arXiv:2003.07289v4 [cs.CV] 14 Jan 2021

VMLoc: Variational Fusion For Learning-Based Multimodal Camera Localization

Kaichen Zhou, Changhao Chen*, Bing Wang,
Muhamad Risqi U. Saputra, Niki Trigoni, Andrew Markham

Department of Computer Science, University of Oxford
{rui.zhou, changhao.chen, bing.wang, muhamad.saputra, niki.trigoni, andrew.markham}@cs.ox.ac.uk

Abstract

Recent learning-based approaches have achieved impressive results in the field of single-shot camera localization. However, how best to fuse multiple modalities (e.g., image and depth) and to deal with degraded or missing input are less well studied. In particular, we note that previous approaches towards deep fusion do not perform significantly better than models employing a single modality. We conjecture that this is because of the naive approaches to feature space fusion through summation or concatenation which do not take into account the different strengths of each modality. To address this, we propose an end-to-end framework, termed VMLoc, to fuse different sensor inputs into a common latent space through a variational Product-of-Experts (PoE) followed by attention-based fusion. Unlike previous multimodal variational works directly adapting the objective function of vanilla variational auto-encoder, we show how camera localization can be accurately estimated through an unbiased objective function based on importance weighting. Our model is extensively evaluated on RGB-D datasets and the results prove the efficacy of our model. The source code is available at https://github.com/Zalex97/VMLoc.

Introduction

Visual localization is of great importance to many intelligent systems, e.g. autonomous vehicles, delivery drones, and virtual reality (VR) devices. Recently, deep learning has shown its strengths in learning camera pose regression from raw images in an end-to-end manner. However, a single modality solution is normally confronted with issues such as environmental dynamics, changes in lighting conditions, and extreme weather, when it is deployed in complex and ever-changing real-world environments. Meanwhile, those intelligent systems are generally equipped with a combination of sensors (e.g. RGB cameras, depth sensors, and LIDAR) which can be exploited to improve the robustness of the systems. Nevertheless, most studies concentrating on sensor fusion (Liang et al. 2019; Bijelic et al. 2019) directly concatenates different feature vectors together without studying the information contained in different feature vectors. Effectively exploiting different sensor modalities and studying their complementary features will contribute to a more accurate and robust localization system.

*Changhao Chen is the corresponding author.
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

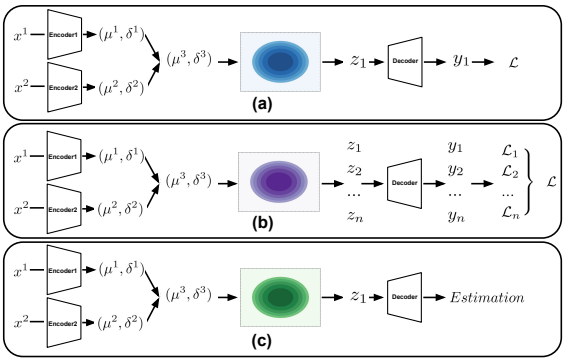

Figure 1: (a) Training process of MVAE. (b) Training process of VMLoc. (c) Inference process of VMLoc.

Compared with learning-based localization algorithms using only RGB images, e.g. PoseNet (Kendall, Grimes, and Cipolla 2015b), Bayesian PoseNet (Kendall and Cipolla 2016), Hourglass Network (Melekhov et al. 2017), CNN+LSTM localization neural network (Walch et al. 2017), PoseNet17 (Kendall and Cipolla 2017), etc., multimodal localization is far less to be explored. VidLoc (Clark et al. 2017a) simply concatenates RGB image and depth map, and processes them together with Convolutional Neural Networks (CNN) for camera localization. However, this direct concatenation on the raw data level won't take advantage of the complementary properties of different modalities. MVAE (Wu and Goodman 2018) proposes to use variational inference to learn an invariant space from multimodal fusion which achieved impressive results in several tasks. This is done by directly optimizing the evidence lower bound (ELBO) whose representation is simplified (Burda, Grosse, and Salakhutdinov 2015). However, MVAE is hard to scale to real-world environments with high-dimensional raw images and rich scene information (e.g. for visual localization problem) as the objective function of MVAE cannot give a tighter estimation of ELBO (Burda, Grosse, and Salakhutdinov 2015).

In order to address the aforementioned problems, we propose VMLoc (**V**ariational Fusion For **M**ultimodal Camera **Loc**alization), a novel framework to learn multimodal 6-DoF camera localization, which learns a joint latent repre-

sentation from a pair of sensor modalities (as depicted in Figure 1 (b) and (c)). Our intuition is that there should exist a common space from multiple modalities that is useful and suitable for solving the task at hand. By using Product-of-Experts (PoE), we combine the individual latent spaces of each modality while at the same time enforcing each modality to concentrate on the specific property that is more useful for the task. We then propose unbiased objective function based on importance weighting to provide the framework with a tighter estimation of ELBO. The main contributions of this work can be summarized as follow:

- We introduce VMLoc, a novel deep neural network framework to combine a pair of sensor modalities, e.g. vision and depth/lidar, for camera localization problem.
- We propose a PoE fusion module to learn the common latent space of different modalities by using an unbiased objective function based on importance weighting.
- Extensive experiments on indoor and outdoor scenarios and systematic research into the robustness and ablation demonstrate the effectiveness of our proposed framework.

Related Work

Sensor Fusion and Multimodal learning

Because of the complementary properties of different sensors, an effective and suitable fusion strategy plays a vital role in learning from multiple sensor modalities in diverse fields, e.g., (Misra et al. 2016), (Valada, Mohan, and Burgard 2019), and (Mees, Eitel, and Burgard 2016).

However, to date, limited research has considered sensor fusion in the context of visual localization, e.g., (Biswas and Veloso 2013) and (Chen et al. 2019). Nevertheless, existing frameworks have very limited performances and they could not provide a better result than state-of-the-art approaches based on monomodal. Learning a joint representation and studying the respective contribution from different modalities are also the focus of multi-modal learning. Among them, some works propose to learn an explicit joint distribution of all modalities, e.g., Joint Multi-modal Variational Autoencoder (JMVAE) (Suzuki, Nakayama, and Matsuo 2016) and PoE (Wu and Goodman 2018). However, both JMVAE and PoE are only tested in simple, simulated datasets, such as MNIST (LeCun et al. 1998) or Fashion MNIST (Xiao, Rasul, and Vollgraf 2017). Other works propose to learn the individual subspace and to achieve cross inference among modalities, e.g., mixture-of-experts (MoE) multimodal variational autoencoder (MMVAE) (Shi et al. 2019) and symbol-concept association network (SCAN) (Higgins et al. 2017b). Compared with them, our work introduces importance weighting strategy into the multimodal variational model to improve its modelling capacity and to reduce the training variance. Meanwhile, we incorporate the geometric loss into the inference process to encourage useful features for pose estimation.

Camera Localization

Camera localization methods can be categorized into the conventional, structure-based models (Sattler et al. 2015; Sattler, Leibe, and Kobbelt 2016; Cavallari et al. 2019; Brachmann et al. 2017; Brachmann and Rother 2018) and deep learning-based models (Kendall, Grimes, and Cipolla 2015a; Walch et al. 2017; Ding et al. 2019).

Conventional Approaches Conventional, structure-based camera localization typically employs Perspective-n-Point (PnP) algorithm (Hartley and Zisserman 2003) applied on 2D-to-3D correspondences. The algorithm consists of finding the correspondences between image features (2D) and world space points (3D) and generating multiple camera pose hypotheses based on the correspondences. The outliers rejection method is then used to cull the multiple hypotheses into a single, robust estimation. (Williams, Klein, and Reid 2011) employed randomized lists classifier to recognize correspondences and use them along with RANSAC to determine the camera pose. (Sattler et al. 2015) finds locally unique 2D-3D matches and employed them to generate reliable localization based on image retrieval techniques. (Sattler, Leibe, and Kobbelt 2016) emphasized that it is important to fuse 2D-to-3D and 3D-to-2D search to obtain accurate localization. However, structure-based models usually require large memory and space to save the 3D model and the descriptors (Sattler et al. 2015). This leads to the development of efficient "straight-to-pose" methods based on a machine learning algorithm.

Deep Learning-based Approaches Instead of basing on 3D-geometry theory to tackle the problem (Chen et al. 2011; Guzman-Rivera et al. 2014; Zeisl, Sattler, and Pollefeys 2015), deep learning models directly learn useful features from raw data (Radwan, Valada, and Burgard 2018; Walch et al. 2017) to regress 6-DoF pose, e.g., PoseNet (Kendall, Grimes, and Cipolla 2015b), PoseNet++ (Melekhov et al. 2017). Further works propose to implement new constraint or new neural network structure to improve the performance, e.g., (Brachman and Schmolze 1985), (Kendall and Cipolla 2017), (Clark et al. 2017b) and (Walch et al. 2017). To date, deep localization has largely considered a single modality as input. However, our work proposes a fusion framework that can effectively use the data from several different sensors to solve the camera localization problem.

Methodology

Problem Formulation

This work is aimed at exploiting multimodal data to achieve more robust and accurate pose estimation $\mathbf{y} = (\mathbf{p}, \mathbf{q})$, which consists of a location vector $\mathbf{p} \in \mathbb{R}^3$ and a quaternion based orientation vector $\mathbf{q} \in \mathbb{R}^4$. The multimodal data are the different observations of an identical scene, but complementary to each other. For example, RGB images contain the appearance and the semantic information of the scene, while depth maps or point cloud data capture the scene structure. Intuitively, these sensor modalities reflect a spatial sense of the scene, and hence a common feature space that is useful for solving the task at hand should exist.

Given two sensor modalities $\mathbf{x^1}$ and $\mathbf{x^2}$, we aim to learn their joint latent representation $\mathbf{z}$. As shown in Figure 1, this problem is formulated as a Bayesian inference model, which

is to maximize the posterior probability conditioned on input data:

$$\mathbf{z} = \arg\max_{\mathbf{z}} [p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})]. \tag{1}$$

Based on this intermediate representation $\mathbf{z}$, the target value $\mathbf{y}$ is obtained by

$$\mathbf{y} = \arg\max_{\mathbf{y}} [p(\mathbf{y}|\mathbf{z})]. \tag{2}$$

The problem becomes how to recover the joint distribution of two modalities. Our work mainly considers a pair of modalities, although the proposed method can be naturally extended to the usage of three or more modalities.

VMLoc

We introduce VMLoc framework to tackle this multimodal learning problem. Our proposed method leverages the variational inference models (Kingma and Welling 2013) to find a distribution $q_\phi(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$, approximating the true posterior distribution $p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$ with the aid of the corresponding geometric information. Algorithm 1 demonstrates the detailed algorithmic description of our proposed VMLoc.

Multimodal Variational Inference Unlike conventional Variantional Auto-Encoders (VAE) (Kingma and Welling 2013), VMLoc reconstructs a target value $\mathbf{y}$ in a different domain, i.e. 6-DoF pose in our case, instead of the original domain data. Similar to the cross-modality VAE (Yang et al. 2019), VMLoc produces the target value $\mathbf{y}$ via a joint latent variable $\mathbf{z}$ of two modalities $\mathbf{x^1}$ and $\mathbf{x^2}$, by maximizing the ELBO as follow:

$$\begin{aligned}\log p(\mathbf{y}) &\geq \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}[\log \frac{p_\theta(\mathbf{y}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}] \\ &= \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})}[p_\theta(\mathbf{y}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x^1},\mathbf{x^2})|p(\mathbf{z})) \\ &= \mathrm{ELBO}(\mathbf{y};\mathbf{x^1},\mathbf{x^2})\end{aligned} \tag{3}$$

where $p(\mathbf{y})$ is the distribution of target value, $p(\mathbf{z})$ is the prior distribution of latent space, $q_\phi(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$ is the inference model to approximate the posterior distribution $p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$, and $p_\theta(\mathbf{y}|\mathbf{z})$ is the decoder network.

Inspired by MVAE (Wu and Goodman 2018), the inference problem of $p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$ can be simplified as learning two conditional distributions $p(\mathbf{z}|\mathbf{x^1})$ and $p(\mathbf{z}|\mathbf{x^2})$ separately. This is under the assumption that two modalities are conditionally independent given the latent representation. Based on that, we can apply the Product-of-Experts (PoE) technique to estimate the joint latent distribution $p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2})$ via:

$$p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2}) = \frac{p(\mathbf{z}|\mathbf{x^1})p(\mathbf{z}|\mathbf{x^2})}{p(\mathbf{z})}. \tag{4}$$

Here, PoE works to combine several simple distributions by producing their density functions (Hinton 1999). In our case, it allows each modality to specialize in their specific property to contribute to the final pose estimation, rather than forcing each modality to recover the full-dimensional information to solve the problem.

As the distribution $p(\mathbf{z}|\mathbf{x^1})$ can be approximated with an inference network $q(\mathbf{z}|\mathbf{x^1}) \equiv \tilde{q}(\mathbf{z}|\mathbf{x^1})q(\mathbf{z})$, Equation 4 is further developed as

$$p(\mathbf{z}|\mathbf{x^1}, \mathbf{x^2}) = \tilde{q}(\mathbf{z}|\mathbf{x^1})\tilde{q}(\mathbf{z}|\mathbf{x^2})q(\mathbf{z}). \tag{5}$$

Thus the learning process can be summarized as: it first learns two individual latent distributions $p(\mathbf{z}|\mathbf{x^1})$ and $p(\mathbf{z}|\mathbf{x^2})$; then combines two distributions to obtain the final joint distribution. In practice, we assume that both the prior distribution and the posterior distribution are Gaussian distributions. With the mean vector and the variance matrix, we can sample from the learned distribution to obtain the joint latent representation of the input modalities.

Importance Weighting Strategy However, directly optimizing ELBO as the objective function can only provide a simplified representation. As shown in Figure 1 (a), in the normal variational model, e.g. MVAE, the representation is only re-sampled once from the latent distribution. In this way, the capability of variational inference has not been fully exploited and our latter experiments in Table 3 support this argument. Therefore, we introduce the importance weighting strategy into the framework. As Figure 1 (b) illustrates, instead of only depending on one sample, this strategy resamples from learned distribution for multiple times to approximate the posterior distribution, which allows the network to model more complicated posterior distribution. In doing so, it provides a strictly tighter log-likelihood lower bound (Burda, Grosse, and Salakhutdinov 2015) through:

$$\mathbb{E}[\log \frac{1}{k}\sum_{i=1}^{k} w_i] \geq \mathbb{E}[\log \frac{1}{k-1}\sum_{i=1}^{k-1} w_i], \tag{6}$$

where $w_i = p(\mathbf{y}, \mathbf{z_i})/q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})$; k is the number of samples; $\mathbf{z_i}$ is the i-th point sampled independently from the latent distribution. Thus, the objective function is rewritten as:

$$\begin{aligned}\log p(\mathbf{y}) &= \log \mathbb{E}[w] \\ &\geq \mathbb{E}[\log \frac{1}{k}\sum_{i=1}^{k} w_i] \\ &= \mathrm{ELBO}(\mathbf{y};\mathbf{x^1},\mathbf{x^2}).\end{aligned} \tag{7}$$

It can be noticed from the above analysis that the larger number of samples leads to a tighter log-likelihood lower bound. With this new objective function, our model uses multiple samples instead of one to approximate the posterior, which then improves its capability to learn more complex posterior. Although by increasing the number of samples we can find a tighter bound, it counter-intuitively leads to a higher variance for gradient estimation in the training process (Rainforth et al. 2018). The gradient for ϕ in Equation 7 are calculated via:

$$\begin{aligned}\nabla_\phi L_k = \mathbb{E}_{\varepsilon_{1:k}}[\sum_{i=1}^{k} \frac{w_i}{\sum_j w_j}(\frac{\partial \log w_i}{\partial \mathbf{z_i}}\frac{\partial \mathbf{z_i}}{\partial \phi} \\ - \frac{\partial}{\partial \phi}\log q_\phi(\mathbf{z_i}|\mathbf{x^1},\mathbf{x^2}))].\end{aligned} \tag{8}$$

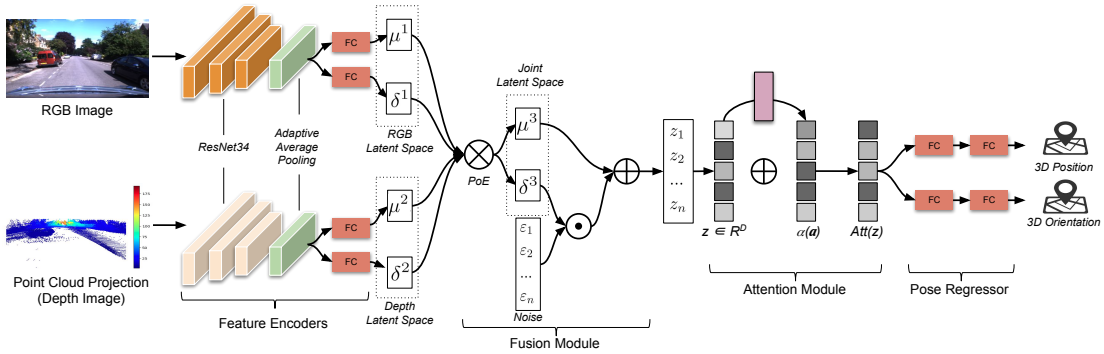

Figure 2: Our framework consists of feature encoders, a fusion module, an attention mechanism module and a pose regressor.

According to (Rainforth et al. 2018), the second item (score function) in Equation 8 can be eliminated, so as to reduce the variance of the gradients. However, they fail to show that eliminating this item is unbiased. To address this concern, (Tucker et al. 2018) eliminates the score function by rewriting Equation 8 as:

$$\nabla_\phi L_k = \mathbb{E}_{\varepsilon_{1:k}}[\sum_{i=1}^{k}(\frac{w_i}{\sum_j w_j})^2 \frac{\partial \log w_i}{\partial \mathbf{z_i}} \frac{\partial \mathbf{z_i}}{\partial \phi}]. \tag{9}$$

Through Equation 9, we mitigate the side effects of increasing k and ensure the gradients of the inference network are unbiased. Instead of directly computing $w_i = p(\mathbf{y}, \mathbf{z_i})/q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})$ as in Equation 9, in practice, we rewrite it as follow:

$$\begin{aligned} w_i &= e^{\log w_i} \\ &= e^{\log p(\mathbf{y}|\mathbf{z_i}) + \log p(\mathbf{z_i}) - \log q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2})} \end{aligned} \tag{10}$$

We further replace the first logarithm likelihood item in Equation 10 with the negative geometric loss which calculates the distance between the predicted pose and the target pose in the pose estimation task.

Geometric Learning To encourage VMLoc to extract useful features for pose estimation, we propose to incorporate geometric information into the optimization objective. We incorporate a learnable geometric loss (Clark et al. 2017b; Brahmbhatt et al. 2018) into the loss function:

$$L_p(\mathbf{y}, \mathbf{y}^*) = ||\mathbf{p} - \mathbf{p}^*||e^{-\beta} + \beta + ||\log \mathbf{q} - \log \mathbf{q}^*||e^{-\gamma} + \gamma \tag{11}$$

where $\mathbf{y}^* = (\mathbf{p}^*, \mathbf{q}^*)$ is the ground truth pose, while β and γ are weights that balance the position and the rotation loss. The β and the γ are optimized during the training process with the initial value β_0 and γ_0. The $\log \boldsymbol{q}$ is the logarithmic form of a unit quaternion $\boldsymbol{q}$, which is defined as:

$$\log \boldsymbol{w} = \begin{cases} \dfrac{\boldsymbol{v}}{||\boldsymbol{v}||} \cos^{-1} \mathbf{u}, & if ||\boldsymbol{v}|| \neq 0 \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{12}$$

where $\boldsymbol{u}$ denotes the real part of a unit quaternion and $\boldsymbol{v}$ is the imaginary part (Wang et al. 2019). In general, the gradient for the inference network ϕ and the decoder network θ (also known as the pose regression network) can be expressed as follow:

$$\nabla_\phi L_k = \mathbb{E}_{\varepsilon_{1:k}}[\sum_{i=1}^{k}(\frac{w_i}{\sum_j w_j})^2 \frac{\partial \log w_i}{\partial \mathbf{z_i}} \frac{\partial \mathbf{z_i}}{\partial \phi}]; \tag{13}$$

$$\nabla_\theta L_k = \mathbb{E}_{\varepsilon_{1:k}}[\sum_{i=1}^{k} \frac{w_i}{\sum_j w_j} \nabla_\theta \log w_i]; \tag{14}$$

$$w_i = e^{-L_p(\mathbf{y}, \mathbf{y}^*) + \lambda(\log p(\mathbf{z_i}) - \log q(\mathbf{z_i}|\mathbf{x^1}, \mathbf{x^2}))}; \tag{15}$$

where λ is the hyperparameter introduced by (Higgins et al. 2017a) to balance the prediction accuracy and the latent space capability. The complete learning algorithm can be found in Algorithm 1.

Algorithm 1 VMLoc algorithm

Require: $\mathbf{x^1}$, $\mathbf{x^2}$ and $\mathbf{y}$
  Initialize parameters $\boldsymbol{\phi_{x^1}}$, $\boldsymbol{\phi_{x^2}}$, $\boldsymbol{\theta}$, β and γ
  for episode=1, N **do**
    Encode $\mathbf{x^1}$ and $\mathbf{x^2}$ with $q_{\boldsymbol{\phi_{x^1}}}(\mathbf{z}|\mathbf{x^1})$ and $q_{\boldsymbol{\phi_{x^2}}}(\mathbf{z}|\mathbf{x^2})$
    Compute the joint distribution via Equation 5
    Sample k points $\mathbf{z_i}$ from joint distribution
    Decode $\mathbf{z_i}$ with $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$
    Update the parameters β and γ with loss function $L_p(\mathbf{y}, \mathbf{y}^*)$ in Equation 11
    Update the parameters $\boldsymbol{\phi_{x^1}}$, $\boldsymbol{\phi_{x^2}}$ with gradient $\nabla_\phi L_k$ in Equation 13
    Update the parameters $\boldsymbol{\theta}$ with gradient $\nabla_\theta L_k$ in Equation 14
  end for

Framework

Now we come to introduce the detailed framework and the training strategy of VMLoc. Figure 2 illustrates the structure of proposed VMLoc, including an RGB image encoder, a depth map encoder, a fusion module, an attention module, and a pose regressor. These two encoders separately encode the RGB images and the depth maps into their own latent space, followed by fusing the multiple individual latent spaces into one joint latent space through PoE. Then,

the latent representation sampled from the joint latent space is re-weighted by a self-attention module. Finally, the re-weighted latent features are taken as the input for the successive pose regressor to predict the 6-DoF camera pose.

Feature Encoders The feature encoders in our framework include an image encoder $q_{\phi_{img}}(\mathbf{z}|\mathbf{x_{img}})$ and a depth encoder $q_{\phi_{dep}}(\mathbf{z}|\mathbf{x_{dep}})$, which separately learn the latent distribution of RGB image and that of depth map. In learning point cloud feature, we transform the lidar data into a image using the cylindrical projection (Chen et al. 2017). CNN has already shown its strengths in the task of visual localization (Brahmbhatt et al. 2018). Among them, the ResNet model has been widely applied in different tasks, e.g., (Brahmbhatt et al. 2018; Wang et al. 2019). Based on these concerns, in our model, we also adopt the ResNet34 structure to construct our RGB image encoder and depth map encoder. To accelerate the convergence speed, the ResNet34 in our model was initialized by the weights of model trained on the Image-Net (He et al. 2016). For both RGB image encoder and depth map encoder, the second last average pooling layer is replaced by the adaptive average pooling layer and is followed by two parallel fully connected layers with the same dimension $D = 1024$, which separately output the mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and the diagonal vector of the variance matrix $\boldsymbol{\sigma} \in \mathbb{R}^D$ of the learned latent distribution.

Fusion Module After we learned the mean vector and the variance matrix for the latent distribution of the RGB image and the depth map, we compute corresponding parameters for the joint distribution via Equation 5. Rather than directly sampling from the joint distribution which isn't differentiable, the reparameterization trick (Kingma and Welling 2013) is applied. Given the mean vector $\boldsymbol{\mu}$ and the variance matrix $\boldsymbol{\sigma}$, we first sample the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then the point of joint latent distribution can be computed as $\mathbf{z} = \boldsymbol{\epsilon}\boldsymbol{\mu} + \boldsymbol{\sigma}$.

Attention Module Considering that certain parts of the features extracted by the model may be useless to the pose regression, we would like to enable our framework to focus on certain representations that are useful to the task. We implement the non-local style self-attention (Wang et al. 2018) in our attention module, which can capture the long-range dependencies and global correlation of the image features (Wang et al. 2019). The computation process of attention module can be summarized as the following two steps. Given a feature vector $\boldsymbol{z} \in \mathbb{R}^D$, we firstly calculate its self-attention as follow:

$$\boldsymbol{a} = \text{Softmax}(\boldsymbol{z}^T \boldsymbol{W}_\theta^T \boldsymbol{W}_\phi \boldsymbol{z}) \boldsymbol{W}_g \boldsymbol{z}, \tag{16}$$

where $\boldsymbol{W}_\theta$, $\boldsymbol{W}_\phi$ and $\boldsymbol{W}_g$ are the learn-able weights. Then, the residual connection will be added to the linear embedding of the self-attention vectors:

$$\text{Att}(\boldsymbol{z}) = \alpha(\boldsymbol{a}) + \boldsymbol{z}, \tag{17}$$

where the $\alpha(\boldsymbol{a}) = \boldsymbol{W}_\alpha \boldsymbol{a}$ and the $\boldsymbol{W}_\alpha$ is a learnable weight which will be optimized during the training process.

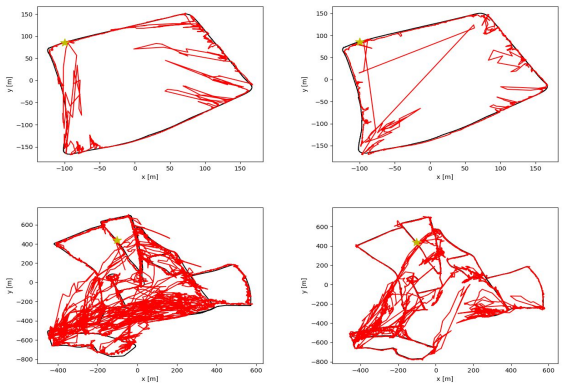

Figure 3: The generated trajectories of LOOP1 (Top) and FULL1 (Bottom) with proposed VMLoc (b) and the baselines MapNet (a). The yellow star denotes the starting point. The ground truth trajectories are shown in black lines, while the red lines are the predicted trajectories.

Pose Regressor Finally, the re-weighted latent vector is taken as input into the pose regressor to estimate 6-DoF pose. The pose regressor consists of two parallel networks sharing the same structure. Each network contains two successive fully connected layers connected by a ReLU activation function. Among them, one network predicts the position vector $\mathbf{p}$, while another network predicts the quaternion based rotation vector $\mathbf{q}$.

Training Strategies In order to force our model to learn from all input modalities and to improve the robustness during the corrupted input conditions, a data augmentation method for multimodal learning is introduced. This data augmentation method can be formulated as follow:

$$(\mathbf{x^1}, \mathbf{x^2}) = \begin{cases} (\mathbf{x^1}, \mathbf{x^2}), & with \quad p_1 \\ (\mathbf{x^1}, None), & with \quad p_2 \\ (None, \mathbf{x^2}), & with \quad p_3, \end{cases} \tag{18}$$

where the sum of p_1, p_2 and p_3 is 1. In our experiments, $p_1 = \frac{3}{5}$, $p_2 = \frac{1}{5}$ and $p_3 = \frac{1}{5}$.

Experiments

Datasets

Our proposed VMLoc framework is evaluated on two common public datasets: 7-Scenes (Shotton et al. 2013) and Oxford RobotCar (Maddern et al. 2017). **7-Scenes Dataset** was collected by a Kinect device, consisting of RGB-D image sequences from seven indoor scenarios. The ground truth was calculated by KinectFusion algorithm. We split the data as training and testing set according to the official instruction. **Oxford RobotCar Dataset** contains multimodal data from car-mounted sensors, e.g., cameras, lidar, and GPS/IMU. We use the same data split of this dataset named LOOP and FULL as in (Brahmbhatt et al. 2018) and (Wang et al. 2019). As there is no depth map in the original Oxford Robot Car dataset, the depth map is obtained by projecting the sparse lidar to the RGB image as shown in Figure 2.

Table 1: The camera localization results from 7-Scenes Dataset. We report the median error of the position and orientation. Hourglass, PoseNet+, MapNet, and Atloc are only based on RGB image (**V**), while VMLoc and VidLoc use RGB and depth images (**V,D**).

Scene	Hourglass(V)	PoseNet+(V)	VidLoc(V,D)	MapNet(V)	AtLoc(V)	VMLoc(V,D)
Chess	$0.15m, 6.17°$	$0.13m, 4.48°$	$0.16m$, NA	$0.08m, 3.25°$	$0.10m, 4.07°$	**0.10m, 3.70°**
Fire	$0.27m, 10.8°$	$0.27m, 11.3°$	$0.19m$, NA	$0.27m, 11.7°$	$0.25m, 11.4°$	**0.25m, 10.5°**
Heads	$0.19m, 11.6°$	$0.17m, 13.0°$	$0.13m$, NA	$0.18m, 13.2°$	$0.16m, 11.8°$	**0.15m, 10.8°**
Office	$0.21m, 8.48°$	$0.19m, 5.55°$	$0.24m$, NA	$0.17m, 5.15°$	$0.17m, 5.34°$	**0.16m, 5.08°**
Pumpkin	$0.25m, 7.01°$	$0.26m, 4.75°$	$0.33m$, NA	$0.22m, 4.02°$	$0.21m, 4.37°$	**0.20m, 4.01°**
Red Kitchen	$0.27m, 10.2°$	$0.23m, 5.35°$	$0.28m$, NA	$0.23m, 4.93°$	$0.23m, 5.42°$	**0.21m, 5.01°**
Stairs	$0.29m, 12.5°$	$0.35m, 12.4°$	$0.24m$, NA	$0.30m, 12.1°$	$0.26m, 10.5°$	**0.24m, 10.0°**
Average	$0.23m, 9.53°$	$0.23m, 8.12°$	$0.23m$, NA	$0.21m, 7.77°$	$0.20m, 7.56°$	**0.19m, 7.01°**

Table 2: The camera localization results for Oxford RobotCar Dataset. We report the mean error of the position and orientation for MapNet, AtLoc, and VMLoc.

Scene	MapNet(V)	AtLoc(V)	VMLoc(V,D)
LOOP1	$8.76m, 3.46°$	$8.61m, 4.58°$	**7.70m, 3.23°**
LOOP2	$9.84m, 3.96°$	$8.86m, 4.67°$	**7.76m, 3.16°**
FULL1	$41.4m, 12.5°$	$29.6m, 12.4°$	**19.5m, 4.32°**
FULL2	$59.3m, 14.8°$	$48.2m, 11.1°$	**35.2m, 8.99°**
Average	$29.8m, 8.68°$	$23.8m, 8.19°$	**17.5m, 4.92°**

Training Details

Our approach is implemented by using PyTorch. The model is trained and tested with an NVIDIA Titan V GPU. During the training process, both RGB images and depth maps are taken as the input, which are rescaled with the shortest side in the length of 256 pixels and normalized into the range of $[-1, 1]$. In the case of VMLoc, the sampling number k is set to be 10. The batch size is set to be 64 and the Adam optimizer is used in the optimization process with the learning rate 5×10^{-5} and the weight decay rate 5×10^{-5}. The training dropout rate is set to be 0.5 and the initialization balance weights are $\beta_0 = -3.0$ and $\gamma_0 = 0.0$. All experiments have been conducted for 5 times to guarantee the reproducibility.

Baseline and Ablation Study

We use the recent state-of-the-art pose regression models as our baseline. Five representative learning-based models, i.e. Hourglass (Melekhov et al. 2017), PoseNet+ (Kendall and Cipolla 2017), VidLoc (Clark et al. 2017a), MapNet (Brahmbhatt et al. 2018), and AtLoc (Wang et al. 2019) are compared with our proposed method. AtLoc depends on single-image to realize accurate pose estimation. MapNet uses a sequence of images for localization which generally performs better than single image localization. We compared with MapNet to show the high accuracy achieved by our single image localization algorithm.

To prove the effectiveness of each module in VMLoc, we compare our model with the image-VMLoc which takes a single RGB image as its input while keeping the importance weighting and the unbiased objective function. To verify the performance of our fusion mechanism, we compare VMLoc with attention-VMLoc which uses attention without PoE and importance weighting, and PoE-VMLoc which uses the MVAE to fuse different distribution without using importance weighting and proposed unbiased objective function. Finally, to prove the robustness of our model, we test the performance of our model under different data degradation conditions.

Table 3: The ablation study of camera localization results with Oxford RobotCar Dataset. We report the mean error of the position and orientation for image-VMLoc, attention-VMLoc, PoE-VMLoc and VMLoc.

Scene	image VMLoc	attention VMLoc	PoE VMLoc	VMLoc
LOOP1	$8.60m, 4.57°$	$9.16m, 4.96°$	$8.57m, 3.98°$	**7.70m, 3.23°**
LOOP2	$8.50m, 3.90°$	$9.78m, 5.66°$	$8.99m, 3.79°$	**7.76m, 3.16°**
FULL1	$30.1m, 10.8°$	$31.2m, 6.04°$	$30.0m, 7.54°$	**19.5m, 4.32°**
FULL2	$48.1m, 9.61°$	$46.5m, 10.1°$	$45.9m, 10.5°$	**35.2m, 8.99°**
Average	$23.9m, 7.22°$	$19.3m, 5.35°$	$18.7m, 5.16°$	**17.5m, 4.92°**

Table 4: The robustness study against corrupted input in Oxford RobotCar Dataset.

Corrupt.	AtLoc(V)	attention VMLoc	PoE VMLoc	VMLoc
No	$48.2m, 11.1°$	$46.1m, 9.50°$	$45.8m, 11.50°$	**35.2m, 8.99°**
		RGB		
$lvl = 1$	$314.3m, 53.1°$	**229.1 m, 46.7°**	$235.5m, 41.5°$	$241.9m, 48.9°$
$lvl = 2$	–	$484.7m, 85.8°$	$478.0m, 85.2°$	**464.7m, 85.5°**
		LIDAR		
$lvl = 1$	–	$47.9m, 9.6°$	$49.7m, 12.1°$	**36.5m, 8.95°**
$lvl = 2$	–	$53.9m, 9.29°$	$87.1m, 36.3°$	**38.7m, 9.70°**

The Performance of VMLoc in Indoor Scenarios

We first evaluate our model on the 7-Scenes dataset to demonstrate its effectiveness in fusing RGB and depth data in indoor scenarios. Table 1 shows the comparison between VMLoc and the competing approaches. The results are reported in the median error (m). VMLoc and VidLoc use both RGB images and depth data, while others are based on RGB images only. Our proposed VMLoc outperforms the other five baseline algorithms in terms of both the position and orientation error. Compared to AtLoc, VMLoc shows a 5.0% improvement upon the position accuracy and a 7.3% improvement in the rotation. In particular, in the Stairs scenario, VMLoc can reduces AtLoc position error from $0.26m$ to $0.24m$ and the rotation error from $10.5°$ to $10.0°$. This because Stairs is a highly texture-repetitive scenario and the structure information captured by the depth map can help to

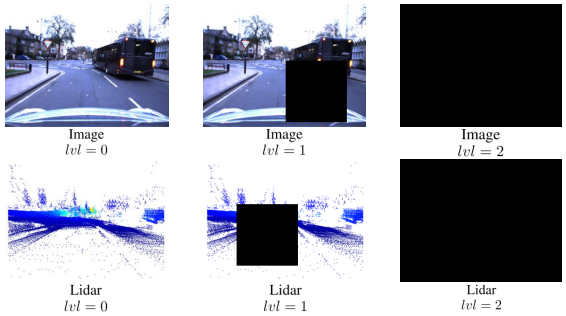

Figure 4: Input images and input projections of lidar with different levels of corruption.

improve the network performance. The improvements made in this case match our expectations and are mainly because of the introduction of depth maps which provide external information for the pose estimation.

Performance of VMLoc in the Driving Scenarios

We further evaluate our models on the Oxford RobotCar outdoor dataset. Table 2 summarizes the results of this experiments. Compared to AtLoc, VMLoc generates 26.5% and 40.0% improvement in position and rotation accuracy respectively. We notice that the overall improvement is due to a large increase in rotation estimation as the projection of point cloud can better capture the structural information of the scene which can largely help estimating more accurate orientation. While the improvements in the position accuracy are less evident than that in rotation accuracy, the main reason may be that the point cloud in the Oxford RobotCar dataset is relatively sparse which can only provide limited complementary geometric information for position estimation. The visualization of MapNet and VMLoc trajectories for LOOP1 and FULL1 are shown in Figure 3. VMLoc yields a closer prediction w.r.t. the ground truth and a smoother trajectory than MapNet.

Ablation Studies of VMLoc

To further verify the performance of the fusion mechanism, the ablation studies are conducted on Oxford Robot Car Dataset. The result is shown in Table 3. With the help of variational learning and importance weighting, image-VMLoc performs better than MapNet and AtLoc as the variational learning learns the latent distribution of input module, which can better deal with the variation of different scenes. By directly concatenating the features of different modalities, attention-VMLoc does not provide a better performance which indicates that the concatenation is not an effective fusion module. By fusing RGB and depth features through PoE, PoE-VMLoc performs better than image-VMLoc, while the slight increase indicates that PoE-VMLoc also does not make full use of the lidar inputs. The best performances in all 4 scenes are achieved by VMLoc which indicates that it is a more effective fusion mechanism than attention-VMLoc and PoE-VMLoc. Moreover, compared with the performance in LOOP1 and LOOP2, a larger increase is yielded in FULL1 and FULL2. Since FULL1 and FULL2 are more complicated than LOOP1 and LOOP2 (Maddern et al. 2017), the model cannot regress precise position only with the help of lidar.

Computational Complexity

The usage of importance weighting may raise concerns about the computational complexity of our algorithm. Nevertheless, during inference, VMLoc requires almost the same GPU time and the same FLOPS as attention-VMLoc and PoE-VMLoc. As the importance weighting is only used in the objective function in the training process, VMLoc has no appreciable difference compared with PoE-VMLoc during inference. With regards to the training process, when the importance weighting is set to 10 and VMLoc, attention-VMLoc and PoE-VMLoc share the same latent feature dimension of each module, $T_{VMLoc} \approx 1.23T_{PoE} \approx 1.02T_{attention}$ where T represents the running time. We can see that the importance weighting does not greatly increase the computational complexity, as the importance sampling mainly influences the pose regressor part of VMLoc which only accounts for less than 5% of the number of parameters.

Robustness Evaluation

To demonstrate the robustness of VMLoc in the case of missing input, we test our models in the condition where input modules are degraded or missing. We adapt the data corruption method from (Chen et al. 2019). In order to describe different data corruption conditions, we define the data corruption level from $lvl = 0, 1, 2$. $lvl = 0$ means that there is no corruption; $lvl = 1$ means that there is mask of dimensions 128×128 pixels overlaying the input; $lvl = 2$ means that the input is totally missing (see Figure 4).

Table 4 shows the performance of different fusion mechanisms under these corruption conditions. It is clear that when either lidar or image input is corrupted, the performances of all fusion mechanisms deteriorate. This indicates that all three learning-based fusion mechanisms make use of all inputs to regress the position and the deterioration of any input would affect their performance. However, in most cases, VMLoc performs better than attention-VMLoc, PoE-VMLoc, and AtLoc, which verifies the effectiveness of our fusion mechanism. Meanwhile, we also notice that the corruption of the image has a larger influence than the corruption of the lidar signal. This denotes that in this case, these fusion mechanisms still mainly relies on RGB image to make the decision which can be due to two reasons. On the one hand, the lidar signal in the Oxford Robot Car dataset is relatively sparse. On the other hand, even though the lidar signal has a wide viewpoint and is invariant to illumination, it is less informative (Tinchev, Nobili, and Fallon 2018; Tinchev, Penate-Sanchez, and Fallon 2019).

Conclusion

Effectively exploiting multimodal data for localization is a challenging problem due to the different characteristics among various sensor modalities. In this paper, we have proposed a novel multimodal localization framework (VMLoc) based on multimodal variational learning. In particular, we designed a new PoE fusion module by employing unbiased

objective function based on importance weighting, which is aimed to learn the common latent space from different modalities. Our experiments have shown that this approach produces more accurate localization compared to existing single image or multimodal learning algorithms, either on benign conditions or when the input data are corrupted.

References

Bijelic, M.; Mannan, F.; Gruber, T.; Ritter, W.; Dietmayer, K.; and Heide, F. 2019. Seeing through fog without seeing fog: Deep sensor fusion in the absence of labeled training data. *arXiv preprint arXiv:1902.08913* .

Biswas, J.; and Veloso, M. 2013. Multi-sensor mobile robot localization for diverse environments. In *Robot Soccer World Cup*, 468–479. Springer.

Brachman, R. J.; and Schmolze, J. G. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9(2): 171–216.

Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; and Rother, C. 2017. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6684–6692.

Brachmann, E.; and Rother, C. 2018. Learning less is more-6d camera localization via 3d surface regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4654–4662.

Brahmbhatt, S.; Gu, J.; Kim, K.; Hays, J.; and Kautz, J. 2018. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2616–2625.

Burda, Y.; Grosse, R.; and Salakhutdinov, R. 2015. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519* .

Cavallari, T.; Golodetz, S.; Lord, N.; Valentin, J.; Prisacariu, V.; Di Stefano, L.; and Torr, P. H. 2019. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *IEEE transactions on pattern analysis and machine intelligence* .

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

Chen, D. M.; Baatz, G.; Köser, K.; Tsai, S. S.; Vedantham, R.; Pylvänäinen, T.; Roimela, K.; Chen, X.; Bach, J.; Pollefeys, M.; et al. 2011. City-scale landmark identification on mobile devices. In *CVPR 2011*, 737–744. IEEE.

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

Clark, R.; Wang, S.; Markham, A.; Trigoni, N.; and Wen, H. 2017a. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6856–6864.

Clark, R.; Wang, S.; Wen, H.; Markham, A.; and Trigoni, N. 2017b. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ding, M.; Wang, Z.; Sun, J.; Shi, J.; and Luo, P. 2019. CamNet: Coarse-to-fine retrieval for camera re-localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2871–2880.

Guzman-Rivera, A.; Kohli, P.; Glocker, B.; Shotton, J.; Sharp, T.; Fitzgibbon, A.; and Izadi, S. 2014. Multi-output learning for camera relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1114–1121.

Hartley, R.; and Zisserman, A. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017a. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Iclr* 2(5): 6.

Higgins, I.; Sonnerat, N.; Matthey, L.; Pal, A.; Burgess, C. P.; Bosnjak, M.; Shanahan, M.; Botvinick, M.; Hassabis, D.; and Lerchner, A. 2017b. Scan: Learning hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389* .

Hinton, G. E. 1999. Products of experts .

Kendall, A.; and Cipolla, R. 2016. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, 4762–4769. IEEE.

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

Kendall, A.; Grimes, M.; and Cipolla, R. 2015a. Convolutional networks for real-time 6-DOF camera relocalization. CoRR abs/1505.07427 (2015).

Kendall, A.; Grimes, M.; and Cipolla, R. 2015b. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, 2938–2946.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Liang, M.; Yang, B.; Chen, Y.; Hu, R.; and Urtasun, R. 2019. Multi-task multi-sensor fusion for 3d object detection. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7345–7353.

Maddern, W.; Pascoe, G.; Linegar, C.; and Newman, P. 2017. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research* 36(1): 3–15.

Mees, O.; Eitel, A.; and Burgard, W. 2016. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 151–156. IEEE.

Melekhov, I.; Ylioinas, J.; Kannala, J.; and Rahtu, E. 2017. Image-based localization using hourglass networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 879–886.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3994–4003.

Radwan, N.; Valada, A.; and Burgard, W. 2018. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters* 3(4): 4407–4414.

Rainforth, T.; Kosiorek, A. R.; Le, T. A.; Maddison, C. J.; Igl, M.; Wood, F.; and Teh, Y. W. 2018. Tighter variational bounds are not necessarily better. *arXiv preprint arXiv:1802.04537* .

Sattler, T.; Havlena, M.; Radenovic, F.; Schindler, K.; and Pollefeys, M. 2015. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2102–2110.

Sattler, T.; Leibe, B.; and Kobbelt, L. 2016. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence* 39(9): 1744–1756.

Shi, Y.; Siddharth, N.; Paige, B.; and Torr, P. 2019. Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models. In *Advances in Neural Information Processing Systems*, 15692–15703.

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

Suzuki, M.; Nakayama, K.; and Matsuo, Y. 2016. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891* .

Tinchev, G.; Nobili, S.; and Fallon, M. 2018. Seeing the wood for the trees: Reliable localization in urban and natural environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8239–8246. IEEE.

Tinchev, G.; Penate-Sanchez, A.; and Fallon, M. 2019. Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU. *IEEE Robotics and Automation Letters* 4(2): 1327–1334.

Tucker, G.; Lawson, D.; Gu, S.; and Maddison, C. J. 2018. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152* .

Valada, A.; Mohan, R.; and Burgard, W. 2019. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision* 1–47.

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

Wang, B.; Chen, C.; Lu, C. X.; Zhao, P.; Trigoni, N.; and Markham, A. 2019. AtLoc: Attention Guided Camera Localization. *arXiv preprint arXiv:1909.03557* .

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Williams, B.; Klein, G.; and Reid, I. 2011. Automatic relocalization and loop closing for real-time monocular SLAM. *IEEE transactions on pattern analysis and machine intelligence* 33(9): 1699–1712.

Wu, M.; and Goodman, N. 2018. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, 5575–5585.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* .

Yang, L.; Li, S.; Lee, D.; and Yao, A. 2019. Aligning Latent Spaces for 3D Hand Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2335–2343.

Zeisl, B.; Sattler, T.; and Pollefeys, M. 2015. Camera pose voting for large-scale image-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2704–2712.