prediction, the network is able to assess the reliability of the latent features from different sensor modalities and estimate both trajectory at scale and global pose. In particular, we propose two fusion modules based on different attention strategies: deterministic soft fusion and stochastic hard fusion, and we offer a comprehensive study of the new strategies compared to trivial direct fusion. We evaluate all fusion strategies in both ideal conditions and on progressively degraded datasets that present 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, Visual-Inertial Odometry, Pointcloud Odometry

1 INTRODUCTION

Mobile agents are often outfitted with multiple sensors. For example, a self-driving vehicle is equipped with a combination of GPS, IMUs, monocular or stereo video cameras, LIDAR. Making such mobile agents fully autonomous and intelligent requires the ability of 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 [54], applying to a variety of tasks such as perception, planning and controlling. Despite different applications, the rationale of sensor fusion is more or less the same: many system state variables cannot be always observable by a single sensor modality, while different sensors can be complementary to each other. 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.

A typical example is integrating visual and inertial sensors in the form of Visual-Inertial Odometry (VIO) [14], [31], [32], [46], which enables ubiquitous mobility for mobile agents by providing robust and accurate pose information. These two sensors are relatively low-cost, light-weight, power-efficient and widely found in robots, smartphones, and VR/AR wearable devices. Single cameras are able to capture the appearance and structure of a 3D scene. However, they are scale-ambiguous, and not robust to most challenging scenarios, e.g. strong lightning changes, lack of textures and high-speed motions. In contrast, IMUs are completely ego-centric, scene-independent, and can provide absolute metric scale, but inertial measurements are corrupted by process noise and biases. Existing VIO approaches generally follow a standard

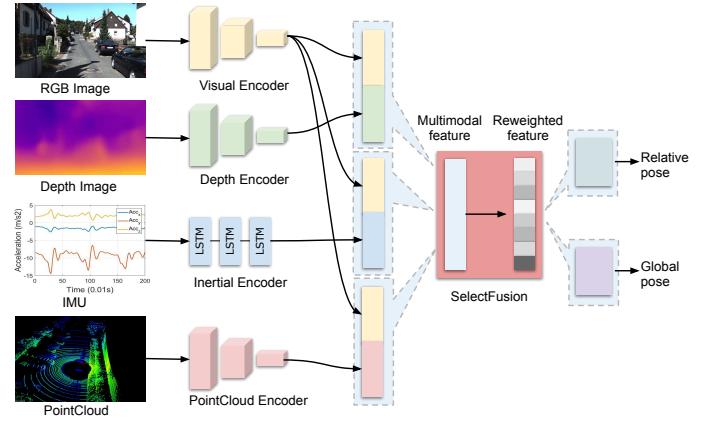


Fig. 1: An overview of the general framework to learn system states from multiple sensor modalities. Our framework can selectively utilize the suitable features for solving problems to improve both the accuracy and robustness. In our example, the network inputs a pair of sensor modalities from RGB image, depth image, inertial measurements, or point cloud data, and outputs relative pose or global locations.

pipeline that involves the fine-tuning of two modules, feature detection and tracking, and of the sensor fusion strategy. These methods rely on hand-crafted features, and the fusion strategy takes the form of Bayesian filtering [32], fixed-lag smoothers or full smoothers [14], [31], [46].

Recently, there is growing interest in applying deep neural networks (DNNs) for *learning to estimate system states* in an end-to-end manner, for example, solving visual-odometry (VO) [59], [72], visual-inertial odometry (VIO) [47], [49] or camera relo-

• All authors are with the Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom
E-mail: changhao.chen@cs.ox.ac.uk

calization [9], [28]. Instead of building analytical models by hand, they are achieved by learning complex mappings directly from raw sensory data to target values. These end-to-end approaches are appealing due to the capability of deep learning in automatic feature extraction from high-dimensional raw data. However, despite the long history of classical sensor fusion algorithms, there is a lack of effective fusion strategy on deep feature space, especially in the tasks of localization and odometry estimation. These previous learning-based methods are not explicitly modelling the sources of degradation in real-world usages. Without considering possible sensor errors, all features are directly fed into other modules for further pose regression in [4], [9], [28], or simply concatenated as in [47]. These factors can possibly cause troubles to the accuracy and safety of neural systems, when the 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 require 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. In this work, we mainly consider 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. Both architectures are trained in an end-to-end fashion.

By explicitly modelling the selection process, we are able to demonstrate the strong correlation between the selected features and the environmental/measurement dynamics by visualizing the sensor fusion masks, as illustrated in Figure 7. 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: the inertial features contribute more in presence of fast rotation, while visual features are preferred during large translations (Figure 10). Thus, the selective sensor fusion provides insights into the underlying strengths of each sensor modality, guiding future multimodal system design. We also 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 [7], and presents a generic framework for selective sensor fusion in multimodal deep pose estimation. The work focuses on an extensive analysis of the model performances and extends the fusion strategies from visual-inertial odometry to the problems of combined LIDAR-visual odometry and combined RGB and depth relocalization.

To summarise, the novel contributions of this work are as follows:

- We present 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 selective sensor fusion masks can be visualized and interpreted, providing deeper insight into the relative strengths of each stream, and guiding further 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 remainder of the paper is organized as follows: Section 2 contains a survey of related work; Section 3 presents a generic framework for multimodal sensor fusion; Section 4 introduces our proposed selective sensor fusion; Section 5 evaluates SelectFusion applied to three multimodal models for relocalization and trajectory estimation through extensive experiments; Section 6 finally draws conclusions.

2 BACKGROUND AND RELATED WORK

This section introduces some relevant prior work and state-of-the-art in both traditional and deep pose estimation, as well as deep multimodal sensor fusion and attention.

2.1 Model-based Pose Estimation

Visual-inertial odometry: Traditionally, visual-inertial odometry approaches can be roughly segmented into three different classes, according to the information fusion methods: filtering approaches [26], fixed-lag smoothers [31] and full smoothing methods [14]. Classical VIO approaches rely on the use of handcrafted visual features. OKVIS [31] presented a keyframe-based approach that jointly optimizes visual feature reprojections and inertial error terms. Semi-direct [53] and direct [56] methods have been proposed in an effort to move towards feature-less approaches, removing the feature extraction pipeline for increased speed. IMU-preintegration [14] provides a theoretical proof of how to avoid continuous preintegration of inertial measurements, thus improving computational speed. Recently, VINS-Mono [46] was proposed as a fast, tightly-coupled, sliding window-based optimization approach for VIO. Our approach shares with these techniques the idea of tuning a sensor fusion strategy.

Camera Relocalization: One of the most established approaches for visual localization is Structure-from-Motion (SfM) [19]. Once a 3D map of the environment is available from SfM, various methods proposed to exploit 3D-to-3D matching [71] or monocular-to-3D matching [48] between captured images and the map. To deal with dynamic environments and appearance variations, Experience-Based Navigation methods (EBN) [62] keep updated maps of the environment called "experience maps". In contrast to these methods, that must keep a memory of the environment, we propose to directly learn to discriminate between reliable and unreliable sensory inputs at prediction time, conditioned on the input itself.

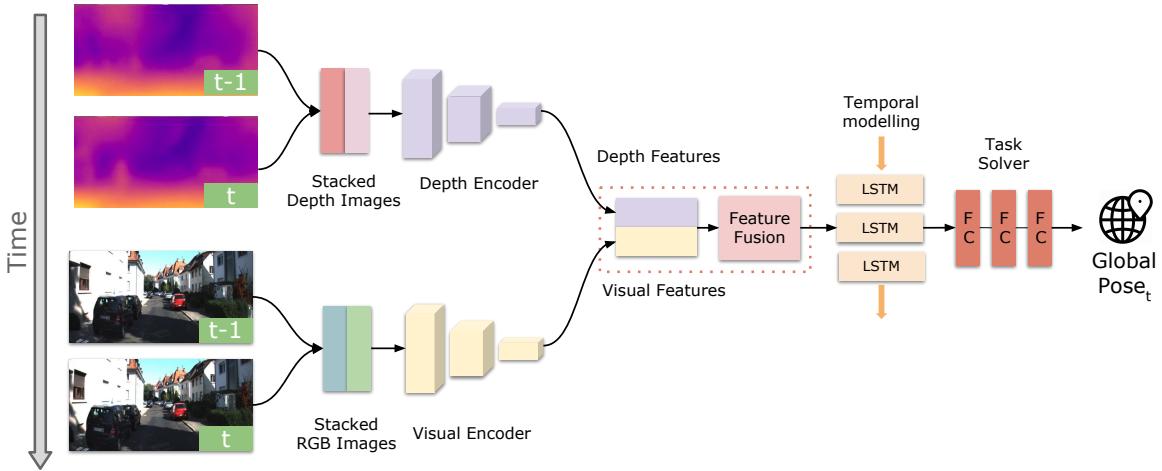


Fig. 2: An overview of our depth-vision relocalization (**Task 1**) architecture with proposed selective sensor fusion, consisting of depth and visual encoders, feature fusion, temporal modelling and task solver (global pose estimation).

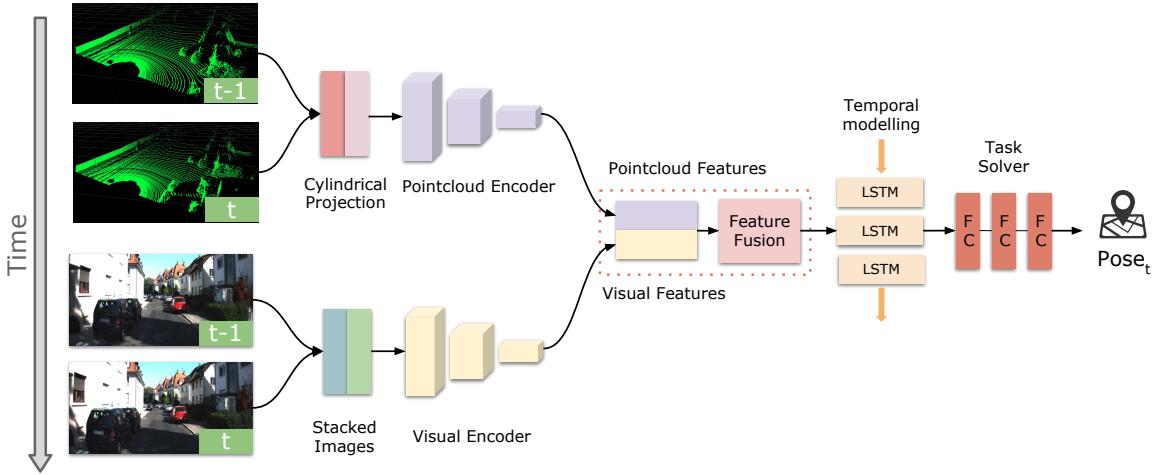


Fig. 3: An overview of our neural LIDAR-visual odometry (**Task 2**) architecture with proposed selective sensor fusion, consisting of visual and LIDAR encoders, feature fusion, temporal modelling and task solver (relative pose regression).

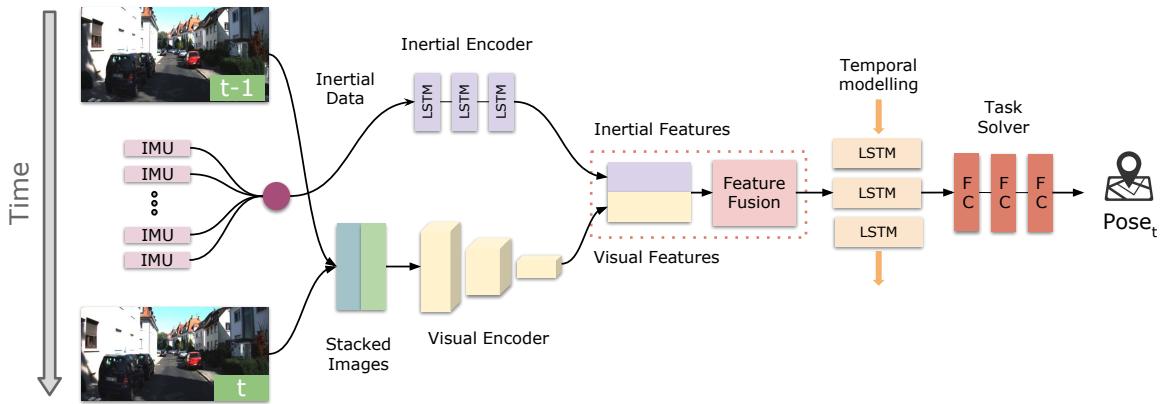


Fig. 4: An overview of our neural visual-inertial odometry (**Task 3**) architecture with proposed selective sensor fusion, consisting of visual and inertial encoders, feature fusion, temporal modelling and task solver (relative pose regression).

LIDAR odometry: LIDAR odometry can exploit the high accuracy of LIDAR sensors, but is sensitive to point cloud registration errors due to non-smooth motion. LOAM [69] relies on the fusion of LIDAR and IMU, and splits the problem into a high frequency/low accuracy process for motion estimation and a low frequency pose refinement process. To the same extent, [70] proposes instead to use fusion of LIDAR and monocular cameras.

2.2 Learning-based Pose Estimation

Visual odometry: Recent data-driven approaches to visual odometry have gained a lot of attentions. The advantage of learned methods is their potential robustness to lack of features, dynamic lightning conditions, motion blur, accurate camera calibration, which are hard to model by hand [52]. DeepVO [59], [60] utilized the combination of CNNs and Long-Short Term Memory (LSTM) networks to learn 6DoF visual odometry from a sequence of images, showing comparable results to traditional methods. [65] introduces a memory component that preserves global information via a feature selection strategy, and a refining component that improves previous predictions with a spatial-temporal attention mechanism based on current and past observations in memory. However, these methods cannot exploit additional sensory inputs such as inertial data. Several approaches [67], [68], [72] use view synthesis and geometric consistency checks [24] as an unsupervised signal in order to train and estimate both egomotion and monocular depth estimation. While joint trajectory and depth estimation shows promising results towards unsupervised visual odometry, the accuracy of such methods is still inferior to traditional visual odometry approaches.

Visual-inertial odometry: Recent work showed how it is possible to learn to estimate odometry from inertial data using recurrent neural networks [6], making deep visual-inertial odometry estimation possible. VINet [47] used neural network to learn visual-inertial odometry, by directly concatenating visual and inertial features. We observed that previous methods do not properly address the problem of learning a meaningful sensor fusion strategy, but simply concatenate visual and inertial features in the latent space. We argue that a gap between deep architectures and traditional model estimation techniques currently lies in a careful design of the fusion strategy. VIOLearn [49] presents an online error correction module for deep visual-inertial odometry that estimates the trajectory by fusing RGB-D images with inertial data. DeepVIO [18] recently proposed 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 in our approach we propose the use of an interpretable mask, that offers insight into the usefulness of the input at any time. Our approach bears more similarity to weighting data streams based on their belief.

Relocalization and SLAM: Deep approaches have also been devoted to visual localization. Posenet [28] was 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 [9], or by adding additional co-visibility constraints based on local maps and the estimated odometry [66]. Other works focus on learning deep representations for dense visual SLAM [3], general map [4], global pose estimation [44], simultaneous localization and segmentation [58].

LIDAR odometry: Learning LIDAR odometry has been explored by LO-Net [33], 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 [36]. Fusion of LIDAR and visual information has been investigated in [17], 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.

2.3 Multimodal learning, Sensor fusion and interpretability

Multimodal learning aims to solve machine learning problems involving multiple data modalities. The success of multimodal learning has been demonstrated in a wide range of applications, e.g. audio-visual speech classification [43] and recognition [22], face recognition [11], manipulation [30], and autonomous navigation [35]. 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 2.2.

Our proposed selective sensor fusion is particularly related to attention mechanisms, that have been widely applied in neural machine translation [55], image caption generation [64], and video description [21]. 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.

On the other hand, interpretability has become a desirable property for learned models, in particular for applications in which such models are used to inform critical decisions in the real world (e.g. navigation of autonomous vehicles). In these instances, black-box models are not adequate [45]. For this reason, interpretable attentive models are gaining traction [29].

3 LEARNING MULTIMODAL REPRESENTATIONS

In this section, we present a uniform framework to learn multimodal representation for state estimation, which lays the foundation for our proposed selective sensor fusion. Figure 2, 3 and 4, show a modular overview of the architecture, consisting of feature encoders (i.e. visual, depth, inertial, and pointcloud encoder), feature fusion, temporal modelling and task solver (i.e. odometry estimation or relocalization). Our model takes in a sequence of raw sensor data, and generates their corresponding system states, i.e. relative poses or global locations. With the exception of our novel feature fusion, the pipeline can be any generic deep state estimation system. In the Feature Fusion component, we propose two different selection mechanisms (soft and hard) and compare them with direct (i.e. a uniform/unweighted mask) fusion, as shown in Figure 5.

3.1 Feature Encoders

3.1.1 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

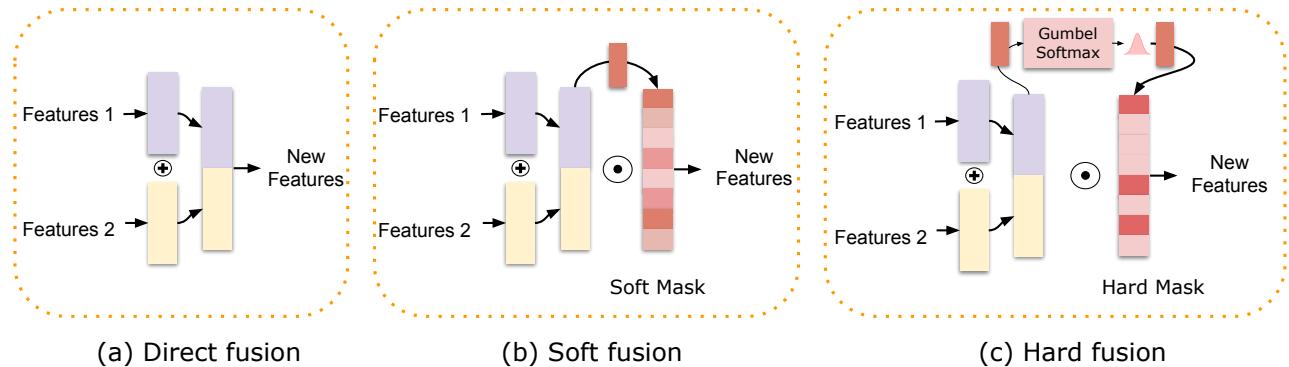


Fig. 5: An overview of three fusion methods: (a) direct fusion, (b) soft fusion and (c) hard fusion.

geometrically meaningful features rather than features overfitted with appearance or context. For this reason, instead of using a PoseNet model [28], as commonly found in other DL-based VO approaches [67], [68], [72], we use a FlowNet-style architecture, i.e. FlowNetSimple [13] as our feature encoder. Flownet provides features that are suited for optical flow prediction, which highly contributes to the 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. 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 dataset¹, since training from scratch would require larger amounts of data compared with our dataset size. The Visual Encoder (FlowNet) is employed to learn visual-inertial odometry and LIDAR-vision odometry, as shown in Figure 4 and 3.

For a global relocalization task, we instead use Residual Neural Network (ResNet) [20] 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 best 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 map the features after ResNet18 to a d dimension visual feature \mathbf{a}_V . The Visual Encoder (ResNet) is used in the depth-vision based relocalization, as illustrated in Figure 2.

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

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

3.1.2 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 the 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 $\mathbf{a}_I \in \mathbb{R}^d$:

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

3.1.3 Depth Feature Encoder

In our work, the 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 the depth feature encoder, but replace the first layer of ResNet model with a 1-channel convolutional network, considering that the 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} :

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

3.1.4 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 to effectively process point cloud data, we convert them into a regular point cloud matrix via the cylindrical projection [8], [33]:

$$\alpha = \arctan(y/x)/\Delta\alpha \quad (4)$$

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

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, the 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 $\mathbf{a}_P \in \mathbb{R}^d$:

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

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

3.2 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:

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

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, 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. Our selective fusion mechanisms re-weights the concatenated inertial-visual features, guided by the concatenated features themselves. The fusion network is another deep neural network and is end-to-end trainable. Details will be discussed in Section 4.

3.3 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 contains the history of the features relevant to the task. After the recurrent network, a fully-connected layer serves as the regressor, mapping 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.

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

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

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

3.3.1 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 presented by a 3-dimensional position vector $\mathbf{p} \in \mathbb{R}^3$ and a 4-dimensional quaternion based orientation vector $\mathbf{q} \in \mathbb{R}^4$. 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:

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

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 [27].

3.3.2 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 Pointcloud Encoder and Visual Encoder (FlowNet) that extract features from lidar pointcloud 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 :

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

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.

3.3.3 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 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 $\mathbf{r} \in \mathbb{R}^3$. By minimizing the MSE of the predicted relative poses, the optimal parameters θ are recovered via:

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

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.

4 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. For example, in the case of visual-inertial odometry, visual and inertial inputs contribute complementarily to the pose regression. Our perspective is that simply considering all features as though 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.

4.1 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:

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

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

4.2 Soft Fusion (Deterministic)

We now propose a soft fusion scheme that explicitly and deterministically models feature selection. Similar to the widely applied attention mechanism [21], [55], [64], 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 is introduced to implement soft selection of the extracted feature representations, before these features are passed to temporal modelling and task solver:

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

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

where $[\mathbf{a}_1; \mathbf{a}_2]$ denotes an MLP function that concatenates features \mathbf{a}_1 and \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. This process is deterministically parameterised by the neural networks, conditioned on both the features \mathbf{a}_1 and features \mathbf{a}_2 . The sigmoid function makes sure that each of the features will be re-weighted in the range $[0, 1]$.

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

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

4.3 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 discrete latent variables. To tackle this, the REINFORCE algorithm [39],

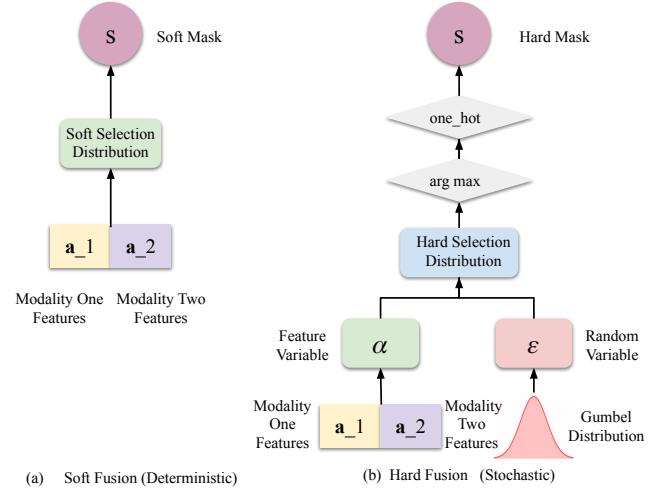


Fig. 6: An illustration of our proposed soft (deterministic) and hard (stochastic) feature selection process.

[63] is generally used to construct the gradient estimator. In our case, we propose to employ a more lightweight method – Gumbel-Softmax resampling [25], [37] 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.

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 α , which is learned by deep neural networks and conditioned on features but with the addition of stochastic noise:

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

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

where each mask $\mathbf{s} = [\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n)}]$ 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 i th feature or not. The total number of features is n . The element $\mathbf{s}^{(i)}$ can be viewed as resampling from a Bernoulli distribution:

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

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

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

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

$$x = \arg \max_k (\log \pi_k + \epsilon_k). \quad (20)$$

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 [38] allows us to efficiently draw a hard mask $\mathbf{s}^{(i)}$ 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:

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

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 ϵ is sampled from a Gumbel distribution, which is a continuous distribution on the simplex that can approximate categorical samples:

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

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

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

where $\tau > 0$ is the temperature that modulates the re-sampling process. Finally, $h^{(i)}$ is transformed into 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 $\alpha = (\pi_k^i)|_{k=1,2}^{i=1..n}$, conditioned on the concatenated feature vectors $[\mathbf{a}_1; \mathbf{a}_2]$ from two modalities:

$$\alpha = \text{ReLU}(\text{FC}([\mathbf{a}_1; \mathbf{a}_2])), \quad (24)$$

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

4.4 Discussions on Neural and Classical Sensor Fusion

In essence, soft fusion gently re-weights each feature in a deterministic way, while hard fusion directly blocks features according to the environment and its reliability. In general, soft fusion is a simple extension of direct fusion that is good for dealing with the uncertainties in the input sensory data. By comparison, the inference in hard fusion is more difficult, but it offers a more intuitive representation. The stochasticity gives the multimodal system better generalisation ability and higher tolerance to imperfect sensory data. The stochastic mask of hard fusion acts as an inductive bias, separating the feature selection process from prediction, which can also be easily interpreted by corresponding to uncertainties of the input sensory data.

Classical sensor fusion strategies normally rely on the hand-crafted physical models and algorithms. For example, in the case of visual-inertial odometry, filtering methods update their belief based on the past state and current observations of visual and inertial modalities [2], [23], [32], [40]. "Learning" within these methods is usually constrained to gain and covariances [1]. This is a deterministic process, and noise parameters are hand-tuned beforehand. Deep learning methods are instead fully learned from data and the hidden recurrent state only contains information relevant to the regressor. Our approach models the feature selection process explicitly with the use of soft and hard masks. Loosely, the proposed soft mask can be viewed as similar to tuning the gain and covariance matrix in classical filtering methods, but based on the latent data representation instead.

5 EXPERIMENTS

Our proposed selective sensor fusion is employed on three different tasks: vision-depth based global relocalization (task 1), deep LIDAR-vision odometry (task 2), and deep visual-inertial odometry (task 3). We show that our proposed framework is not restricted into specific modality or task. Moreover, we investigate the robustness of neural models under data degradation by generating data degradation above public dataset. In addition, our selective sensor fusion offers an interpretation of the fusion process via a visualization of the soft/hard fusion mask.

5.1 Datasets

We conducted extensive experiments above four well-known public datasets to learn from different pairs of sensor modalities: the 7-Scenes dataset [50] for vision-depth based relocalization, the KITTI odometry dataset [15] for vision-pointcloud based odometry estimation, the KITTI raw dataset [15] and the EuRoC MAV dataset [5] for vision-IMU based odometry estimation.

5.1.1 7-Scenes Dataset (vision+depth)

The 7-Scenes dataset [50] 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. It has been widely used as a common benchmark for camera relocalization. We follow the official data split to train and test our models above this dataset for global pose estimation. This dataset is used to learn relocalization from vision and depth images.

5.1.2 KITTI Odometry Dataset (vision+LIDAR)

The KITTI Odometry dataset [15] provides 11 sequences (00-10) with visual images, LIDAR point cloud and ground truth. 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. The images are resized to 512×256 . The challenges with this dataset are the relatively low frame rate (10Hz) for image data, the presence of many dynamic objects, high car speeds of up to 90 km/h and changing lightning conditions due to strong shadows cast by buildings and trees. We use this dataset to train deep vision-lidar odometry.

TABLE 1: The results of vision-depth relocalization (Task 1) on the 7-Scenes dataset, reported in position error (m) and orientation error ($^{\circ}$)

| Scene | PoseNet | LSTM-Pose | DSO | VidLoc (V) | VidLoc(V+D) | Ours (V+D) | Ours (Soft) | Ours (Hard) |
|-------------|--------------|--------------|--------------|------------|-------------|----------------------|----------------------|----------------------|
| Chess | 0.32 m, 8.12 | 0.24 m, 5.77 | 0.17 m, 8.13 | 0.18 m, NA | 0.16 m, NA | 0.16 m, 5.30 | 0.15 m, 5.46 | 0.14 m, 5.02 |
| Fire | 0.47 m, 14.4 | 0.34 m, 11.9 | 0.19 m, 65.0 | 0.21 m, NA | 0.19 m, NA | 0.26 m , 10.2 | 0.28 m, 10.3 | 0.26 m, 9.80 |
| Heads | 0.29 m, 12.0 | 0.21 m, 13.7 | 0.61 m, 68.2 | 0.14 m, NA | 0.13 m, NA | 0.16 m, 12.5 | 0.15 m, 12.1 | 0.15 m, 12.4 |
| Office | 0.48 m, 7.68 | 0.30 m, 8.08 | 1.51 m, 16.8 | 0.26 m, NA | 0.24 m, NA | 0.24 m, 6.78 | 0.22 m , 6.79 | 0.23 m, 6.39 |
| Pumpkin | 0.47 m, 8.42 | 0.33 m, 7.00 | 0.61 m, 15.8 | 0.36 m, NA | 0.33 m, NA | 0.22 m, 5.10 | 0.21 m, 4.97 | 0.21 m, 4.93 |
| Red Kitchen | 0.59 m, 8.64 | 0.37 m, 8.83 | 0.23 m, 10.9 | 0.31 m, NA | 0.28 m, NA | 0.25 m , 6.41 | 0.26 m, 6.36 | 0.25 m , 6.76 |
| Stairs | 0.47 m, 13.8 | 0.40 m, 13.7 | 0.26 m, 21.3 | 0.26 m, NA | 0.24 m, NA | 0.37 m, 11.8 | 0.35 m, 11.9 | 0.30 m, 11.3 |
| Average | 0.44 m, 10.4 | 0.31 m, 9.85 | 0.26 m, 29.4 | 0.25 m, NA | 0.23 m, NA | 0.24 m, 8.30 | 0.23 m, 8.27 | 0.22 m, 8.08 |

5.1.3 KITTI RAW dataset (visual+inertial)

The KITTI Raw dataset [15] contains the raw data collection from car-driving scenarios. High-frequency inertial data (100 Hz) is only available in the raw unsynchronized data package. We manually synchronized inertial data and images according to their timestamps, in order to exploit the visual and inertial data to learn odometry estimation. We used 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 is at 100 Hz. This dataset is adopted to learn visual-inertial odometry.

5.1.4 EuRoC MAV dataset (visual+inertial)

The EuRoC dataset [5] 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_04_difficult* for testing, and left the other sequences for training. We downsampled the images and IMUs to 10 Hz and 100 Hz respectively. This dataset is used for training deep visual-inertial odometry model.

5.2 Experimental Setup and Baselines

All networks were implemented with PyTorch and trained on a NVIDIA Titan X GPU. Our source code will be released here².

As baselines, we always choose a deep vision-only model and a multimodal model with direct fusion, plus additional state-of-the-art baselines according to the task. The vision-only model is composed by 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 multimodal features. All of the networks including baselines were trained with a batch size of 16 using the Adam optimizer, with a learning rate $\text{lr} = 1e^{-4}$. The hyper-parameters inside the networks were identical for a fair comparison. The single modality model and multimodal model with direct fusion can be viewed as an ablation study of our proposed approach. Besides these, several other representative methods are chosen as the baselines in each task: in vision-depth relocalization, we use the PoseNet [28], LSTM-Pose [57], Direct Sparse Odometry (DSO) [12] and VidLoc [9] to show the competitive performance of our models using SelectFusion; in vision-lidar odometry, we use the VISO2_M [16] and LOAM (LIDAR Odometry and Mapping)

[69] as baselines; in visual-inertial odometry, we use MSCKF [23] and OKVIS [31] as baselines.

5.3 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. The features are extracted from RGB and depth images using the visual and depth feature encoders discussed in Section 3. Table 1 shows the results of our proposed framework compared with direct fusion (Ours (V+D)), soft fusion (Ours (Soft)) and hard fusion (Ours (Hard)). For a fair comparison, the only difference in the three models is the feature fusion part. The performance of each model is reported in the mean error of the position and orientation, following the convention of prior work [9], [12], [28], [57].

Clearly, our proposed hard fusion further improves the performance of the direct fusion with 8.33% in the position and 2.65% in the orientation. We also choose four representative visual localization approaches as baselines, i.e. PoseNet [28], LSTM-Pose [57], Direct Sparse Odometry (DSO) [12] and VidLoc [9]. VidLoc can be viewed as a simple direct fusion, but it uses full-size images, and different feature encoders. Our proposed hard fusion model consistently outperform these methods, showing the effectiveness of our proposed selective sensor fusion in learning multimodal representation for global relocalization.

5.4 Task 2: Deep LIDAR-Vision Odometry

We now focus on the problem of learning LIDAR-vision odometry in a car-driving scenario. It is achieved by extracting effective features from point cloud data and RGB images for relative pose estimation. The pointcloud feature encoder and visual feature encoder (FlowNet-style) are used to process raw pointcloud data and RGB images respectively. Our proposed selective sensor fusion framework can be naturally extended to this task to automatically select useful features. 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 [15]. This metric is calculated by averaging the Root Mean Square Errors (RMSEs) of the translation and rotation for all the subsequences of lengths (100,..., 800) meters.

Table 2 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 (Ours (V+L)) contributes to a large improvement no matter in translation or rotation. Ours (V+L), Ours (Soft) and

2. https://github.com/changhaoc/selective_sensor_fusion

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

| Seq. | VISO2_M | | LOAM | | Vision Only | | LIDAR Only | | Ours (V+L) | | Ours (Soft) | | Ours (Hard) | |
|------|---------------|---------------------|---------------|---------------------|---------------|---------------------|---------------|---------------------|---------------|---------------------|---------------|---------------------|---------------|---------------------|
| | $t_{rel}(\%)$ | $r_{rel}(^{\circ})$ |
| 05 | 19.22 | 17.58 | 0.75 (0.57) | 0.38 | 4.74 | 1.89 | 9.55 | 3.60 | 4.73 | 1.82 | 4.65 | 1.83 | 4.25 | 1.67 |
| 07 | 23.61 | 29.11 | 0.69 (0.63) | 0.50 | 8.27 | 3.30 | 8.63 | 3.75 | 4.31 | 2.34 | 4.36 | 2.19 | 4.46 | 2.17 |
| 10 | 41.56 | 32.99 | 1.51 (0.79) | 0.57 | 9.18 | 1.89 | 15.59 | 4.77 | 5.92 | 1.73 | 8.35 | 2.01 | 5.81 | 1.55 |
| Ave. | 28.13 | 26.66 | 0.98 (0.66) | 0.48 | 7.40 | 2.36 | 11.26 | 4.04 | 4.99 | 1.96 | 5.78 | 2.01 | 4.84 | 1.80 |

- $t_{rel}(\%)$ is the average translational RMSE drift (%) on lengths of 100m-800m.
- $r_{rel}(^{\circ})$ is the average rotational RMSE drift ($^{\circ}/100m$) on lengths of 100m-800m.
- The Vision-Only, Lidar Only, Ours (V+L), Ours (Soft), and Ours (Hard) models are trained on Sequence 00, 01, 02, 03, 04, 06, 08 and 09 with same hyperparameters for a fair comparison.

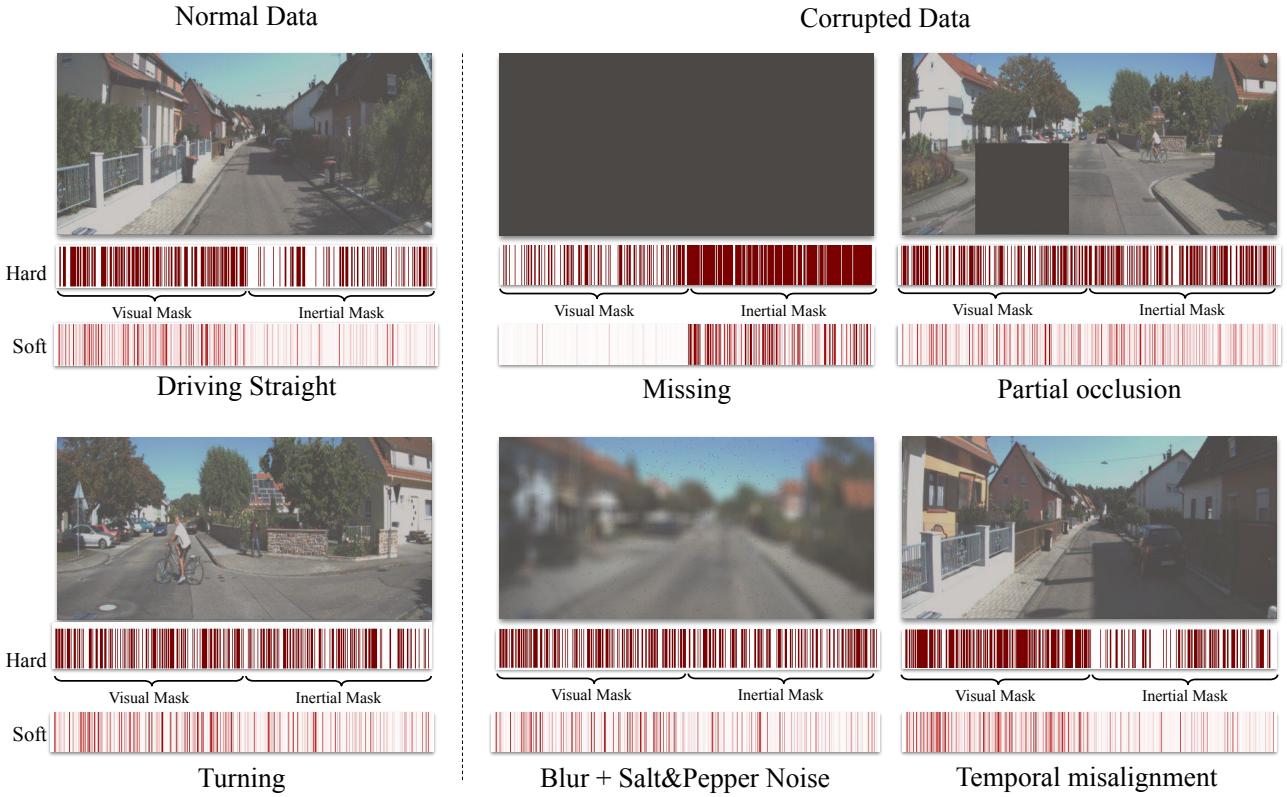


Fig. 7: Visualization of the learned hard and soft fusion masks under different conditions for Task 3 Deep VIO on self-driving scenarios (left: normal data; middle and right: corrupted data). The number (hard) or weights (soft) of selected features in the visual and inertial sides can reflect the self-motion dynamics (increasing importance of inertial features during turning), and data corruption conditions.

Ours (Hard) are our frameworks with a naive direct fusion, soft fusion and hard fusion. Our proposed hard fusion is capable of improving the performance over the naive fusion model about 3.0% in translation and 8.2% in orientation. Note that these models are built on the same modules, except the feature fusion part for a fair comparison. Meanwhile, two classical methods, i.e. VISO_M (Monocular Visual Odometry) [16] and LOAM (LIDAR Odometry and Mapping) [69] are chosen to compare with our data-driven approaches. As we can see, the learning based methods greatly outperform monocular visual odometry, but still have a certain gap with the state-of-the-art LIDAR odometry (LOAM). The model based methods are tailored to the specific visual odometry or LIDAR odometry problem: LOAM is built on scene geometry information and quite accurate with the good-quality pointcloud data; the monocular visual odometry (VISO_M) relies on hand-crafted features and is quite challenging above high-dimensional images. In comparison, the data-driven models can automatically

extract suitable features, which means that they are not restricted into specific sensor modality or task, leaving potentials to explore an universal framework for deep states estimator.

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

Finally, we come to evaluate our proposed model above KITTI raw dataset (self-driving scenario) and EuRoC MAV dataset (UAV scenario) on learning visual-inertial odometry (VIO), a fundamental research in robotic community. The relative pose error is adopted as evaluation metric [51], which calculates the root mean squared error (RMSE) of the translational and rotational transformations between two frames of images over all testing trajectories as:

$$\text{RMSE}(\mathbf{t}, \mathbf{r}) = (\sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{t} - \hat{\mathbf{t}}\|^2}, \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{r} - \hat{\mathbf{r}}\|^2}) \quad (25)$$

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

| | Normal Data | Vision Degradation | All Degradation |
|-------------|----------------------------|----------------------------|----------------------------|
| Vision Only | 0.00464 m, 0.0439° | 0.0119 m, 0.149° | 0.00973 m, 0.115° |
| VIO Direct | 0.00366 m, 0.0279° | 0.00912 m, 0.0303° | 0.00797 m, 0.0404° |
| VIO Soft | 0.00367 m, 0.0263 ° | 0.00874 m, 0.0285° | 0.00757 m, 0.0429° |
| VIO Hard | 0.00362 m, 0.0265° | 0.00928 m, 0.0276 ° | 0.00782 m, 0.0402 ° |

- The results are reported in the averaged translational RMSE (m) and rotational RMSE (°) between any two frames of images over the testing trajectories.
- The Vision-Only, VIO Direct, VIO Soft, and VIO Hard models are trained on the sequences except MH_04_difficult of EuRoC MAV dataset [5] with same hyperparameters for a fair comparison, and tested on Sequence MH_04_difficult.

TABLE 4: The results of deep visual-inertial odometry (Task 3) on autonomous driving scenario

| | Normal Data | Vision Degradation | All Degradation |
|-------------|-------------------------|-------------------------|-------------------------|
| Vision Only | 0.116 m, 0.136° | 0.177 m, 0.355° | 0.142 m, 0.281° |
| VIO Direct | 0.116 m, 0.106° | 0.175 m, 0.164° | 0.148 m, 0.139° |
| VIO Soft | 0.118 m, 0.098 ° | 0.173 m, 0.150 ° | 0.152 m, 0.134 ° |
| VIO Hard | 0.112 m, 0.110° | 0.172 m, 0.151° | 0.145 m, 0.150° |

- The results are reported in the averaged translational RMSE (m) and rotational RMSE (°) between any two frames of images over the testing trajectories.
- 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 [15] with same hyperparameters for a fair comparison, and tested on Sequence 05, 07 and 10.

TABLE 5: Comparison with classical methods for visual-inertial odometry

| | Normal data | Full visual degradation | Occlusion+blur | Full sensor degradation |
|-------|-------------------|-------------------------|-------------------|-------------------------|
| KITTI | 0.116 m, 0.044° | Fail | 2.4755 m, 0.0726° | Fail |
| EuRoC | 0.0283 m, 0.0402° | 0.0540 m, 0.0591° | 0.0198 m, 0.0400° | Fail |

- We implemented MSCKF on KITTI dataset and OKVIS on EuRoC Mav dataset in presence of normal data, full visual degradation, occlusion + blur and full sensor degradation as the baselines of classical methods.
- The results are reported in the averaged translational RMSE (m) and rotational RMSE (°) between any two frames of images over the testing trajectories.

where (\mathbf{t}, \mathbf{r}) are predicted relative poses, $(\hat{\mathbf{t}}, \hat{\mathbf{r}})$ are groundtruth values and n is the number of the transformations between two frames of prediction over all testing trajectories.

Table 3 reports the results on EuRoC MAV dataset in presence of normal data, all combined visual degradation and all combined visual+inertial degradation. The details of data degradation can be found at Section 5.6. In particular, we compare with two deep approaches: DeepVO (Vision-Only) and an implementation of VINet (VIO Direct). Compared with the direct fusion, our hard fusion can further improve the performance by 1% and 5% in the translation and rotation, while the soft fusion shows similar rotational performance as hard fusion, but worse translational performance. With all data degradation, our proposed hard fusion can consistently outperform VIO Direct, while soft fusion produces better translation in this case.

Table 4 shows the aggregate results of deep VIO in self-driving scenarios above KITTI raw dataset. In presence of normal data, vision degradation and all degradation, the hard fusion outperforms other baselines in translation, while the soft fusion shows better performance in rotation. Figure 8 shows a visual comparison of the resulting three test trajectories (Seq 05, Seq 07, Seq 10) in presence of visual and combined kinds of degradation. We compare the two VO and vanilla VIO baselines with the proposed soft and hard fusion strategies. It can be noticed how, while at start VIO performs as well as soft and hard fusion, on average, over time the proposed selective fusion strategies outperform the vanilla fusion, since the increased robustness reduces error accumulation. This is particularly visible in the most challenging Seq 05. As expected, a VO approach heavily underperforms in presence of large amounts of angular rotations (Seq 05, Seq 07). Another interesting result is how VO performs slightly better in presence

of IMU degradation and camera-IMU synchronization (Figure 8d and 8f). That shows how a vanilla VIO fusion is unable to deal with these issues, to the point of underperforming compared to vision-only approaches. This result further corroborates the fact that in deep learning-based approaches explicitly learning the belief on the different components makes the estimation more robust, while stacking sensors without a sensible fusion strategy can lead to catastrophic fusion, similarly to traditional approaches. Catastrophic fusion happens when the single components of the system before fusion significantly outperform the overall system after fusion [41].

5.6 Robustness to Data Degradation for Deep Visual-Inertial Odometry

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, by adding various types of noise and occlusion to the original data, as described in the following subsections.

5.6.1 Vision Degradation

In order to simulate the effects of occlusions, motion blur and missing frames that commonly affect video streams, we corrupt input images in three ways:

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 [61].

Blur+noise: we apply Gaussian blur with $\sigma=15$ pixels to the input images, with additional salt-and-pepper noise. Motion blur and noise can happen when the camera or the light condition changes

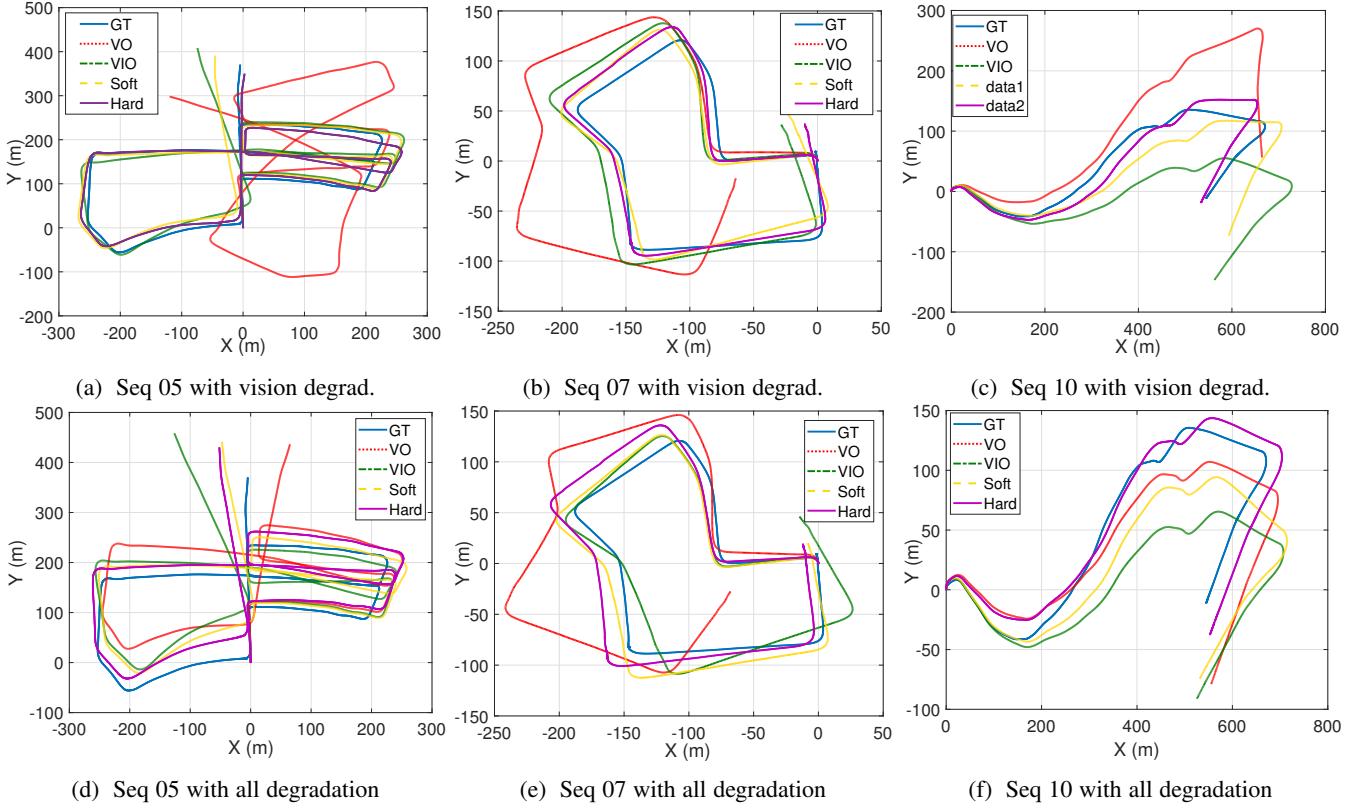


Fig. 8: Estimated trajectories on the KITTI dataset for Task 3 deep visual-inertial odometry (VIO). Top row: dataset with vision degradation (10% occlusion, 10% blur, and 10% missing data); bottom row: data with all degradation (5% for each). Here, GT, VO, VIO, Soft and Hard mean the ground truth, neural vision-only model, neural visual inertial models with direct, soft, and hard fusion. The neural VIO models with our proposed soft and hard selective masks showed better performance in terms of accuracy and robustness, compared with direct fusion and vision only model.

substantially [10].

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.

5.6.2 IMU Degradation

In order to simulate the effect of large unmodelled noise in inertial data, as well as missing samples, we degrade the inertial data in the following ways:

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 [42].

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 from the bus.

5.6.3 Cross-Sensor Degradation

We also model the two misalignment issues that commonly affect visual-inertial systems:

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 [32]. 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 [34].

5.6.4 Results on Data Degradation

Table 6 shows the relative performance of the proposed data fusion strategies, compared with the baselines. In particular, we compare with a DeepVO [59] (Vision-Only) implementation, and finally with an implementation of VINet [47] (VIO Direct), which uses a naive fusion strategy by concatenating visual and inertial features. In the vision degraded set the input images are randomly degraded by adding occlusion, blurring+noise 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. As a metric, we always report the average absolute error on relative translation and rotation estimates over the trajectory, in order to avoid the shortcomings of approaches using global reference frames to compute errors.

Some interesting behaviours emerge from Table 6. Firstly, as expected, both the proposed fusion approaches outperform VO and the baseline VIO fusion approaches when subject to degradation. Our intuition is that the visual features are likely to be local and discrete, and as such, erroneous regions can be blanked out, which

TABLE 6: 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 | 0.117 m, 0.148° | 0.117 m, 0.153° | 0.213 m, 0.456° | 0.116 m, 0.136° | 0.116 m, 0.136° | 0.116 m, 0.136° | 0.116 m, 0.136° |
| VIO Direct | 0.116 m, 0.110° | 0.117 m, 0.107° | 0.191 m, 0.155° | 0.118 m, 0.115° | 0.118 m, 0.163° | 0.119 m, 0.137° | 0.120 m, 0.111° |
| VIO Soft | 0.116 m, 0.105° | 0.119 m, 0.104° | 0.198 m, 0.149° | 0.119 m, 0.105° | 0.118 m, 0.129° | 0.119 m, 0.128° | 0.119 m, 0.108° |
| VIO Hard | 0.112 m , 0.126° | 0.114 m , 0.110° | 0.187 m , 0.159° | 0.114 m , 0.120° | 0.115 m , 0.140° | 0.111 m , 0.146° | 0.113 m , 0.133° |

- The results are reported in the averaged translational RMSE (m) and rotational RMSE (°) between any two frames of images over the testing trajectories.
- 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 [15] with same hyperparameters for a fair comparison, and tested on Sequence 05, 07 and 10.

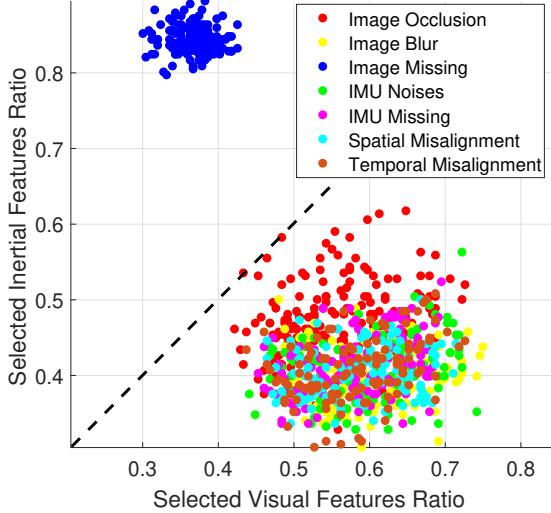


Fig. 9: A comparison of visual and inertial features selection rate in seven data degradation scenarios for Task 3.

would benefit the fusion network when it is predominantly relying on vision. Conversely, inertial data is continuous and thus a more gradual re-weighting as performed by the soft fusion approach would preserve these features better. As inertial data is more important for rotation, this could explain this observation. More interestingly, the soft fusion always improves the angle component estimation, while the hard fusion always improves the translation component estimation.

5.7 Comparison with Classical VIOs

For KITTI, due to the lack of tight time synchronization between IMU and camera, both OKVIS [31] and VINS-Mono [46] consistently fail. For this reason, we instead provide results from an implementation of MSCKF [23]³. For EuRoC MAV we compare with OKVIS [31]⁴.

As shown in Table 5, on KITTI, MSCKF also fails in presence of full degradation due to the missing images; on EuRoC, OKVIS is able to handle missing images but both baselines fail with full sensor degradation due to the temporal misalignment. Learning-based methods reach comparable position/translation errors, but the orientation error is always lower for traditional methods. Because DNNs shine at extracting features and regressing translation from raw images, while IMUs improve filtering methods to get better orientation results on normal data. Interestingly, the

3. The code can be found at: <https://uk.mathworks.com/matlabcentral/fileexchange/43218-visual-inertial-odometry>

4. The code can be found at: <https://github.com/ethz-asl/okvis>

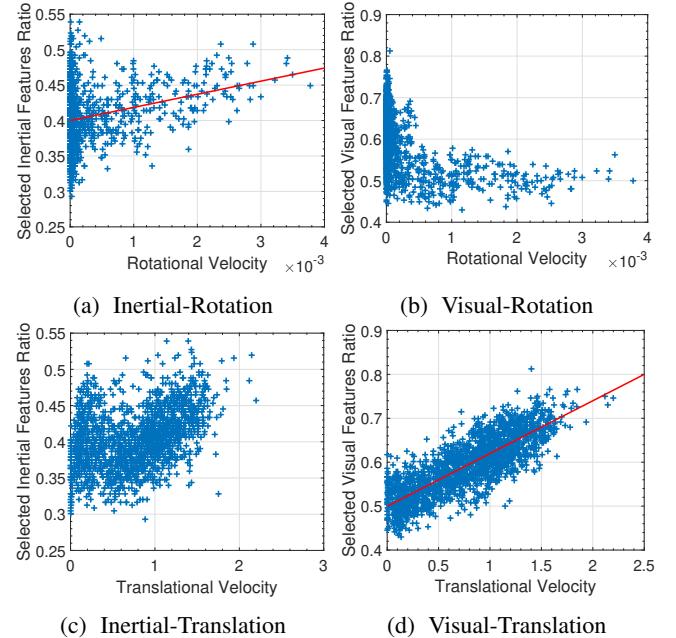


Fig. 10: Task 3: Correlations between the number of inertial/visual features and amount of rotation/translation show that the inertial features contribute more with rotation rates, e.g. turning, while more visual features are selected with increasing linear velocity.

performance of learning-based fusion strategies degrade gracefully in the presence of corrupted data, while filtering methods fail abruptly with the presence of large sensor noise and misalignment issues.

5.8 Interpretation of Selective Fusion

Incorporating 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 9 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 10 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.

6 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 proposed 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. The proposed model is not restricted to specific modality or task. 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 single modality and multimodal model with direct fusion baselines, and also show competitive performance over other classical approaches. In order to investigate the performance in various data degradation conditions, we extended two public datasets to include degraded and misaligned data streams, and study the influence of different modalities under different degradation and self-motion/environmental circumstances. In addition, we are able to provide insightful interpretations of fusion process by visualizing the learned masks. Future research directions would include an investigation of sensor fusion with three or more modalities. A detailed research into the relation between fusion masks and uncertainty estimation is also promising.

ACKNOWLEDGMENTS

This work was partially supported by EPSRC Program Grant Mobile Robotics: Enabling a Pervasive Technology of the Future (GoW EP/M019918/1). The authors would like to thank Yishu Miao from MoIntelligence and Wei Wu from Tencent for helpful discussions. The authors also thank Bing Wang and Wei Wang for their assistance on the experiments of vision-depth relocalization and vision-lidar odometry.

REFERENCES

- [1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [2] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. Iterated extended kalman filter visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017.
- [3] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM Learning a Compact, Optimisable Representation for Dense Visual SLAM. In *CVPR*, 2018.
- [4] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. In *CVPR*, pages 2616–2625, 2018.
- [5] 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*, 2016.
- [6] C. Chen, C. X. Lu, A. Markham, and N. Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [7] 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.
- [8] 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.
- [9] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization. In *CVPR*, 2017.
- [10] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce. Learning to estimate and remove non-uniform image blur. In *CVPR*, pages 1075–1082, 2013.
- [11] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, 2015.
- [12] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.
- [13] P. Fischer, E. Ilg, H. Philip, C. Hazrabs, P. V. D. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *International Conference on Computer Vision, ICCV*, 2015.
- [14] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.
- [15] 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.
- [16] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968. Ieee, 2011.
- [17] J. Graeter, A. Wilczynski, and M. Lauer. Limo: Lidar-monocular visual odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7872–7879. IEEE, 2018.
- [18] L. Han, Y. Lin, G. Du, and S. Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. *arXiv preprint arXiv:1906.11435*, 2019.
- [19] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [20] 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.
- [21] 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*, 2017-October:4203–4212, 2017.
- [22] D. Hu, X. Li, et al. Temporal multimodal learning in audiovisual speech recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3574–3582, 2016.
- [23] J. S. Hu and M. Y. Chen. A sliding-window visual-IMU odometer based on tri-focal tensor geometry. In *ICRA*, pages 3963–3968. IEEE, 2014.
- [24] G. Iyer, J. Krishna Murthy, G. Gupta, M. Krishna, and L. Paull. Geometric consistency for self-supervised end-to-end visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 267–275, 2018.
- [25] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [26] E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.
- [27] 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.
- [28] 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.
- [29] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.
- [30] 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 *2019 International Conference on Robotics and Automation (ICRA)*, pages 8943–8950. IEEE, 2019.
- [31] 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.
- [32] M. Li and A. I. Mourikis. High-precision, Consistent EKF-based Visual-

- Inertial Odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.
- [33] 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.
- [34] Y. Ling, L. Bao, Z. Jie, F. Zhu, Z. Li, S. Tang, Y. Liu, W. Liu, and T. Zhang. Modeling Varying Camera-IMU Time Offset in Optimization-Based Visual-Inertial Odometry. In *The European Conference on Computer Vision (ECCV)*, 2018.
- [35] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. *Conference on Robot Learning (CoRL 2017)*, 2017.
- [36] 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.
- [37] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
- [38] C. J. Maddison, D. Tarlow, and T. Minka. A* Sampling. In *NIPS*, pages 1–9, 2014.
- [39] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.
- [40] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.
- [41] J. R. Movellan and P. Mineiro. Robust sensor fusion: Analysis and application to audio visual speech recognition. *Machine Learning*, 32(2):85–100, 1998.
- [42] N. Naser, El-Sheimy; Haiying, Hou; Xiaojii. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 57(JANUARY):684–694, 2008.
- [43] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.
- [44] E. Parisotto, D. S. Chaplot, J. Zhang, and R. Salakhutdinov. Global Pose Estimation with an Attention-based Recurrent Network. In *CVPR*, 2018.
- [45] D. Pedreschi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini. Meaningful explanations of black box ai decision systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9780–9784, 2019.
- [46] 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, Aug 2018.
- [47] H. W. A. M. N. T. Ronald Clark, Sen Wang. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI, 2017.
- [48] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.
- [49] 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*, 2019.
- [50] 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 *CVPR*, pages 2930–2937, 2013.
- [51] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.
- [52] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke. The limits and potentials of deep learning for robotics. *International Journal of Robotics Research*, 37(4-5):405–420, 2018.
- [53] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges. Semi-direct ekf-based monocular visual-inertial odometry. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6073–6078. IEEE, 2015.
- [54] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.
- [55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In *NIPS*, 2017.
- [56] L. von Stumberg, V. Usenko, and D. Cremers. Direct sparse visual-inertial odometry using dynamic marginalization, 2018.
- [57] 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.
- [58] P. Wang, R. Yang, B. Cao, W. Xu, and Y. Lin. DeLS-3D: Deep Localization and Segmentation with a 3D Semantic Map. In *CVPR*, 2018.
- [59] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. *International Conference on Robotics and Automation*, 2017.
- [60] S. Wang, R. Clark, H. Wen, and N. Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5):513–542, 2018.
- [61] T. C. Wang, A. A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015.
- [62] H. Wen, R. Clark, S. Wang, X. Lu, B. Du, W. Hu, and N. Trigoni. Efficient indoor positioning with visual experiences via lifelong learning. *IEEE Transactions on Mobile Computing*, 18(4):814–829, 2018.
- [63] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- [64] 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 *ICML*, 2015.
- [65] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8575–8583, 2019.
- [66] 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.
- [67] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018.
- [68] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [69] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, page 9, 2014.
- [70] J. Zhang and S. Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.
- [71] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, volume 6, pages 33–40. Citeseer, 2006.
- [72] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.



Changhao Chen is currently a PhD student in Department of Computer Science, University of Oxford. Before that, he obtained his MEng degree at National University of Defense Technology, China, and BEng Degree at Tongji University, China. His research interest lies in machine learning, robotics and computer vision with special interests on localization, mapping and perception.



Dr. Stefano Rosa is currently a research fellow 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.



Dr. Xiaoxuan Lu is currently a PostDoctoral researcher at Department of Computer Science, University of Oxford. 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.



Prof. 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.



Prof. 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.