Robotica (2025), **43**, pp. 1660–1674
doi:10.1017/S0263574725000360

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Improve generalization for neural visual-SLAM with Bayes online learning

Jun Liu and Haihang Deng

School of Physics and Electronic Engineering, Northeast Petroleum University, Daqing, Peoples R China
Corresponding author: Haihang Deng; Email: kuokuo980512@163.com

Received: 27 July 2024; **Revised:** 18 February 2025; **Accepted:** 24 February 2025; **First published online:** 31 March 2025

Keywords: deep learning; SLAM; EMA; Bayes online learning

Abstract

Among various deep learning-based SLAM systems, many exhibit low accuracy and inadequate generalization on non-training datasets. The deficiency in generalization ability can result in significant errors within SLAM systems during real-world applications, particularly in environments that diverge markedly from those represented in the training set. This paper presents a methodology to enhance the generalization capabilities of deep learning SLAM systems. It leverages their superior performance in feature extraction and introduces Exponential Moving Average (EMA) and Bayes online learning to improve generalization and localization accuracy in varied scenarios. Experimental validation, utilizing Absolute Trajectory Error (ATE) metrics on the dataset, has been conducted. The results demonstrate that this method effectively reduces errors by 20% on the EuRoC dataset and by 35% on the TUM dataset, respectively.

1. Introduction

Robots have become ubiquitous in various sectors, including manufacturing, agriculture, service industries, and autonomous driving [8]. Robots equipped with advanced sensors can detect motion and efficiently map unknown environments using simultaneous localization and mapping (SLAM) technology [23, 25, 42]. In this study, we utilize a visual SLAM (VSLAM) system based on images from monocular, stereo, and RGB-D cameras [51]. The camera captures successive image frames, providing a rich tapestry of visual information. These data facilitate the extraction of feature points in the environment and enable the estimation of motion and position by computing geometric relationships between adjacent frames. This process relies on the extraction of feature points from images and the subsequent matching of these points to compute relative motion.

In exploring SLAM technology, researchers have employed a variety of strategies to overcome the inherent challenges. In essence, a SLAM system is a state estimation problem. In the construction of traditional SLAM systems, real-time collection of information such as distance, image, and motion is achieved by extracting and processing sensor data. In vision-based SLAM systems, feature point extraction is performed by analyzing successive images using techniques such as ORB [7, 28], SIFT [45], and the optical flow [30, 40, 41] method. This enables motion estimation by comparing feature changes between successive images. Traditional SLAM relies on these hand-designed methods for feature extraction, but these methods perform poorly in environments with large lighting variations and feature-poor environments. Additionally, traditional SLAM systems typically presuppose a static environment, exhibiting constrained capabilities in managing dynamic scenes. With enhanced computational capacities, contemporary SLAM systems increasingly diverge from traditional probability-based estimation methods, favoring advanced techniques such as least squares methods and Bundle Adjustment (BA) to optimize the 3D structure and camera trajectories across the map while reducing reliance on

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

filtering methods like Kalman and particle filtering for managing data uncertainty. Although technological advances have made SLAM systems more stable and reliable, they still face the challenge of lack of robustness in practical applications and may suffer from problems including but not limited to feature point trajectory loss, as well as cumulative positioning drift over time.

Deep learning, a pivotal subfield of machine learning, addresses complex challenges by emulating neural networks akin to those found in the human brain. Recently, the application of deep learning in SLAM systems, exemplified by technologies such as SfM-Learner [54] and CNN-SVO [26], has become increasingly prevalent. VSLAM systems that leverage deep learning exhibit robust generalization capabilities, enabling effective operation in complex environmental scenarios [52]. Deep learning is typically employed across multiple sessions to address issues such as feature point trajectory loss, divergence in optimization algorithms, and the accumulation of drift. Although SLAM algorithms that integrate deep learning mitigate many of these issues, these systems, despite their enhanced robustness, often exhibit less precision than classical systems on standard benchmarks.

Developed in 2021 by Zachary Teed and Jia Deng at Princeton University, the deep learning-based visual SLAM system, DROID-SLAM [41], has garnered widespread acclaim since its unveiling. DROID-SLAM is closer to the DeepV2D [39] and BA-Net [37] architectures. DROID-SLAM is based on an advanced optical flow estimation method, RAFT [40], which mainly improves optical flow computation by using deep learning techniques. DROID-SLAM utilizes the advanced RAFT optical flow estimation method, significantly enhancing optical flow computations through the application of deep learning techniques. The RAFT network comprises a feature coding layer that extracts features pixel-by-pixel, a feature correlation layer producing 4D correlation data for all pixels, and a cyclic updating algorithm utilizing GRUs to iteratively refine the optical flow field by leveraging this 4D correlation information. Leveraging RAFT, DROID-SLAM incorporates an iterative update mechanism that persistently refines the camera pose and scene depth information. Another pivotal innovation in DROID-SLAM is the integration of the Dense Bundle Adjustment (DBA) layer. DROID-SLAM is highly accurate and more robust, with a much lower incidence of catastrophic failures. Although primarily trained on monocular video, DROID-SLAM can also utilize stereo or RGB-D video to enhance its performance during testing phases. While DROID-SLAM is an excellent system, in many deep learning-driven vision SLAM systems, including DROID-SLAM, the model's ability to generalize is affected by its training data, and these systems can suffer from degraded performance when confronted with environments that are significantly different from the training set. The efficacy of the features learned by the model during training is intimately tied to the diversity and representativeness of the training data. If the scenarios or objects in the test environment are not adequately represented in the training set, the model may not be able to understand and interpret the features of these new scenarios properly.

In this research, by enhancing the deep learning-based SLAM system, DROID-SLAM, we present the following contributions aimed at augmenting the system's generalization capabilities:

- We employ the Exponential Moving Average (EMA) technique, an averaging methodology that assigns greater weight to more recent data points. In the realm of deep learning, EMA is utilized to smooth model parameters; in our research, we have integrated EMA into the DROID-SLAM system to enhance both the model's generalization capabilities and the stability of the training process.
- We implement the Bayes update online learning method to integrate it seamlessly into the DROID-SLAM system. Within the deep learning framework, the Bayes update online learning method is primarily employed to perpetually adjust and refine model parameters, thereby allowing the system to adapt dynamically to new information in the data stream.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

2. Related work

Classical SLAM systems have undergone extensive exploration and enhancement since their establishment as a foundational concept; consequently, a comprehensive review is beyond the scope of this study. Instead, we focus on recent advancements in three primary categories: Deep Learning VSLAM, Exponential Moving Average (EMA), and Bayes online learning.

Deep Learning VSLAM—In 2007, Davison et al. pioneered MonoSLAM [11], a groundbreaking method for simultaneous localization and map construction using a monocular camera. This seminal solution established a robust foundation for further research in the field. Concurrently, Klein and Murray introduced a VSLAM solution [22] that integrated a keyframe mechanism, marking another significant advancement. This method distinctly delineated the front-end and back-end of the system, significantly enhancing system accuracy and robustness. Despite the relative limitations in application scenarios and occasional tracking losses, the foundational concepts and key technologies of these early VSLAM solutions continue to influence contemporary SLAM systems significantly. These pioneering ideas have not only galvanized subsequent research but also propelled the expansion of VSLAM technology into a wider array of application domains.

Traditional SLAM systems are categorized into two principal schools of thought: the feature point method [7, 28, 30] and the direct method [13]. The feature point method concentrates on precisely extracting and tracking uniformly distributed corner points, along with their distinctive descriptors, from successive video frames. Owing to their invariant properties, these descriptors effectively facilitate the matching of key points across frames. Subsequently, the system can initially recover the camera's motion orientation and accordingly construct the environmental map's coordinate system. Furthermore, the system meticulously refines the camera pose and the map through an iterative optimization process designed to minimize reprojection errors, thereby enhancing the accuracy of both positioning and map construction. The direct method adopts an alternative method, directly estimating the camera's position and orientation based on photometric errors, thus eliminating the need to extract corner points and their descriptors. This method offers computational efficiency and, as it does not depend on global feature descriptions, faces challenges in characterizing overall image properties and managing closed-loop detection. Particularly, the direct method has yet to discover a perfect solution for eliminating positional drift that accumulates over extended operations, limiting its application in high-precision or long-duration autonomous navigation tasks.

In recent years, the integration of computer vision with deep learning has significantly matured, leading to substantial enhancements in the practical performance of vision-related tasks such as instance classification [48], object detection [2], and behavior recognition [19]—improving accuracy, execution efficiency, and robustness. Consequently, scholars have investigated the incorporation of deep learning methods into SLAM systems, aiming to achieve enhanced accuracy and robustness in environmental perception and localization. Deep learning is commonly applied to various subcomponents within SLAM systems, including deep-learning-based visual odometry [12], closed-loop detection [53], and semantic SLAM [47]. A significant portion of the research concentrates on end-to-end trained SLAM systems [20]; however, these methods do not constitute complete SLAM systems and primarily focus on small-scale reconstruction, processing only two to a dozen frames of data. These systems often lack essential features of contemporary SLAM systems, such as loop closure and global bundle adjustment, thereby limiting their capability for large-scale reconstruction. My research is grounded in the advanced deep learning SLAM system, DROID-SLAM [41], inspired by Dusmanu et al. [14], who developed a neural network within the SfM pipeline to enhance keypoint localization accuracy. The DROID-SLAM system executes feature extraction using RAFT [40], facilitated by a neural network that processes through the DBA layer. DROID-SLAM is highly accurate and represents a significant improvement over prior efforts, demonstrating robustness with a markedly reduced incidence of catastrophic failures.

Although initially trained on monocular video, DROID-SLAM can also utilize stereo or RGB-D video to enhance performance during testing. However, similar to many deep learning-based SLAM systems, DROID-SLAM faces challenges in generalization when encountering environments significantly

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

different from its training set. Numerous studies have advanced modifications based on DROID-SLAM [31], and our research also builds upon this foundational algorithm.

Increase genenoralization capacity—The generalization capability of neural networks in deep learning refers to their adeptness at adapting to novel, previously unseen data. This capability serves as a measure of whether the knowledge acquired by the model during the learning and training phases can be effectively leveraged for new datasets. To enhance the generalization ability of models, researchers have employed techniques such as data augmentation [38], regularization [50], cross-validation [29], early stopping [3], batch normalization [33], and ensemble methods [17]. Our research draws inspiration from the seminal works of Zhaowei Cai et al. and Aodong Li et al.

Zhaowei Cai et al. [6] proposed an innovative technique as an alternative to batch normalization (BN) [4], termed Exponential Moving Average Normalization (EMAN). Unlike traditional batch normalization, where statistics are computed independently for each batch, EMAN leverages the BN statistics from the student model within the teacher model. This technique updates its statistics through an EMA, enhancing generalization by diminishing reliance on individual data batches, thereby rendering the model more stable and efficient in handling diverse datasets. Originally utilized in statistics and signal processing, primarily for smoothing time series data in economic and financial market analysis [18], this concept is now gradually being integrated into deep learning methodologies. In our research, EMA performs smoothing updates within the intermediate layers of the network, thereby enhancing both the model's performance and its generalization capability.

Aodong Li et al. [24] introduced an innovative Bayes online learning framework designed to address the challenges posed by non-static data distributions. This framework integrates concepts such as change-point detection, switching dynamical systems, and Bayes online learning, making it particularly adept at managing the frequent distributional shifts common in online learning environments. Inspired by this work, we considered incorporating Bayes online learning to enhance the online adaptability of the entire SLAM system, enabling it to better navigate new environments.

Bayes online learning is extensively employed across various domains, including financial market forecasting, medical care, autonomous vehicle technology, and other fields. In Bayes online learning, model parameters are treated as random variables, with their distributions encapsulating our uncertainty regarding these parameters. During the learning process, each arrival of new data transforms the previous posterior distribution into a new prior distribution, which is then combined with the new data to form an updated posterior distribution. This iterative process continues, progressively refining the model parameters. Online learning can enhance the SLAM system's ability to generalize to previously unseen environments. Through continuous learning and integration of new data, the model can persistently optimize and adjust its internal parameters, thereby effectively managing diverse scenarios.

3. Method

The research builds upon the deep learning-based SLAM system, DROID-SLAM [41], with the objective of enhancing the model's generalization to unknown environments using a single training set. This is achieved through the integration of EMA and Bayes online updating.

In selecting techniques to enhance the model's generalization capability, we adopt the EMA and Bayes online learning methods, primarily due to their advantages in online adaptability [6]. In contrast, alternative methods typically necessitate additional modifications or extensions to the model during the training phase. For instance, data augmentation techniques [34] enhance data diversity by expanding the training dataset, while transfer learning [1] necessitates additional training on top of pre-existing models to adapt to new tasks or environments. These methods generally incur additional computational resources and extended training time.

In contrast to these methods, our method enables real-time model updates during the inference phase. By employing EMA to smooth changes in historical data and dynamically adjusting the model's confidence through Bayes methods, we effectively handle non-smoothness and noise in the data.

Moreover, this method incurs minimal computational overhead when processing large-scale real-time data. In our research, we will investigate the development of a more efficient SLAM system and explore

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

the integration of EMA with Bayes online updating. Through these enhancements, we aim to enhance the system's generalization capability and mitigate the accuracy degradation of SLAM systems when operating in unknown environments.

3.1. Framework

The work reconstructs scenes by analyzing a sequence of ordered images $\{I_t\}_{t=0}^{N}$. Specifically, for each image t, we maintain two unknown state variables: camera pose $G_t \in SE(3)$ and inverse depth $d_t \in \mathbb{R}_{+}^{H\times W}$. The two-state variable sets, the pose set$\{G_t\}_{t=0}^{N}$and the inverse depth set $\{d_t\}_{t=0}^{N}$, are iteratively updated as each new frame is processed. EMA is integrated into the iterative network, performing smoothing updates at an intermediate layer. For the remainder of this paper, it is important to note that all references to depth pertain to the inverse depth parameterization. Utilizing inverse depth enhances the sensitivity and accuracy of detecting changes in scene depth while preserving the algorithm's efficiency.

By constructing a frame-graph (ν, ε), which represents the covariance between frames, the edges $(i, j) \in \varepsilon$ indicate that images I_i and I_j contain overlapping regions and share points. The frame graph is dynamically constructed during both the training and inference phases. Following each update of pose or depth information, the system undergoes a visibility re-evaluation process to update the graphical representation of the frames dynamically. When the camera's field of view revisits a previously mapped region to achieve loop closure (i.e., identifying and correcting cyclic portions of the path to enhance the map's consistency and accuracy), we introduce long-distance connecting edges into the frame graph. These edges span spatial regions that might otherwise be considered discontinuous, thereby establishing comprehensive spatial connections. Ultimately, to enhance the algorithm's broad applicability and online adaptability, we incorporate an online learning method grounded in Bayes theory into the overall algorithm. This method enables the algorithm to continuously update its internal model or parameters based on newly acquired data while processing the data stream in real time, thus achieving a sensitive response to environmental changes and continuous optimization. Through the Bayes framework, we can rationally incorporate prior knowledge and dynamically adjust the model's posterior distribution based on observed data, thereby enhancing the algorithm's adaptability to new situations and unknown changes while maintaining stability.

3.2. Feature extraction and correlation

The feature extraction component of DROID-SLAM parallels the RAFT framework, which reconstructs scenes by parsing a sequence of ordered images $\{I_t\}_{t=0}^{N}$ newly introduced into the system. It leverages RAFT's dual-network strategy: constructing both a correlation volume network and a context network. The feature network comprises six residual blocks and three downsampling layers, effectively reducing the resolution of each input image to one-eighth of the original, thus generating a high-level feature map.

Constructing the correlation pyramid is a crucial step in enhancing performance. For each pair of pixel points $(i, j) \in \varepsilon$ in an image pair, we compute a four-dimensional correlation volume, as illustrated in Eq. (1). This is accomplished by calculating the dot product of the feature vectors corresponding to feature points $g_\theta(I_i)$ and $g_\theta(I_j)$ in the two images. This process captures not only the similarity between pixels but also preserves the spatial location information. A four-layer correlation pyramid is constructed by average pooling the last two dimensions of the correlation volume.

$$C^{ij}_{u_1v_1u_2v_2} = (g_\theta(I_i)_{u_1v_1}) \cdot (g_\theta(I_j)_{u_2v_2}) \tag{1}$$

Upon completing the construction of the correlation pyramid, the next step involves performing a correlation lookup. Utilizing a lookup operator to index the correlation volume within a grid of radius r, the operator retrieves the corresponding correlation information tensor at various resolutions, $L_r : \mathbb{R}^{H\times W} \times \mathbb{R}^{H\times W\times 2} \to \mathbb{R}^{H\times W\times (r+1)^2}$.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

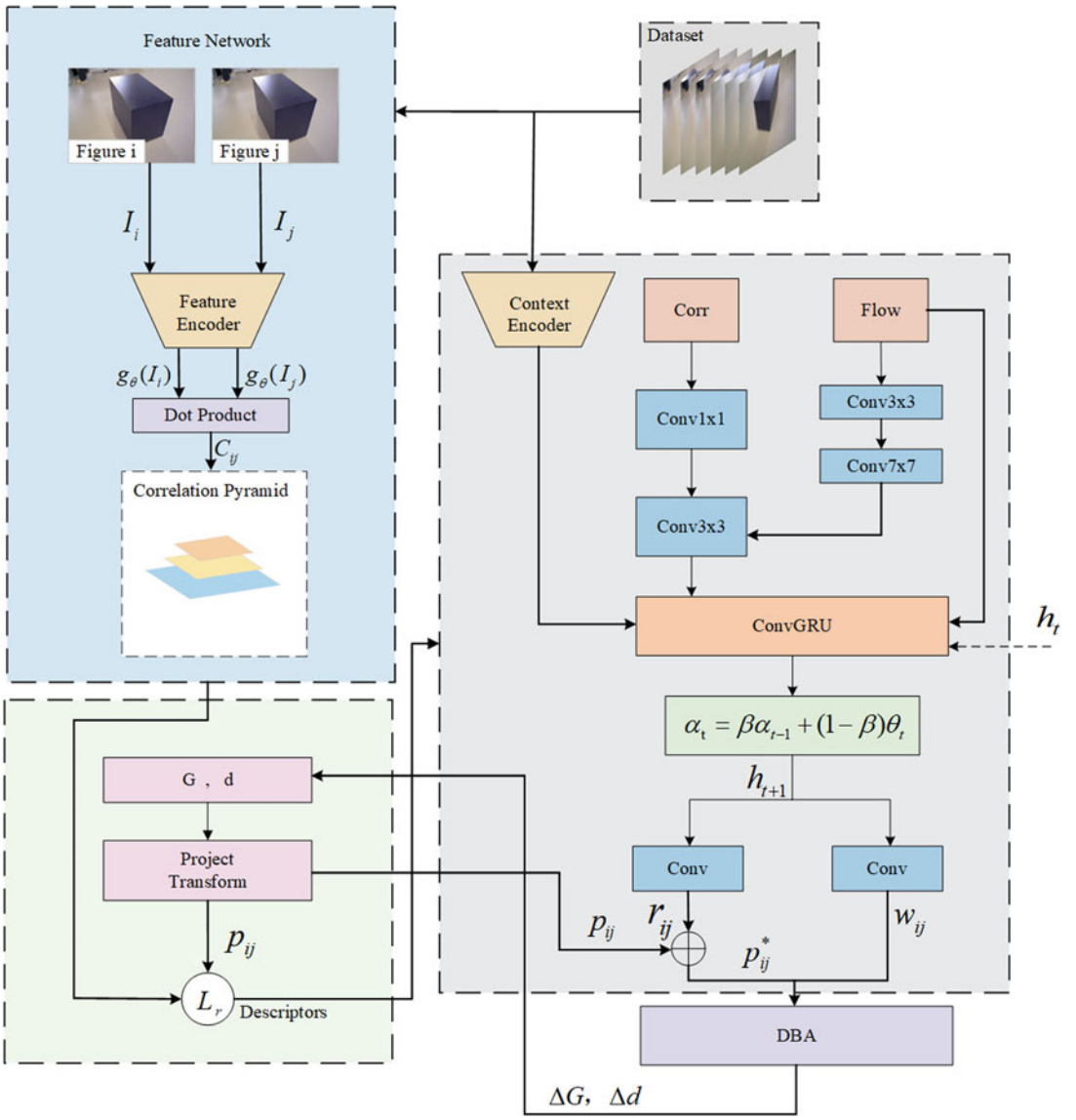


Figure 1. *The image shows how the input image is optimized from feature extraction, feature finding, and finally through the fusion of the EMA's update operator to the DBA layer all in one process.*

3.3. Updating operators and training

Integrating Exponential Moving Average (EMA) into the core component of DROID-SLAM, the update operator, facilitates smoother updates within the system. At the heart of the operator lies a 3×3 Convolutional Gated Recurrent Unit (ConvGRU) with a hidden state h. The operator updates the hidden state and generates a bit position $\Delta\xi^{(k)}$ along with an inverse depth update $\Delta d^{(k)}$, subsequently applying these updates to the current inverse depth and bit position estimations, as illustrated in Eqs. (2) and (3).

$$G^{(k+1)} = \mathrm{Exp}(\Delta\xi^{(k)}) \circ G^{(k)} \tag{2}$$

$$d^{(k+1)} = \Delta d^{(k)} + d^{(k)} \tag{3}$$

The architectural diagram of the update operator for the fusion EMA is shown in Figure 1.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

3.3.1 Updating operators for fusion of EMA

Correspondence is established prior to initiating the update process. For the pixel coordinate grid defined for the i-th frame image, $p_i \in \mathbb{R}^{H\times W\times 2}$, the correspondence field p_{ij} is calculated.

$$p_{ij} = \Lambda_c(G_{ij} \circ \Lambda_c^{-1}(p_i, d_i)) \quad p_{ij} \in \mathbb{R}^{H\times W\times 2} \tag{4}$$

$$G_{ij} = G_j \circ G_i^{-1} \tag{5}$$

Eq. (4) for Λ_c is the mapping of 3D points onto the image, and the corresponding position of each point in frame i in frame j is calculated by using the known positional transformations and the camera model. Eq. (5) is used to calculate the relative pose between two frames, which is obtained by the combination of the pose of frame j with the inverse of the pose of frame i.

The network retrieves correlation features from the correlation volume $C_i j$ for each edge $(i, j) \in \varepsilon$ by indexing the correlation volume with the corresponding field and computes them based on the optical flow differences induced by camera motion. The residuals from the previous Bundle Adjustment (BA) solution are linked to the optical flow field, allowing the network to receive feedback from previous iterations. The correlation features offer visual similarity information, while the optical flow provides smooth motion information, thereby enhancing robustness. The correlation and optical flow features are injected into the GRU after processing through two convolutional layers, with contextual features introduced via elementwise addition. The GRU operates locally, while the global context is extracted by averaging the hidden states across the spatial dimensions of the image, serving as an additional input to the GRU.

$$\alpha_t = \beta\alpha_{t-1} + (1-\beta)\theta_t \tag{6}$$

Concurrently, this work introduces the EMA method within the network's GRU component. The EMA method is implemented to address the jittering of model weights at the actual optimal point during the last n steps. By averaging these steps, the model becomes more robust. During the optimization process, θ_t represents the model weight at the current moment, α_t is the shadow weight averaged over the previous t moments, and β is the weighted weight value, as depicted in Eq. (6). We maintain the shadow weights throughout the training process; these weights are not involved in training but are utilized during the inference phase to enhance the system's generalization capability. EMA mitigates error amplification through the weighted averaging of historical data, thereby reducing oscillations during the training phase, particularly in the initial stages of training. As the model adapts to new data distributions, EMA effectively prevents overfitting to anomalous or noisy data points. EMA is employed to smooth the model parameters, including pose estimation and feature confidence, which are vulnerable to noise and dynamic environments, and to update these parameters with each model iteration.

Finally, in generating the updated hidden state h_{k+1}, the modified flow field r_{ij} and the confidence map w_{ij} of the dense flow field are predicted. The modified flow field is used to correct the dense counterpart field shown in Eq. (7).

$$p_{ij}^* = r_{ij} + p_{ij} \tag{7}$$

The pixel-level damping factor and the 8×8 mask used for the inverse depth estimation during up-sampling are predicted by aggregating features that share the same source view i. The inverse depth estimation is performed similarly to the inverse depth estimation during up-sampling.

$$E(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \left\| \mathbf{p}_{ij}^* - \Lambda_c\left(\mathbf{G}_{ij}' \circ \Lambda_c^{-1}(\mathbf{p}_i, \mathbf{d}_i')\right)\right\|_{\Sigma_{ij}}^2 \quad \Sigma_{ij} = \operatorname{diag} \mathbf{w}_{ij} \tag{8}$$

The corrected p_{ij}^*, weighted by w_{ij}, is subsequently fed into the Dense Bundle Adjustment (DBA) layer. A set of flow revisions is transformed into a series of pose and pixel depth updates by optimizing the cost function, as delineated in Eq. (8). The DBA layer is integrated into the computational map, and backpropagation is conducted through this layer during training.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

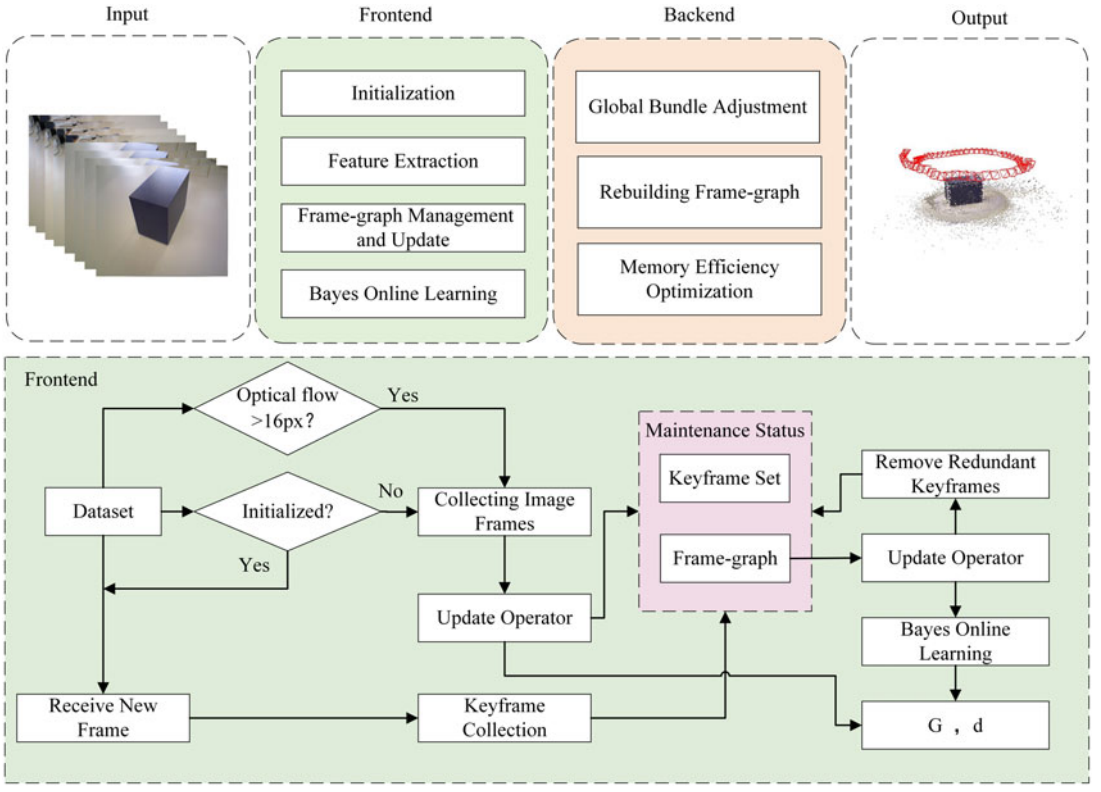


Figure 2. *The figure prominently displays the key components of the SLAM system at the top, with the green area at the bottom illustrating the principal optimization process of the frontend, including the integration of Bayes online learning. The pink area delineates the maintenance status throughout the update process.*

3.3.2 Training

The training process, conducted using PyTorch akin to DROID-SLAM, addresses the canonical degrees of freedom issue in monocular setups by fixing the first two poses of the training sequence. This method enhances both the conditional and gradient stability of the linear system. The training video is dynamically generated to maintain an average optical flow between 8px and 96px across neighboring frames. Supervision is achieved through a dual loss system, combining pose loss and optical flow loss, which serves to enhance both the stability and downstream performance of the model.

3.4. SLAM system

The operational process of our SLAM system bears similarity to DROID-SLAM [41]. However, our innovation incorporates Bayes online learning into the frontend of the SLAM system, as depicted in Figure 2. This enhancement significantly boosts the system's generalization capabilities in unfamiliar scenarios.

3.4.1 Bayes online learning

Bayes online learning can be effectively applied in this context to adaptively update model parameters based on incoming data streams. Here's how it can be integrated:

Bayes online learning maintains a posterior distribution over model parameters, which is updated sequentially as new data arrives. Given the model weight at time t, θ_t, and the shadow weight averaged

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

Algorithm 1. Bayes Online Learning for SLAM

Require: D_t: New data at time t
Require: α_{t-1}: Shadow weight at time $t-1$
Require: w_{target}: Target weight based on new data
Require: γ: Learning rate

Initialization:
$\alpha_0 \leftarrow 0$ {Initialize shadow weight}
for each time step $t = 1$ to T **do**
  Update Shadow Weight:
  $\alpha_t \leftarrow \gamma \cdot (w_{target} - \alpha_{t-1}) + \alpha_{t-1}$ {Update shadow weight based on target and previous weight}
  Compute Cost Function:
  $E(\mathbf{G}', \mathbf{d}', \theta_t) \leftarrow \sum_{(i,j)\in\mathcal{E}} \left|\mathbf{p}_{ij}^* - \Lambda_c\left(\mathbf{G}_{ij}' \circ \Lambda_c^{-1}(\mathbf{p}_{ij}, \mathbf{d}_i')\right)\right|^{\Sigma_{ij}^2}$
  Add Regularization Term:
  $E(\mathbf{G}', \mathbf{d}', \theta_t) \leftarrow E(\mathbf{G}', \mathbf{d}', \theta_t) + \lambda \cdot R(\theta_t|D_t)$
  Calculate Regularization:
  $R(\theta_t|D_t) \leftarrow -\log P(\theta_t|D_t)$ {Regularization based on posterior distribution}
  Optimize SLAM Parameters:
  Update SLAM parameters θ_t to minimize $E(\mathbf{G}', \mathbf{d}', \theta_t)$
  Perform Inference: Use updated α_t for inference and parameter adaptation without retraining the model.
end for

over previous moments, α_t, Bayes inference provides a principled way to update these weights. The shadow weight α_t can be updated recursively using the following equation:

$$\alpha_t = \gamma \cdot (w_{target} - \alpha_{t-1}) + \alpha_{t-1} \tag{9}$$

Where γ is the learning rate of the update processing. This allows the system to track the optimal point more robustly by averaging recent updates, thereby reducing the impact of noise or rapid fluctuations in the model parameters.

During the inference phase, the shadow weights α_t are utilized to enhance the system's generalization capability without directly affecting the training process. This integration enhances the model's stability and adaptability over time, which is crucial for tasks where continuous learning and adaptation to new data are required.

Incorporating Bayes online learning into the optimization process involves adapting the cost function $E(\mathbf{G}', \mathbf{d}')$ from Eq. (8) to include the posterior distribution over model parameters. The updated cost function can be formulated as:

$$E(\mathbf{G}', \mathbf{d}', \theta_t) = \sum_{(i,j)\in\mathcal{E}} \left|\mathbf{p}ij^* - \Lambda_c\left(\mathbf{G}ij' \circ \Lambda_c^{-1}(\mathbf{p}_{ij}, \mathbf{d}i')\right)\right|^{\Sigma ij^2} + \lambda \cdot R(\theta_t|D_t) \tag{10}$$

where $R(\theta_t \mid D_t)$ is the regularization term based on the posterior distribution $P(\theta_t \mid D_t)$ of parameters θ_t given the current data D_t.

$$R(\theta_t \mid D_t) = -\log P(\theta_t \mid D_t) \tag{11}$$

This defines $R(\theta_t \mid D_t)$ as the negative logarithm of the posterior probability $P(\theta_t \mid D_t)$. This regularization term ensures that the model parameters remain within a reasonable range and prevents overfitting, which is especially important in online learning scenarios where new data continuously updates the model's parameters.

Thus, by integrating Bayes online learning with the existing methods described, the model not only benefits from enhanced stability through shadow weights α_t but also adapts more effectively to new

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

data while maintaining the robustness and generalization capabilities. This method supports continuous improvement and adaptation in dynamic environments, making it suitable for applications such as real-time data processing and adaptive learning systems.

4. Experiments

The experiments are designed to assess the enhancement in generalization ability of our algorithm by utilizing a variety of datasets. The experimental framework is methodically structured into two core components: comparative analysis and ablation study. The principal evaluation metric employed in this study is the Absolute Trajectory Error (ATE) [36] for comparative analysis. Consistent with our focus on enhancing generalization capability, the training strategy closely mirrors that of DROID-SLAM, utilizing monocular video sequences derived from the synthetic Tartan Air dataset. Specific implementation details are deliberately omitted for brevity.

4.1. Comparative analysis

The comparative experiments are assessed using the EuRoC and TUM-RGBD datasets. These datasets have established themselves as standard benchmarks for evaluating SLAM algorithms, owing to their extensive adoption within the research community. Leveraging these datasets enables a more comprehensive comparison of the performance across various algorithms, thereby ensuring the fairness, consistency, and reproducibility of the experimental results.

EuRoC [5] dataset encompassing both monocular and binocular configurations, serves as a classical benchmark for SLAM systems, thereby enhancing the reliability of evaluations. This dataset comprises multiple sequences of data collected by a micro air vehicle equipped with a camera and an IMU. We employ the EuRoC dataset to assess the performance of both monocular and binocular systems, as detailed in Table I.

Many deep learning-based SLAM algorithms [9, 39, 43] exhibit weak generalization when tested on the EuRoC dataset, making it a more precise benchmark for evaluating generalization capabilities. Within a monocular setting, our primary focus is to compare the results yielded by DROID-SLAM [41] on the dataset. The method performs excellently on the V202 dataset, with a 37% reduction in error rate compared to DROID-SLAM, and a 20% reduction in overall error rate across all datasets. Although our method does not outperform DROID-SLAM on the MH05 and V102 datasets for visual odometry, this discrepancy is primarily attributable to rapid camera motion and abrupt illumination changes within these datasets, which lead to low correlation between adjacent frames and complicated feature matching, thereby impacting the accuracy of visual odometry. However, our method demonstrates a significant improvement in visual odometry performance, with an overall error rate reduction of 31%. Despite mediocre performance on individual datasets, the substantial overall reduction in error demonstrates a significant enhancement in the generalization ability of our method.

We benchmark the performance of our method against data from DROID-SLAM and other classical algorithms. When comparing DeepV2D trained on the TartanAir [44] dataset to versions trained on NYUv2 [35] and ScanNet [10], the error rate is markedly higher, and both the generalization ability and robustness are inferior to our algorithm, which exhibits fewer issues with drift.

TUM-RGBD [36] The TUM-RGBD dataset is a widely utilized resource within robotics and computer vision research, featuring a diverse array of complex scenes. It provides timestamped RGB and depth image sequences and ground truth data for camera poses. We present a comparison against other deep learning SLAM systems in Table II, highlighting our algorithm's enhanced generalization capabilities. We also cross-reference data from DROID-SLAM [41] to demonstrate the effectiveness of our algorithm.

For the nine sequences successfully tracked within the TUM-RGBD dataset, the teddy sequences have an ATE that is 58% lower than that of DROID-SLAM. While some sequences show less optimization, the overall ATE is 35% lower compared to DROID-SLAM. Concurrently, for the performance across these nine sequences, our ATE is 90% lower than that achieved by DeepFactor [9].

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

Table I. *In the evaluation of monocular SLAM on the EuRoC datasets, the symbol † represents visual odometry methods. The underlined data points represent the primary data we are comparing against.*

		MH01	**MH02**	**MH03**	**MH04**	**MH05**	**V101**	**V102**	**V103**	**V201**	**V202**	**V203**	**Avg**
Classical	ORB-SLAM [27]	0.071	0.067	0.071	0.082	0.060	0.015	0.020	X	0.021	0.018	X	-
	DSO [15]†	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	SVO [16]†	0.100	0.120	0.410	0.430	0.300	0.070	0.210	X	0.110	0.110	1.080	-
	DSM [55]	0.039	0.036	0.055	0.057	0.067	0.095	0.059	0.076	0.056	0.057	0.784	0.126
	ORB-SLAM3 [7]	0.016	0.027	0.028	0.138	0.072	0.033	0.015	0.033	0.023	0.029	X	-
Deep/Hyb.	DeepFactors [9]	1.587	1.479	3.139	5.331	4.002	1.520	0.679	0.900	0.876	1.905	1.021	2.040
	DeepV2D [39]†	0.739	1.144	0.752	1.492	1.567	0.981	0.801	1.570	0.290	2.202	2.743	1.298
	DeepV2D (Tartan Air)†	1.614	1.492	1.635	1.775	1.013	0.717	0.695	1.483	0.839	1.052	0.591	1.173
	TartanVO[1] [43]†	0.639	0.325	0.550	1.153	1.021	0.447	0.389	0.622	0.433	0.749	1.152	0.680
	D3VO+DSO [49]†	-	-	0.080	-	0.090		-	0.110	-	0.050	0.190	-
	DROID-SLAM [41]†	0.163	0.121	0.242	0.399	0.270	0.103	0.165	0.158	0.102	0.115	0.204	0.186
	DROID-SLAM	0.013	0.014	0.022	0.043	0.043	0.037	0.012	0.020	0.017	0.013	0.014	0.022
	Ours(odometry only)†	**0.157**	**0.075**	**0.098**	**0.286**	**0.320**	**0.060**	**0.167**	**0.052**	**0.038**	**0.087**	**0.119**	**0.132**
	Ours	**0.010**	**0.009**	**0.016**	**0.039**	**0.032**	**0.031**	**0.012**	**0.016**	**0.015**	**0.008**	**0.011**	**0.018**

Table II. *We compare the Absolute Trajectory Error (ATE) of deep learning SLAM systems based on nine sequences from the TUM-RGBD dataset against the ATE of our algorithm.*

	360	**desk**	**desk2**	**floor**	**plant**	**room**	**rpy**	**teddy**	**xyz**	**avg**
DeepV2D [39]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
DeepV2D(TartanAir)	0.182	0.652	0.633	0.579	0.582	0.776	0.053	0.602	0.150	0.468
DeepFactors [9]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
DROID-SLAM [41]	0.111	0.018	0.042	0.021	0.016	0.049	0.026	0.048	0.012	0.038
Ours	**0.057**	**0.014**	**0.021**	**0.016**	**0.014**	**0.032**	**0.017**	**0.020**	**0.008**	**0.022**

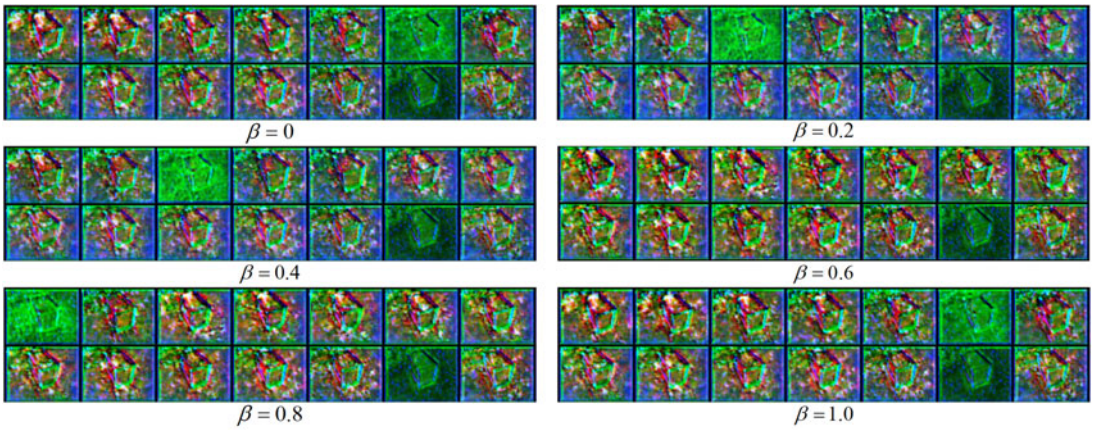


Figure 3. *The images illustrate the visualization of the latent space for varying β-parameters.*

4.2. Ablation study

To evaluate the impact of individual model components on overall performance, we conduct an in-depth analysis of their contributions to the final results, systematically removing the EMA and Bayes online learning components. This process further aids in identifying which components are most critical for enhancing the model's generalization ability. In this section, we utilize the ETH3D [32] (RGB-D) dataset to assess the impact of each individual component.

By analyzing the latent space representation presented in Figure 3, we determined the optimal β value for EMA. The results indicate that a β value of 0.6 yields the most pronounced feature differentiation, thereby enhancing the network's ability to represent features more effectively. In contrast, at other β values, the feature differentiation of the embedding is less pronounced, which fails to significantly improve the network's representational capacity.

Finally, we conducted ablation experiments utilizing six subsets of the ETH3D (RGB-D) dataset, each varying in illumination conditions and camera transformation speed. We employed ATE to assess the impact of each component on the algorithm, thereby demonstrating the contribution of our method to the system by comparing the results after systematically removing different components.

The comparison results are presented in Table III, where the ATE is reduced by 6.4% and 7.1%, respectively, upon incorporating the EMA and Bayes online learning methods. Overall, the EMA method contributes more significantly to the improvement of system performance. However, in certain datasets (indicated by the lower horizontal line in Table III), the presence of low-texture regions, rapid camera motion, insufficient prior information, and low correlation between adjacent frames led to suboptimal performance of both components. Our method integrates both the EMA and Bayes online learning components, achieving an overall ATE reduction of 21%, thereby outperforming each component

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

Table III. *The table presents the Absolute Trajectory Error (ATE) results for the six sequences derived from the ETH3D (RGB-D) dataset in the ablation study.*

	table3	**table4**	**table7**	**cables1**	**plant2**	**planar2**	**avg**
DVO-SLAM [21]	0.0082	0.0181	0.0074	0.0042	0.0021	0.0024	0.0071
ElasticFusion [46]	-	0.0122	-	0.0181	0.0173	0.0111	0.0147
ORB-SLAM2 [28]	0.0073	0.0084	0.0103	0.0075	0.0032	0.0054	0.0070
ORB-SLAM3 [7]	0.0053	0.0083	0.0135	0.0110	0.0026	0.0042	0.0075
DROID-SLAM [41]	0.0080	0.0081	0.0058	0.0026	0.0049	0.0014	0.0051
DROID-SLAM + EMA	0.0074	0.0077	0.0052	0.0020	0.0051	0.0014	0.0048
DROID-SLAM + Bayes	0.0081	0.0074	0.0055	0.0024	0.0040	0.0012	0.0047
Ours	**0.0065**	**0.0071**	**0.0040**	**0.0019**	**0.0037**	**0.0011**	**0.0041**

individually. The interaction between the two components effectively addresses issues related to insufficient prior information and dataset limitations, ultimately demonstrating superior performance across all sequences.

5. Conclusion

In this study, we improve the generalization capacity of the DROID-SLAM system across diverse location scenarios using non-training datasets, by integrating Exponential Moving Average (EMA) and Bayes online learning techniques. In non-training environments, previously trained models frequently struggle to adapt to novel and unseen conditions. The integration of EMA and Bayes methods facilitates real-time adaptation during the inference phase, thereby effectively mitigating the challenges posed by data variations and uncertainties in real-world scenarios. We observe that the enhanced DROID-SLAM system demonstrates substantial improvements over traditional SLAM systems, other deep learning-based SLAM methods, and previous versions of DROID-SLAM.

Through comprehensive experiments, we find that the enhanced DROID-SLAM system shows notable improvements compared to traditional SLAM systems, other deep learning-based SLAM methods, and previous versions of DROID-SLAM. However, our improvements still have certain limitations. In the absence of Visual Odometry focusing on global maps and loopback detection, the system may fail to achieve the desired robustness in environments with drastic lighting changes or significant motion blur. Future research may explore further integration with other advanced technologies, as well as the selection of more sophisticated feature extraction and adaptive methods to further enhance the system's performance in complex environments. Additionally, the impact of the algorithms in practical applications could be assessed through hardware deployment, with a focus on optimizing computational efficiency and real-time performance, while enhancing robustness under more realistic and complex environmental conditions. Further optimization of these methods may yield more substantial performance improvements.

Author contributions. Jun Liu and Haihang Deng conceived and designed the study. Haihang Deng performed statistical analyses and wrote the article.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

References

[1] S. E. Ada, E. Ugur and H. L. Akin, "Generalization in transfer learning: Robust control of robot locomotion," *Robotica* **40**(11), 3811–3836 (2022).

[2] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir and A. u. Haq, "Object detection through modified YOLO neural network," *Sci. Programming-NETH* **2020**(1), 8403262 (2020).

[3] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu and T. Liu, "Understanding and improving early stopping for learning with noisy labels," *Adv. Neural Inf. Process. Syst.* **34**, 24392–24403 (2021).

[4] N. Bjorck, C. P. Gomes, B. Selman and K. Q. Weinberger, "Understanding batch normalization," *Adv. Neural Inf. Process. Syst.* **31**, 7694–7705 (2018).

[5] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Rob. Res.* **35**(10), 1157–1163 (2016).

[6] Z. Cai, A. Ravichandran, S. Maji, C. Fowlkes, Z. Tu and S. Soatto, "Exponential moving average normalization for self-supervised and semi-supervised learning," **In:** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Piscataway, NJ: IEEE (2021) pp. 194–203.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robotics* **37**(6), 1874–1890 (2021).

[8] F. A. A. Cheein, "SLAM-based maneuverability strategy for unmanned car-like vehicles", *Robotica* **31**(6), 905–921 (2013).

[9] J. Czarnowski, T. Laidlow, R. Clark and A. J. Davison, "DeepFactors: Real time probabilistic dense monocular SLAM," *IEEE Robot. Autom. Lett.* **5**(2), 721–728 (2020).

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," **In:** *Proceedings of the IEEE conference on computer vision and pattern recognition*, Piscataway, NJ: IEEE (2017) pp. 5828–5839.

[11] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007).

[12] A. A. Deraz, O. Badawy, M. A. Elhosseini, M. Mostafa, H. A. Ali and A. I. El-Desouky, "Deep learning based on LSTM model for enhanced visual odometry navigation system," *Ain Shams Eng. J.* **14**(8), 102050 (2023).

[13] L. Di Giammarino, L. Brizi, T. Guadagnino, C. Stachniss and G. Grisetti, "MD-SLAM: Multi-cue direct SLAM," **In:** *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Piscataway, NJ: IEEE (2022) pp. 11047–11054.

[14] M. Dusmanu, J. L. Schönberger and M. Pollefeys, "Multi-view optimization of local feature geometry," **In:** *Computer Vision–ECCV 2020: 16th European Conference Part*, Springer, Glasgow, UK (2020) pp. 670–686. August 23–28, 2020, Proceedings, Part I 16 2020.

[15] J. Engel, V. Koltun and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(3), 611–625 (2017).

[16] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," **In:** *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Piscataway, NJ: IEEE (2014) pp. 15–22.

[17] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer and P. N. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intel.* **115**, 105151 (2022).

[18] I. Gruevski and S. Gaber, "Basic time series models in financial forecasting," *J. Econ.* **6**(1), 76–89 (2021).

[19] N. Jaouedi, N. Boujnah and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *J. King Saud Univ. Comput. Inf. Sci.* **32**(4), 447–453 (2020).

[20] M. M. Johari, C. Carta and F. Fleuret, "ESLAM: Efficient dense slam system based on hybrid representation of signed distance fields," **In:** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Piscataway, NJ: IEEE (2023) pp. 17408–17419.

[21] C. Kerl, J. Sturm and D. Cremers, "Dense visual slam for RGB-D cameras," **In:** *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Piscataway, NJ: IEEE (2013) pp. 2100–2106.

[22] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," **In:** *2007, 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Piscataway, NJ: IEEE (2007) pp. 225–234.

[23] H.-C. Lee, S.-H. Lee, M. H. Choi and B.-H. Lee, "Probabilistic map merging for multi-robot RBPF-SLAM with unknown initial poses," *Robotica* **30**(2), 205–220 (2012).

[24] A. Li, A. Boyd, P. Smyth and S. Mandt, "Detecting and adapting to irregular distribution shifts in Bayesian online learning," *Adv. Neural Inf. Process. Syst.* **34**, 6816–6828 (2021).

[25] S. Li, J. Gu, Z. Li, S. Li, B. Guo, S. Gao, F. Zhao, Y. Yang, G. Li and L. Dong, "A visual SLAM-based lightweight multi-modal semantic framework for an intelligent substation robot," *Robotica* **42**(7), 2169–2183 (2024).

[26] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang and H. Zhang,"CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction," **In:** *2019 International Conference on Robotics and Automation (ICRA)*, Piscataway, NJ: IEEE (2019) pp. 5218–5223.

[27] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015).

[28] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.* **33**(5), 1255–1262 (2017).

[29] Y. Nie, L. De Santis, M. Carratù, M. O'Nils, P. Sommella and J. Lundgren, "Deep melanoma classification with k-fold cross-validation for process optimization," **In:** *2020 IEEE international symposium on medical measurements and applications (MeMeA)*, IEEE (2020) pp. 1–6.

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press

[30] T. Qin, P. Li and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.* **34**(4), 1004–1020 (2018).
[31] A. Rosinol, J. J. Leonard and L. Carlone, "Probabilistic volumetric fusion for dense monocular slam," **In:** *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Piscataway, NJ: IEEE (2023) pp. 3097–3105.
[32] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," **In:** *Conference on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ: IEEE (2017) pp. 3260–3269.
[33] M. Segu, A. Tonioni and F. Tombari, "Batch normalization embeddings for deep domain generalization," *Pattern Recognit.* **135**, 109115 (2023).
[34] W. Sheng, X. Lu and X. Li, "Data augmentation by separating identity and emotion representations for emotional gait recognition," *Robotica* **41**(5), 1452–1465 (2023).
[35] N. Silberman, D. Hoiem, P. Kohli and R. Fergus, "Indoor segmentation and support inference from RGBD images, **In:** "*Computer Vision–ECCV 2012: 12th European Conference on Computer Vision*," Springer, Florence, Italy (2012) pp. 746–760. October 7–13, 2012, Proceedings, Part V 12 2012.
[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," **In:** *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Piscataway, NJ: IEEE (2012) pp. 573–580.
[37] C. Tang and P. Tan, "BA-Net: Dense bundle adjustment network," (2018) arXiv preprint arXiv: 1806.04807
[38] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," **In:** *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Piscataway, NJ: IEEE (2018) pp. 1542–1547.
[39] Z. Teed and J. Deng, "Deepv2d: Video to depth with differentiable structure from motion," (2018) arXiv preprint arXiv: 1812.04605
[40] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," **In:** *Computer Vision–ECCV 2020: 16th European Conference*, Springer, Glasgow, UK (2020) pp. 402–419. August 23–28, 2020, Proceedings, Part II 16 2020.
[41] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," *Adv. Neural Inf. Process. Syst.* **34**, 16558–16569 (2021).
[42] H. Wang, C. Zhang, Y. Song, B. Pang and G. Zhang, "Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios," *Robotica* **38**(2), 350–373 (2020).
[43] W. Wang, Y. Hu and S. Scherer, "TartanVO: A generalizable learning-based VO," **In:** *Conference on Robot Learning*, PMLR (2021) pp. 1761–1772.
[44] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor and S. Scherer, "TartanAir: A dataset to push the limits of visual SLAM," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Piscataway, NJ: IEEE (2020b) pp. 4909–4916.
[45] Y. Wang, L. Zhao, L. Gong, X. Chen and S. Zuo, "A monocular SLAM system based on SIFT features for gastroscope tracking," *Med. Biol. Eng. Comput.* **61**(2), 511–523 (2023).
[46] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," *Robotics: Science and Systems*, Rome, Italy: RSS Foundation (2015) pp. 3. volume 11.
[47] L. Xiao, J. Wang, X. Qiu, Z. Rong and X. Zou, "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robot. Auton. Syst.* **117**, 1–16 (2019).
[48] M. Xiao, Y. Li, X. Yan, M. Gao and W. Wang, "Convolutional neural network classification of cancer cytopathology images: taking breast cancer as an example," **In:** *Proceedings of the 2024 7th International Conference on Machine Vision and Applications*, New York, NY: ACM (2024) pp. 145–149.
[49] N. Yang, L. v. Stumberg, R. Wang and D. Cremers, "D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," **In:** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Piscataway, NJ: IEEE (2020) pp. 1281–1292.
[50] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," (2017) arXiv preprint arXiv:1705.10941
[51] X. Yu, W. Zheng and L. Ou, "CPR-SLAM: RGB-D SLAM in dynamic environment using sub-point cloud correlations," *Robotica* **42**(7), 2367–2387 (2024).
[52] K. Zhang, C. Dong, H. Guo, Q. Ye, L. Gao, S. Xiang, X. Chen and Y. Wu, "A semantic visual SLAM based on improved mask R-CNN in dynamic environment," *Robotica* **42**(10), 3570–3591 (2024).
[53] Q. Zhong and X. Fang, "A BigBiGAN-based loop closure detection algorithm for indoor visual SLAM," *J. Electr. Comput. Eng.* **2021**(1), 10 (2021).
[54] T. Zhou, M. Brown, N. Snavely and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," **In:** *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Piscataway, NJ: IEEE (2017) pp. 1851–1858.
[55] J. Zubizarreta, I. Aguinaga and J. M. M. Montiel, "Direct sparse mapping," *IEEE Trans. Robot.* **36**(4), 1363–1370 (2020).

Cite this article: J. Liu and H. Deng (2025). "Improve generalization for neural visual-SLAM with Bayes online learning", Robotica **43**, 1660–1674. https://doi.org/10.1017/S0263574725000360

https://doi.org/10.1017/S0263574725000360 Published online by Cambridge University Press