Fast Online Adaptation of Visual SLAM via Variational Information Transfer and Preservation

Sangni Xu
South China University of Technology
AU
saxu2211@uni.sydney.edu.au

Hao Xiong
Australian Institute of Health Innovation, Macquarie University
AU
hao.xiong@mq.edu.au

Qiuxia Wu
South China University of Technology, China
CN
qxwu@scut.edu.cn

zhihui wang
Dalian University of Technology
CN
zhwang@dlut.edu.cn

Shlomo Berkovsky
Australian Institute of Health Innovation, Macquarie University
AU
shlomo.berkovsky@mq.edu.au

Zhiyong Wang
The University of Sydney
AU
zhiyong.wang@sydney.edu.au

Abstract

Simultaneous Localisation and Mapping (SLAM) in computer vision involves estimating the camera poses and the surrounding depth information. Current deep learning based approaches achieve great success, yet most of them suffer from the domain generalisation issue. Accordingly, the online adaptation based methods have been proposed, enabling the SLAM model to continuously adapt to the changing open-world environments. However, these models are not computationally efficient while pursing accurate adaptation. In this work, we present a novel variational information transfer and preservation based visual SLAM method that aims to adapt fast while maintaining good precision. To reduce model size for faster adaptation, we introduce a lightweight network with a shared encoder for estimates of both poses and depths. To ensure adaptation precision, we exploit a large-sized network to pass our network the knowledge using a proposed information theory inspired knowledge distillation method that variationally maximizes the mutual information between the large network and ours. With pre-learned knowledge preservation, our model then learns to adapt against catastrophic forgetting by introducing the variational distribution of network weights pre-learned from knowledge distillation into the information bottleneck framework. During learning and adaptation, we keep these pre-learned weights fixed and utilise several adapters to adjust the feature representations instead. In terms of both speed and accuracy, our method surpasses several state-of-the-art baselines in evaluations of online visual SLAM adaptation.

CCS Concepts

• **Computing methodologies** → **Computer vision**.

Keywords

visual SLAM, self-supervised learning, online adaptation, information theory, information bottleneck, meta-learning

ACM Reference Format:

Sangni Xu, Hao Xiong, Qiuxia Wu, zhihui wang, Shlomo Berkovsky, and Zhiyong Wang. 2024. Fast Online Adaptation of Visual SLAM via Variational Information Transfer and Preservation. In *ACM Multimedia Asia (MMASIA '24), December 03–06, 2024, Auckland, New Zealand*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3696409.3700212

MMASIA '24, December 03–06, 2024, Auckland, New Zealand

ACM ISBN 979-8-4007-1273-9/24/12
https://doi.org/10.1145/3696409.3700212

1 Introduction

Visual SLAM plays a crucial role in various real-world applications, including robotics, autonomous vehicles, and augmented reality. Deep learning-based SLAM methods have emerged as powerful techniques, leveraging the capabilities of neural networks to achieve remarkable performance in mapping and localisation tasks. These works can be broadly classified into self-supervised [4, 5, 42] and supervised [34, 38] methods.

However, a common challenge among both self-supervised and supervised methods is their generalizability to the data from unknown domains, for which its environments and scenarios differ from the training data (as shown in Fig. 1). In real-world setting, it is imperative to enhance the generalizability of the model since the environments tend to be dynamic and continuously changing. Hence, domain adaptation (DA) techniques are exploited to bridge the domain discrepancy by aligning images, feature representations or decision boundaries across source and target domains [6, 24, 25]. These methods normally require the data of both source and target domains for offline adaptation learning. However, the offline learning is slow and is inapplicable to the real-world scenarios with constantly changing environments. In that case, learning and updating SLAM models on-the-fly is more desirable.

Accordingly, the online adaptation based visual SLAM methods [19, 26, 39, 40] have been proposed and demonstrate the superiority of continuous adaptation in an online manner. Despite that, it tends to be confronted with the catastrophic forgetting problem, for which the model forgets previously learned knowledge when adapting to the new data. Forgetting previously learned knowledge may causes the model becomes overly specialized to the new data, potentially losing its ability to generalize across different contexts

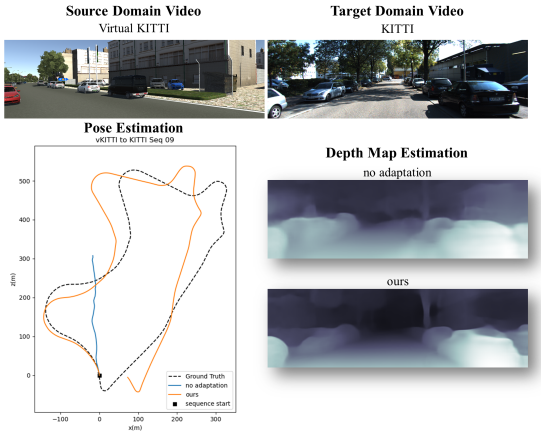

Figure 1: Existing visual SLAM approaches tend to suffer from domain shit problem. In contrast, our visual SLAM model performs faster and more accurate online adaptation to new environments.

and induce the overfitting. Other than catastrophic forgetting, these methods cannot maintain a good balance between the computational efficiency and the adaptation precision.

To attain fast online adaptation with high accuracy, we first harness a lightweight network with the shared encoder for both estimates of pose and depth. The lightweight architecture reduces model size and ensures fast adaptation but may loses precision. To improve adaptation quality, we propose a variational information transfer and preservation scheme that enhances the capability of our model by distilling knowledge from a large-sized network to our network in the pre-training step, and then preserve the pre-learned knowledge to prevent catastrophic forgetting in the learning-to-adapt step. In the pre-training step, we reformulate our model as a Bayesian neural network that exploits information theory to perform knowledge distillation by maximizing the mutual information between the larger network and ours. As a result, this Bayesian reformulation introduces uncertainty to the network weights where the transferred knowledge are reserved eventually. The reason we explicitly consider uncertainty into knowledge distillation is because the larger network may contain irrelevant knowledge to the task of our network, and such irrelevant information introduces the uncertainty affecting the performance. In next learning-to-adapt step, we explore it under the framework of information bottleneck which learns the latent encoding aiming to filter out irrelevant information from input data while maximally satisfying the tasks of depth and pose estimations. We further reformulate the information bottleneck framework by introducing the variational distribution of network weights learned from previous pre-training step, such that exploiting the pre-learned network parameters to avoid pre-learned knowledge forgetting and facilitate the adaptation. Rather than optimising the parameters of our pre-learned network, we keep them fixed and integrate our model with adapters to adjust the feature representations for adaptation. We train the learning-to-adapt step using the meta-learning algorithm [10]. After that, our model is able to adapt consistently and quickly to new scenarios in an online manner. Our contributions can be summarized as:

- We introduce an efficient network architecture including a shared encoder for fast online adaptation of visual SLAM.
- We propose a variational information transfer and preservation scheme that maintains accurate adaptation by harnessing the knowledge transferred from a larger model with considering uncertainty in the transferred information. Meanwhile, it exploits the variational distribution of pre-learned parameters to preserve pre-learned knowledge and facilitate adaptation against catastrophic forgetting.
- Comprehensive evaluations on two adaptation scenarios demonstrate the superiority of our proposed method over several state-of-the-art methods.

2 Related Work

2.1 Learning-based Visual SLAM

Learning-based visual SLAM can be broadly categorized as supervised SLAM and self-supervised SLAM. Supervised SLAM acquires the ability to predict depths and poses by learning from annotated data. In the realm of pose estimation, prior studies [7, 34, 38] utilized a combination of CNN and LSTM to extract both spatial and temporal features. Other than that, attention mechanisms were also employed by some methods [29, 36, 38] to extract non-local temporal features for visual SLAM. Unlike supervised visual SLAM, the self-supervised visual SLAM eliminates the need of ground truths for training. Traditional approaches [4, 42] applied two separate networks for estimates of poses and depths, and trained the network under the appearance consistency constraint. Based on these techniques, subsequent methods [27, 43] incorporated recurrent neural networks to capture temporal information, and the works of [28, 33] investigated a novel self-supervised loss with optical flow and additional geometric consistency constraints [4, 35] to address the scale inconsistency issue.

Though these methods demonstrated favorable performance on the training data, they failed to generalise well on unknown datasets. Therefore, the objective of our study is to enhance the generalisability of the visual SLAM model.

2.2 Domain Adaptation based Visual SLAM

Most existing visual SLAM methods rely on Generative Adversarial Networks (GAN) for domain adaptation. GAN can be either utilised to learn invariant representations across different domains [18, 22, 30] or align images from source and target domains, thereby achieving a harmonization of styles and appearances among different domains [6, 41]. Besides GAN, Gurram *et al.* [17] harnessed the Gradient-Reversal-Layer (GRL) [13] to learn domain invariant features. However, these approaches performed adaptation offline and cannot respond to continuously changing environments.

Instead of offline adaptation, some works [23, 26, 32, 37, 39, 40] have focused on the online visual SLAM adaptation. During continuous adaptation, the previously learned knowledge may be overwritten, and the catastrophic forgetting of pre-learned knowledge leads to the performance degradation. To mitigate this issue, a common solution was utilising a memory buffer to replay pre-learned

knowledge during adaptation [23, 32], yet the memory buffer requires additional hardware memory for information storage. Some other methods [26, 37, 39] used meta-learning to address this issue. For example, [39] introduced adapters to help preserve pre-learned knowledge, and train them within a meta-learning framework [10]. However, the existing online adaptation visual SLAM methods are either not memory-efficient to make them scalable or not sufficiently lightweight to run adaptation quickly.

2.3 Variational Information Bottleneck

The Information Bottleneck [31] aims to find a brief but comprehensive explanation by extracting from input the compressed representation that can maximally explains the task. In the field of deep learning, Alemi *et al.* [2] extended this idea and proposed the Variational Information Bottleneck (VIB) using a variational bound of the Information Bottleneck objective.

In works of [9, 20], VIB has demonstrated its potential in enhancing the representation learning. Some other studies investigated VIB in transfer learning [1] and fine-tuning [3], enabling to eliminate noises while extracting more relevant information to facilitate their tasks. To our knowledge, few studies apply VIB to SLAM or online adaptation. Inspired by this, in this work we explore incorporating VIB into the problem of visual SLAM online adaptation.

3 Method

Our approach includes three main steps: 1) pre-training with variational information transfer, 2) learning to adapt by preserving pre-learned knowledge and 3) online adaptation.

In step 1), our lightweight network is pre-trained on the source domain dataset using a novel information theory inspired knowledge distillation approach to achieve fast inference while maintaining accurate adaptation. The next step 2) preserves the knowledge learned from step 1) to prevent catastrophic forgetting, and introduce a number of adapters to learn to adapt under the framework of information bottleneck using the meta-learning. In step 3), our model is able to adapt quickly and continuously to the video from a target domain with changing environments.

3.1 Network Architecture

The overall framework of our method is illustrated in Fig. 2. We utilise the architecture as described in [37] which proposed two independent networks to predict depths and poses, respectively. Unlike [37], we reduce its model size for faster inference using a shared encoder of ResNet-50 and keep the same two decoders for separate estimates of depths and poses. For the teacher network, instead of using a shared encoder, we still exploit the two independent networks to maintain its performance. We further enhance its capacity using the larger ResNet-101 as the encoders, and accordingly the number of deconvolution layers in its decoders is increased as well. The large-sized teacher network ensures high precision and then helps enhance the performance of our network by distilling its knowledge to ours. To prevent forgetting past experiences, we introduce a number of adapters into our network for adaptation while keeping pre-learned network weights fixed. The adapters only contain the convolutional LSTM module, and are attached to blocks of the shared encoder and two decoders.

3.2 Pre-training with Variational Information Transfer

In the pre-training step, we aim to train our visual SLAM network on the source domain data, and then utilise the pre-trained network to perform fast online adaptation on the target video from a different domain. We train our model using the self-supervised loss $\mathcal{L}_s$ as defined in [37]. To ensure fast inference and adaptation, we exploit a lightweight architecture with the shared encoder to predict both depth maps and poses. Though achieving fast inference, the lightweight network is inclined to degrade the performance. We therefore harness a teacher network that distills its knowledge to our network so that retaining the performance of our model. Here, we regard our network as the student and the teacher is a large-sized network with higher knowledge capacity.

The overall loss of the pre-training stage is defined as a combination of the self-supervised learning and the information theory inspired knowledge distillation:

$$\mathcal{L}_{pre} = \mathcal{L}_s - \lambda_{dist} \sum_{i}^{K} I(\mathbf{z}_t^i; \mathbf{z}_s^i), \tag{1}$$

where $\mathcal{L}_s$ is the self-supervised loss and λ_{dist} is a hyper-parameter. The second term $I(\mathbf{z}_t^i; \mathbf{z}_s^i)$ defines the mutual information between the ith feature map $\mathbf{z}_t^i$ of the teacher network and the ith feature map $\mathbf{z}_s^i$ of the student network. Here, we formulate the knowledge distillation as maximizing the mutual information between the feature maps of student and teacher networks, and the feature maps are from a total of K layers in the network. It is noteworthy that $\mathbf{z}_t^i, \mathbf{z}_s^i$ refer to feature maps from either pose estimation network or depth estimation network.

In information theory, the mutual information $I(\mathbf{z}_t^i; \mathbf{z}_s^i)$ between $\mathbf{z}_t^i$ and $\mathbf{z}_s^i$ is defined as:

$$I(\mathbf{z}_t^i; \mathbf{z}_s^i) = \mathbb{E}_{(\mathbf{z}_t^i, \mathbf{z}_s^i) \sim p(\mathbf{z}_t^i, \mathbf{z}_s^i)} [\log p(\mathbf{z}_t^i | \mathbf{z}_s^i)] + H(\mathbf{z}_t^i), \tag{2}$$

where $H(\mathbf{z}_t^i)$ is the entropy term that is independent of our optimisation and can be ignored. Meanwhile, we further factorise the term $\log p(\mathbf{z}_t^i | \mathbf{z}_s^i)$, such that $\log p(\mathbf{z}_t^i | \mathbf{z}_s^i) = \log \int p(\mathbf{z}_t^i | \mathbf{z}_s^i, \theta_s) p(\theta_s) d\theta_s$ and θ_s denotes the weight of the student network. Due to the intractable integration in this term, we introduce a variational distribution $q(\theta_s)$ and apply the Jensen's inequality to obtain the evidence lower bound (ELBO) of $\log \int p(\mathbf{z}_t^i | \mathbf{z}_s^i, \theta_s) p(\theta_s) d\theta_s$ as:

$$\log \int p(\mathbf{z}_t^i | \mathbf{z}_s^i, \theta_s) p(\theta_s) d\theta_s \geq \int q(\theta_s) \log \frac{p(\mathbf{z}_t^i | \mathbf{z}_s^i, \theta_s) p(\theta_s)}{q(\theta_s)} d\theta_s. \tag{3}$$

For the ELBO in Eq. 3, we further apply the Monte Carlo integration using dropout as Bayesian approximation [12]:

$$\log \int p(\mathbf{z}_t^i | \mathbf{z}_s^i, \theta_s) p(\theta_s) d\theta_s \geq \log p(\mathbf{z}_t^i | \mathbf{z}_s^i, \hat{\theta}_s) - ||\theta_s||^2, \tag{4}$$

where $\hat{\theta}_s \sim q(\theta_s)$. After inserting Eq. 4 back to the Eq. 2, we have:

$$I(\mathbf{z}_t^i; \mathbf{z}_s^i) \geq \mathbb{E}_{(\mathbf{z}_t^i, \mathbf{z}_s^i) \sim p(\mathbf{z}_t^i, \mathbf{z}_s^i)} [\log p(\mathbf{z}_t^i | \mathbf{z}_s^i, \hat{\theta}_s)] - ||\theta_s||^2 + H(\mathbf{z}_t^i), \tag{5}$$

Similarly, $p(\mathbf{z}_t^i | \mathbf{z}_s^i, \hat{\theta}_s)$ in Eq. 5 is intractable, we thus introduce a variational distribution $q(\mathbf{z}_t^i | \mathbf{z}_s^i)$ to approximate it. Due to the fact that the KL divergence $D_{KL}(p(\mathbf{z}_t^i | \mathbf{z}_s^i, \hat{\theta}_s), q(\mathbf{z}_t^i | \mathbf{z}_s^i))$ is always

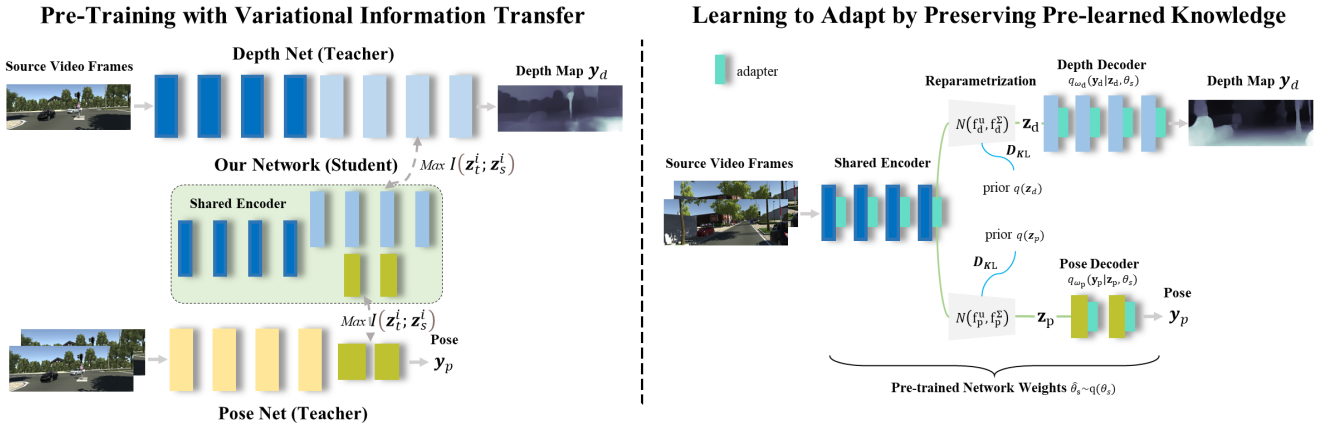

Figure 2: The proposed framework includes the pre-training step and the learning-to-adapt step. In the pre-training, we introduce the information theory inspired knowledge distillation $I(z_t^i; z_s^i)$ that maximizes the mutual information between ours and the larger teacher network to enhance our capability. Afterwards, the learning-to-adapt step keeps pre-learned weights θ_s fixed to preserve pre-learned knowledge and introduce adapters for adaptation under a framework of information bottleneck.

positive, we obtain:

$$I(\mathbf{z}_t^i; \mathbf{z}_s^i) \geq \mathbb{E}_{(\mathbf{z}_t^i, \mathbf{z}_s^i) \sim p(\mathbf{z}_t^i, \mathbf{z}_s^i)} [\log q(\mathbf{z}_t^i | \mathbf{z}_s^i)] - ||\theta_s||^2 + H(\mathbf{z}_t^i), \quad (6)$$

Here, $p(\mathbf{z}_t^i, \mathbf{z}_s^i)$ can be approximated using the empirical data distribution [2] $p(\mathbf{z}_t^i, \mathbf{z}_s^i) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{z}_t^i) \delta(\mathbf{z}_s^i)$. Then,

$$I(\mathbf{z}_t^i; \mathbf{z}_s^i) \geq \frac{1}{N} \sum_{n=1}^{N} \log q(\mathbf{z}_{t(n)}^i | \mathbf{z}_{s(n)}^i) - ||\theta_s||^2 - H(\mathbf{z}_t^i), \quad (7)$$

where N refers to the number of training samples and $q(\mathbf{z}_{t(n)}^i | \mathbf{z}_{s(n)}^i)$ is a Gaussian distribution with the identity matrix as the covariance matrix. Therefore, the log-likelihood here is equivalent to the L_2 norm of knowledge distillation. Other than that, our information theory based knowledge distillation $I(\mathbf{z}_t^i; \mathbf{z}_s^i)$ also introduces the uncertainty to network weights θ_s, for which θ_s in essence produces the feature map $\mathbf{z}_s^i$ for learning knowledge from the teacher network. By doing that, we consider the uncertainty induced by the potentially irrelevant knowledge transferred from the teacher. The variational distribution $q(\theta_s)$ containing pre-learned knowledge will be exploited by subsequent steps to facilitate online adaptation.

3.3 Learning to Adapt by Preserving Pre-Learned Knowledge

In this step, we exploit the information bottleneck principal to enhance the adaptation capability of our network by filtering out irrelevant information while passing through the information maximally explainable to target domain datasets. To prevent catastrophic forgetting, we keep fixed the network weights pre-learned via section 3.2 and introduce into our network a number of adapters that adjust feature representations for adaptation.

Let X, Y_p, Y_d, Z_p, Z_d denote input, pose prediction, depth prediction, latent encodings of pose and depth. Besides, we denote the parameters of adapters attached to the shared encoder, the depth estimation decoder and the pose estimation decoder as ω_e, ω_d and ω_p, respectively. Then, we define our learning objective as:

$$\mathcal{L}_{meta} = I(Z_d, Y_d; \omega_d) - \beta I(Z_d, X; \omega_e) + I(Z_p, Y_p; \omega_p) - \beta I(Z_p, X; \omega_e). \quad (8)$$

In Eq. 8, it minimises the mutual information between the input and learned latent encodings, while maximising the mutual information between latent encodings and predictions of depths and poses. However, Eq. 8 is intractable, as per [2] we first derive upper bounds of $I(Z_d, X; \omega_e)$ and $I(Z_p, X; \omega_e)$ using $q(\mathbf{z}_d)$ and $q(\mathbf{z}_p)$ as variational approximation to intractable $p(\mathbf{z}_d)$ and $p(\mathbf{z}_p)$:

$$\begin{aligned} I(Z_d, X; \omega_e) &\leq \mathbb{E}_{(\mathbf{z}_d, \mathbf{x}) \sim p(\mathbf{z}_d, \mathbf{x})} \left[\log \frac{p_{\omega_e}(\mathbf{z}_d | \mathbf{x})}{q(\mathbf{z}_d)} \right], \\ I(Z_p, X; \omega_e) &\leq \mathbb{E}_{(\mathbf{z}_p, \mathbf{x}) \sim p(\mathbf{z}_p, \mathbf{x})} \left[\log \frac{p_{\omega_e}(\mathbf{z}_p | \mathbf{x})}{q(\mathbf{z}_p)} \right], \end{aligned} \quad (9)$$

By introducing the variational distributions $q_{\omega_p}(\mathbf{y}_p | \mathbf{z}_p, \theta)$ and $q_{\omega_d}(\mathbf{y}_d | \mathbf{z}_d, \theta)$, lower bounds of $I(Z_d, Y_d; \omega_d)$ and $I(Z_p, Y_p; \omega_p)$ are:

$$\begin{aligned} I(Z_d, Y_d; \omega_d) &\geq \mathbb{E}_{(\mathbf{z}_d, \mathbf{y}_d) \sim p(\mathbf{z}_d, \mathbf{y}_d)} [\log q_{\omega_d}(\mathbf{y}_d | \mathbf{z}_d, \theta)] + H(\mathbf{y}_d), \\ I(Z_p, Y_p; \omega_p) &\geq \mathbb{E}_{(\mathbf{z}_p, \mathbf{y}_p) \sim p(\mathbf{z}_p, \mathbf{y}_p)} [\log q_{\omega_p}(\mathbf{y}_p | \mathbf{z}_p, \theta)] + H(\mathbf{y}_p), \end{aligned} \quad (10)$$

where $H(\mathbf{y}_d)$, $H(\mathbf{y}_p)$ refer to the entropy and are independent of the optimisation and so can be ignored. To preserve pre-learned knowledge against catastrophic forgetting, we take expectation with respect to $q(\theta_s)$ on both sides of Eq. 8. Here, $q(\theta_s)$ refers to the variational distribution of network weights θ_s and is obtained via the previous step of pre-training. Then, we have:

$$\begin{aligned} \mathcal{L}_{meta} \geq\ & \mathbb{E}_{\theta \sim q(\theta)} \mathbb{E}_{(\mathbf{z}_d, \mathbf{y}_d) \sim p(\mathbf{z}_d, \mathbf{y}_d)} [\log q_{\omega_d}(\mathbf{y}_d | \mathbf{z}_d, \theta)] \\ & + \mathbb{E}_{\theta \sim q(\theta)} \mathbb{E}_{(\mathbf{z}_p, \mathbf{y}_p) \sim p(\mathbf{z}_p, \mathbf{y}_p)} [\log q_{\omega_p}(\mathbf{y}_p | \mathbf{z}_p, \theta)] \\ & - \beta D_{KL}(p_{\omega_e}(\mathbf{z}_d | \mathbf{x}) || q(\mathbf{z}_d)) - \beta D_{KL}(p_{\omega_e}(\mathbf{z}_p | \mathbf{x}) q(\mathbf{z}_p)). \end{aligned} \quad (11)$$

By applying dropout as Bayesian approximation [12] and using empirical data distribution [2] $p(\mathbf{x}, \mathbf{y}_d) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x}) \delta(\mathbf{y}_d)$,

$p(\mathbf{x}, \mathbf{y}_p) = \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x})\delta(\mathbf{y}_p)$, we can approximate the lower bound in Eq. 11 as the following:

$$\begin{aligned}\mathcal{L}_{meta} \geq \frac{1}{N}\sum_{n=1}^{N}\Big[&\mathbb{E}_{\mathbf{z}_d|\mathbf{x}_n\sim p_{\omega_e}(\mathbf{z}_d|\mathbf{x}_n)}[\log q_{\omega_d}(\mathbf{y}_{d(n)}|\mathbf{z}_d,\hat{\theta}_s)]\\ &+\mathbb{E}_{\mathbf{z}_p|\mathbf{x}_n\sim p_{\omega_e}(\mathbf{z}_p|\mathbf{x}_n)}[\log q_{\omega_p}(\mathbf{y}_{p(n)}|\mathbf{z}_p,\hat{\theta}_s)]\\ &-\beta D_{KL}(p_{\omega_e}(\mathbf{z}_d|\mathbf{x}_n)||q(\mathbf{z}_d))-\beta D_{KL}(p_{\omega_e}(\mathbf{z}_p|\mathbf{x}_n)||q(\mathbf{z}_p))\Big],\end{aligned} \tag{12}$$

where $\hat{\theta}_s \sim p(\theta_s)$ and N is the number of training samples. We assume Gaussian distributions for $p_{\omega_e}(\mathbf{z}_d|\mathbf{x}) = \mathcal{N}(\mathbf{z}_d|f_d^{\mu}(\mathbf{x}), f_d^{\sigma}(\mathbf{x}))$ and $p_{\omega_e}(\mathbf{z}_p|\mathbf{x}) = \mathcal{N}(\mathbf{z}_p|f_p^{\mu}(\mathbf{x}), f_p^{\Sigma}(\mathbf{x}))$, where f_d and f_p refer to the network modules. To enable back-propagation, we utilise the reparameterization trick [21] to write $p_{\omega_e}(\mathbf{z}_d|\mathbf{x})d\mathbf{z}_d = p(\epsilon)d\epsilon$ and $p_{\omega_e}(\mathbf{z}_p|\mathbf{x})d\mathbf{z}_p = p(\varepsilon)d\varepsilon$, where $\mathbf{z}_d = f_d(\mathbf{x},\epsilon)$ and $\mathbf{z}_p = f_p(\mathbf{x},\varepsilon)$ are deterministic functions of $\mathbf{x}$ and Gaussian random variables ϵ, ε. The lower bound in Eq. 12 becomes:

$$\begin{aligned}\mathcal{L}_{adapt} = \frac{1}{N}\sum_{n=1}^{N}\Big[&-\mathbb{E}_{\epsilon\sim p(\epsilon)}\log q_{\omega_d}(\mathbf{y}_{d(n)}|f_d(\mathbf{x}_n,\epsilon),\hat{\theta}_s)\\ &-\mathbb{E}_{\varepsilon\sim p(\varepsilon)}\log q_{\omega_p}(\mathbf{y}_{p(n)}|f_p(\mathbf{x}_n,\varepsilon),\hat{\theta}_s)\\ &+\beta D_{KL}(p_{\omega_e}(\mathbf{z}_d|\mathbf{x}_n)||q(\mathbf{z}_d))+\beta D_{KL}(p_{\omega_e}(\mathbf{z}_p|\mathbf{x}_n)||q(\mathbf{z}_p))\Big],\end{aligned} \tag{13}$$

Here, the two terms of KL divergence are computationally analytic by defining $p_{\omega_e}(\mathbf{z}_d|\mathbf{x}_n)$, $q(\mathbf{z}_d)$, $p_{\omega_e}(\mathbf{z}_p|\mathbf{x}_n)$ and $q(\mathbf{z}_p)$ as Gaussian distributions. The other two negative log-likelihoods contain predictive functions for estimates of poses and depths and are defined by the self-supervised loss as decribed in [37]. We then optimise adapters to learn adaptation with the loss $\mathcal{L}_{adapt}$ using the meta-learning algorithm MAML [10]. To preserve pre-learned knowledge against catastrophic forgetting, we keep pre-learned network weights θ_s fixed and instead optimise adapter parameters including ω_e, ω_d and ω_p.

3.4 Online Adptation

The previous learning-to-adapt step equips our model with the capability of adapting online. Likewise, when performing adaptation upon target domain videos, we only update adapters such that the knowledge learned from the source domain can be preserved to facilitate the adaptation. In similar to the learning-to-adapt step, we harness the meta-learning method MAML [10] to update adapters.

4 Experiments

4.1 Implementation Details

Our model was implemented using PyTorch, and the input image size is 192×640. The dataset we used in our experiments include virtual KITTI [11], CityScapes [8], KITTI [14], and KITTI odometry [15]. In the pre-training step, our shared encoder was initialised on the ImageNet beforehand and was pre-trained on the source domain data for a total of 20,000 iterations. For the teacher network, we first trained it for 100 epochs with ground truths data, and then ran another 100 epochs using self-supervision. The learning rate of training both teacher and student was set to $1e^{-4}$.

Afterwards, in the learning-to-adapt step, the meta-learning algorithm MAML [10] was used to train all adapters. In this step, the learning rates of inner and outer optimisations were respectively set to $1e^{-4}$ and $1e^{-5}$. We used the Stochastic Gradient Descent (SGD) for the inner optimisation of the meta-learning, and used the Adam optimiser for the outer optimisation. The adapters were meta-trained for 10,000 iterations. The hyperparameters λ_{dist}, β were set to 5 and $1e^{-3}$. Meanwhile, the batch sizes of pre-training, learning-to-adapt and online adaptation were 4, 2, 1, respectively. In all steps, the length of the input sequence is 5.

4.2 Evaluation Metrics

Depth Evaluation: As previous methods [16, 42], we evaluate depth estimations using mean absolute relative error, average squared relative error, root mean squared error, root mean squared log error, and accuracy under thresholds at $\delta \in \{1.25^1, 1.25^2, 1.25^3\}$. Since self-supervised methods are unable to predict depth values with the absolute scale, we use the method of [42] to align the scaling of results produced by those methods with that of the ground truth.
Pose Evalutation: To assess the performance of pose estimations, we compute the average root mean square error (RMSE) [15] for both predicted translations and rotations. Likewise, the predicted poses are not in absolute scale and are thus adjusted by a scaling factor to match with the scale of ground truth poses.

4.3 Evaluation of Depth and Pose Estimation

We compare our method against several state-of-the-art online adaptation baselines including Zhang *et al.* [39], Li *et al.* [26], CL-SLAM [32], CoMoDA [23], Hornauer *et al.* [18], Xu *et al.* [37]. Our evaluations involve pre-training models using virtual KITTI, and then evaluated the online adaptation performance on KITTI and CityScapes, respectively.

Table 1 shows the comparison results of all methods that adapted from Virtual KITTI to KITTI. **Our method outperforms each individual baseline with respect to most of the metrics.** Take CL-SLAM [32] as an example, out of 11 metrics we only have 4 metric values inferior to those of [32]. This is remarkable because we exploit a lightweight network that not only adapts quickly but also demonstrates accurate adaptation. It benefits from our information theory based framework which captures more task-relevant information to ensure the accuracy. In Fig. 3, it illustrates estimated trajectories of KITTI dataset, and our approach is able to produce trajectories with less drifting from the ground truths.

Please refer to the supplementary material for adaptation results of Virtual KITTI to CityScapes.

4.4 Computational Efficiency

We utilise frames per second (fps), model size and FLOPS to evaluate computational efficiency. Here, fps indicates how many frames are processed in one second, model size counts the number of model parameters, and FLOPS measures the number of floating-point arithmetic calculations being performed by the processor within a second. As shown in Table 3, **our method is ranked as second best against each individual baseline in terms of evaluations**

Table 1: Quantitative comparison of depth and pose predictions in vKITTI to KITTI adaptation scenario.

Methods	Error (lower is better)				Accuracy (higher is better)			Seq 09		Seq 10	
	AbsRel ↓	SqRel ↓	RMSE ↓	RMSLog ↓	< 1.25 ↑	$< 1.25^2$ ↑	$< 1.25^3$ ↑	t_{rel} ↓	r_{rel} ↓	t_{rel} ↓	r_{rel} ↓
Zhang *et al.* [39]	0.153	-	5.508	-	0.776	0.923	-	-	-	-	-
Li *et al.* [26]	0.152	1.226	5.510	0.231	0.796	0.919	0.964	17.94	5.36	26.27	7.01
CL-SLAM [32]	0.153	1.218	5.506	0.223	0.802	0.923	0.965	16.68	5.24	22.07	8.35
CoMoDA [23]	0.155	1.23	5.521	0.227	0.787	0.920	0.964	21.57	6.13	27.42	10.08
Hornauer *et al.* [18]	0.179	1.478	6.499	0.263	0.727	0.908	0.962	-	-	-	-
Xu *et al.* [37]	0.151	1.221	5.507	0.224	0.801	0.925	0.965	17.51	4.98	24.97	7.97
ours	0.151	1.220	5.506	0.224	0.803	0.925	0.965	17.30	4.89	24.67	7.95

Table 2: Ablation study on key components of our visual SLAM model for online adaptation.

Methods	Error (lower is better)				Accuracy (higher is better)			Seq 09		Seq 10	
	AbsRel ↓	SqRel ↓	RMSE ↓	RMSLog ↓	< 1.25 ↑	$< 1.25^2$ ↑	$< 1.25^3$ ↑	t_{rel} ↓	r_{rel} ↓	t_{rel} ↓	r_{rel} ↓
Base	0.161	1.229	5.527	0.232	0.789	0.920	0.963	17.98	5.21	25.93	8.69
w/o VIT	0.152	1.223	5.508	0.227	0.801	0.922	0.964	17.54	4.97	24.81	8.03
w/o IB	0.151	1.225	5.508	0.228	0.797	0.922	0.963	17.64	5.17	25.16	8.22
ours	0.151	1.220	5.506	0.224	0.803	0.925	0.965	17.30	4.89	24.67	7.95

Table 3: Comparison of computational efficiency with our method and other baselines.

Method	fps ↑	Model Size (no. of parms) ↓	FLOPS ↓
Li *et al.* [26]	23	42.2M	7.2G
CL-SLAM [32]	18	113.5M	7.8G
CoMoDA [23]	24	66.1M	7.3G
Hornauer *et al.* [18]	25	43.6M	3.5G
Xu *et al.* [37]	18	89.2M	6.7G
ours	20	64.4M	5.4G

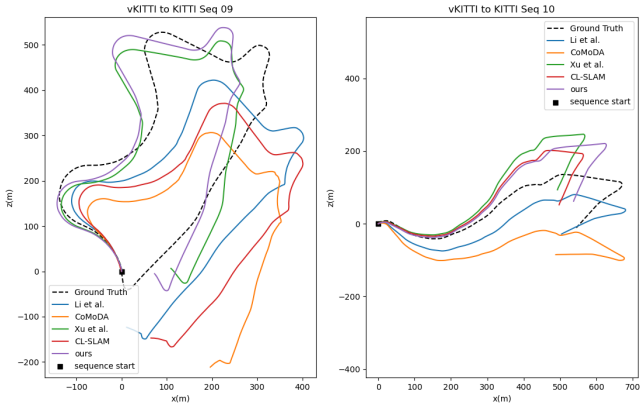

Figure 3: Trajectories of the moving camera predicted by different methods on KITTI dataset.

across all fps, model size and FLOPS. Although [18] is most efficient, it is unable to estimate poses, and its depth estimations are inferior to ours. Hence, our method is better than other baselines considering both efficiency and accuracy.

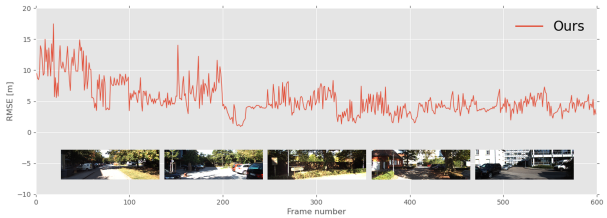

Figure 4: Illustration of long-range adaptation capability of our method. The x-axis denotes the number of testing frames, and the y-axis represents the value of RMSE error.

4.5 Ablation Study

We analyse the effectiveness of different components in our model, and use **Base** to denote the base network without proposed knowledge distillation and information bottleneck based framework. Besides, **w/o VIT** refers to substitute proposed knowledge distillation with the standard knowledge distillation, and **w/o IB** means removing our information bottleneck based framework only.

In Table 2, we observe a substantial performance degradation of **w/o VIT** when compared to **ours**. This is because substituting our knowledge distillation disregards the uncertainty in the transferred knowledge that may cause the performance degradation. Similarly, **w/o IB** shows worse results. This indicates our information bottleneck framework enables to extract domain-relevant features that enhance the adaptation. We also provide visual comparisons of depth estimations in the supplementary material.

4.6 Long-range Online Adaptation

In Fig. 4, we illustrate the long-term adaptation ability of our method by online adapting our model that is pre-trained on virtual KITTI to approximately 600 target frames of KITTI, and visualize the change of RMSE values across the adaptation. During adaptation, we can see that our model is able to effectively prevent catastrophic

forgetting. That is, the RMSE values of our method becomes clearly smaller after adapting to about 200 frames, and then the curve of RMSE values becomes stable to the end of all frames.

5 Conclusion

In this work, we propose a fast online adaptation method of visual SLAM. To achieve fast inference, we exploit a lightweight network architecture. Meanwhile, we introduce the information theory inspired knowledge distillation in the pre-training step to ensure the precision of our lightweight network by transferring knowledge from a larger teacher network with high knowledge capacity. Followed by that, we train our model to learn adaptation under the reformulated framework of information bottleneck by incorporating variational distribution of network weights pre-learned from the pre-training step. As a result, our model performs adaptation without forgetting past experiences and can also capture more domain-specific information to facilitate the adaptation. In terms of running speed and adaptation precision, extensive experiments demonstrate the superiority of our method over several state-of-the-art online visual SLAM adaptation baselines.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. 2019. Variational Information Distillation for Knowledge Transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *ICLR*. https://arxiv.org/abs/1612.00410

[3] Yonatan Belinkov, James Henderson, et al. 2020. Variational information bottleneck for effective low-resource fine-tuning. In *International Conference on Learning Representations*.

[4] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. 2019. Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video. In *NeurIPS*.

[5] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. 2021. Unsupervised Scale-consistent Depth Learning from Video. *International Journal of Computer Vision (IJCV)* (2021).

[6] Konstantinos Bousmalis, Nathan Silberman, David Dohan, D. Erhan, and Dilip Krishnan. 2017. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In *CVPR*. 95–104.

[7] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. 2017. VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. In *AAAI*.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*.

[9] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. 2020. Learning Robust Representations via Multi-View Information Bottleneck. In *International Conference on Learning Representations*.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

[11] A Gaidon, Q Wang, Y Cabon, and E Vig. 2016. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. In *CVPR*.

[12] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[13] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*.

[14] A Geiger, P Lenz, C Stiller, and R Urtasun. 2013. Vision meets robotics: The KITTI dataset. *IJRR* (2013).

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*.

[16] Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. 2019. Digging Into Self-Supervised Monocular Depth Estimation. In *ICCV*.

[17] Akhil Gurram, Ahmet Faruk Tuna, Fengyi Shen, Onay Urfalioglu, and Antonio M. López. 2021. Monocular Depth Estimation through Virtual-world Supervision and Real-world SfM Self-Supervision. *arXiv preprint arXiv:2103.12209* (2021). arXiv:2103.12209

[18] Julia Hornauer, Lazaros Nalpantidis, and Vasileios Belagiannis. 2021. Visual Domain Adaptation for Monocular Depth Estimation on Resource-Constrained Hardware. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 954–962.

[19] Pan Ji, Runze Li, Bir Bhanu, and Yi Xu. 2021. MonoIndoor: Towards Good Practice of Self-Supervised Monocular Depth Estimation for Indoor Environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 12787–12796.

[20] Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. 2023. How does information bottleneck help deep learning?. In *Proceedings of the 40th International Conference on Machine Learning*.

[21] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[22] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. 2018. AdaDepth: Unsupervised Content Congruent Adaptation for Depth Estimation. In *CVPR*.

[23] Yevhen Kuznietsov, Marc Proesmans, and Luc Van Gool. 2021. CoMoDA: Continuous Monocular Depth Adaptation Using Past Experiences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

[24] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. 2021. Unsupervised Domain Adaptation for Semantic Segmentation by Content Transfer. In *AAAI*. 8306–8315.

[25] Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. 2020. Domain Conditioned Adaptation Network. In *AAAI*.

[26] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. 2020. Self-Supervised Deep Visual Odometry With Online Adaptation. In *CVPR*.

[27] Shunkai Li, Fei Xue, Xin Wang, Zike Yan, and Hongbin Zha. 2019. Sequential Adversarial Learning for Self-Supervised Deep Visual Odometry. In *ICCV*.

[28] Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. 2021. Transformer Guided Geometry Model for Flow-Based Unsupervised Visual Odometry. *arXiv preprint arXiv:2101.02143* (2021).

[29] E. Parisotto, D. S. Chaplot, J. Zhang, and R. Salakhutdinov. 2018. Global Pose Estimation with an Attention-Based Recurrent Network. In *CVPRW*.

[30] Koutilya PNVR, Hao Zhou, and David Jacobs. 2020. SharinGAN: Combining Synthetic and Real Data for Unsupervised Geometry Estimation. In *CVPR*.

[31] Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. The Information Bottleneck Method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*. 368–377.

[32] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. 2023. Continual SLAM: Beyond Lifelong Simultaneous Localization and Mapping Through Continual Learning. In *Robotics Research*, Aude Billard, Tamim Asfour, and Oussama Khatib (Eds.).

[33] Rui Wang, Stephen M. Pizer, and Jan-Michael Frahm. 2019. Recurrent Neural Network for (Un-)Supervised Learning of Monocular Video Visual Odometry and Depth. In *CVPR*.

[34] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. 2017. DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. In *ICRA*.

[35] Mingkang Xiong, Zhenghong Zhang, Weilin Zhong, Jinsheng Ji, Jiyuan Liu, and Huilin Xiong. 2020. Self-supervised Monocular Depth and Visual Odometry Learning with Scale-consistent Geometric Constraints. In *IJCAI*.

[36] Sangni Xu, Hao Xiong, Qiuxia Wu, and Zhiyong Wang. 2021. Attention-based Long-term Modeling for Deep Visual Odometry. In *2021 Digital Image Computing: Techniques and Applications (DICTA)*. 1–8. https://doi.org/10.1109/DICTA52665.2021.9647140

[37] Sangni Xu, Hao Xiong, Qiuxia Wu, Tingting Yao, Zhihui Wang, and Zhiyong Wang. 2023. Online Visual SLAM Adaptation against Catastrophic Forgetting with Cycle-Consistent Contrastive Learning. In *2023 IEEE International Conference on Robotics and Automation*. 1–7.

[38] Fei Xue, Xin Wang, Shunkai Li, Qiuyuan Wang, Junqiu Wang, and Hongbin Zha. 2019. Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry. In *CVPR*.

[39] Zhenyu Zhang, Stephane Lathuiliere, Elisa Ricci, Nicu Sebe, Yan Yan, and Jian Yang. 2020. Online Depth Learning Against Forgetting in Monocular Videos. In *CVPR*.

[40] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. 2020. Towards Better Generalization: Joint Depth-Pose Learning Without PoseNet. In *CVPR*.

[41] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. 2018. T2Net: Synthetic-to-Realistic Translation for Solving Single-Image Depth Estimation Tasks. In *ECCV*.

[42] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. 2017. Unsupervised Learning of Depth and Ego-Motion from Video. In *CVPR*.

[43] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. 2020. Learning Monocular Visual Odometry via Self-Supervised Long-Term Modeling. In *ECCV*.