Abstract—Accurate and robust localization is a fundamental need for mobile agents. Visual–inertial odometry (VIO) algorithms exploit the information from the camera and inertial sensors to estimate position and translation. Recent deep-learning-based VIO models attract attention as they provide pose information in a data-driven way, without the need of designing hand-crafted algorithms. Existing learning-based VIO models rely on recurrent models to fuse multimodal data and process sensor signals, which are hard to train and not efficient enough. We propose a novel learning-based VIO framework with external memory attention that effectively and efficiently combines visual and inertial features for state estimation. Our proposed model is able to estimate pose accurately and robustly, even in challenging scenarios, for example, on overcast days and water-filled ground, which are difficult for traditional VIO algorithms to extract visual features. Experiments validate that it outperforms both traditional and learning-based VIO baselines in different scenes.

Index Terms—Attention mechanisms, deep neural networks, inertial sensor, multimodal learning, sensor fusion, visual–inertial odometry (VIO).

1. INTRODUCTION

Accurate positioning plays a vital role in mobile agents to support many applications, for example, self-driving vehicles, delivery robots, and augmented reality. It is difficult to achieve long-term localization only using a single sensor in complex environments. With a rich combination of sensors [e.g., camera, inertial measurement unit (IMU), light detection and ranging (LIDAR)] on mobile platforms, effective sensor fusion is always a central problem in robust designing localization and navigation system. However, the problem of sensor fusion is complicated due to real-world issues, such as the diversity of error characteristics as well as the spatial and temporal misalignment of different sensors. Due to the natural complementarity of the camera and IMU, researchers throw their interest in integrating the camera and IMU as visual–inertial odometry (VIO) to provide pose information. Fully exploiting the properties and advantages of two sensors is important to improving positioning accuracy and robustness.

Existing VIO algorithms can be divided into traditional model-based methods [1], [2], [3] and deep-learning-based methods [4], [5]. Traditional model-based VIOs rely on hand-designed feature detectors, and filtering [6] or nonlinear optimization [1]-based sensor fusion. Learning-based VIO models can extract features automatically and learn pose from images and IMUs in an end-to-end data-driven way [4], [7], [8]. These methods have shown their superiority in robust operation in challenging environments.

However, previous learning-based VIOs are not efficient enough and hard to train and infer. Effective feature fusion is less explored, compared with much effort on feature learning. To solve the feature fusion problem more effectively and efficiently, we propose EMA-VIO, a novel framework that introduces External Memory Attention into learning-based VIOs. Previous researches normally leverage long short-term memory (LSTM) network to fuse visual and inertial features [4], [7], [8]. It is difficult for recurrent models to train and infer in parallel, as the recurrent model has to process sequential data in order. Recently, in natural language processing (NLP) and image processing, there is growing interest in replacing the recurrent model with attention mechanisms. Our proposed external memory attention-based sensor fusion...
is able to improve the performance of VIOs and reduces the memory and computation requirement. To further enhance the capacity of pose learning, we impose multistate pose constraints to refine the pose estimation. Experiments in the real world with our own robot platforms demonstrate that our EMA-VIO can operate in less-feature environments.

The main contributions of this article can be summarized in the following three parts.

1) We propose a novel external memory attention-based fusion mechanism for the VIO learning framework that effectively combines features from two sensors and improves the model performance.

2) A multistate geometric constraint is introduced into the pose loss function that speeds up training convergence and contributes more accurate pose results.

3) Real-world experiments were conducted to evaluate our proposed learning-based VIO model above to our own robot platform. It shows that our learning model can perform well in challenging environments, that is, overcast days and wet ground.

II. RELATED WORK

Recently, deep learning has been widely applied in solving real-world problems, for example, COVID-19 diagnosis [9]. Our work is mainly relevant to deep-learning-based visual positioning, deep-learning-based VIO, and attention mechanisms. Thus, we discuss them in this section.

A. Deep-Learning-Based Visual Positioning

Learning-based visual positioning methods have made great progress in recent years. DeepVO [10] proposes end-to-end visual odometry based on deep neural networks, which use convolutional neural networks (ConvNets) to encode image sensor data into visual features, and a two-layer LSTM network to update visual features to provide pose in an end-to-end manner. There are some recent works that further improve the performance of the learning model [11], [12], [13], [14], [15]. Especially, Brahmbhatt et al. [16] proposed a geometry-aware learning framework for camera localization that improves the accuracy capability of both visual odometry and relocalization. AtLoc [17] proposed a camera localization model guided by a self-attention mechanism, which introduces the encoded features into the attention mechanism to form an attention matrix and then the attention matrix is summed with the original visual features to form new features, which it improves the accuracy of localization. We introduce external memory attention into the learning-based localization model that has never been seen in previous works.

B. Deep-Learning-Based VIO

VINet [4] is an end-to-end trainable VIO model that takes sensor data from a monocular camera and IMU as input and adopts ConvNet and recurrent network to encode multimodal features from visual and inertial information, respectively. VINet concatenates the two kinds of features directly according to channel dimensions and process features with the recurrent model. To further study the multimodal fusion problem for VIO, [18] proposed a selective fusion mechanism for learning-based VIO to fuse features from two sides. Compared with direct fusion, their proposed fusion method achieves better performance in localization and is more robust in dealing with multiple cases of sensor data degradations. SelfVIO [19] proposes a self-supervised trained monocular VIO for localization and depth prediction, but it cannot provide pose in an absolute scale metric. In [20], two ConvNets are proposed for the feature extraction of inertial data, replacing the LSTM network in the inertial feature extraction part of the selectfusion network [18], and the extracted features are then spliced and interpolated to form the inertial feature matrix, after which the feature extraction is performed by the attention mechanism. However, all of these frameworks still rely on recurrent models to fuse multimodal features and update system states, which are hard to train and limit the efficiency of the entire framework. CodeVIO [21] proposes a lightweight, tightly coupled VIO network along with a depth network to provide accurate state estimates and dense depth estimates of the surroundings. Aslan et al. [22] learned the pose of a drone in a more stable way, by denoising inertial data, adopting Inception-v3 to extract visual features from two consecutive images, and producing pose via Gaussian Process Regression.

C. Attention Mechanisms

There are growing interests in attention mechanisms in the deep learning community. Attention has been introduced into the encoder–decoder architecture to associate information in a long range. Mnih et al. [23] proposed to use the attention mechanism in recurrent networks for image classification, which inspires the subsequent development of attention mechanisms. Bahdanau et al. [24] applied the attention mechanism to the NLP domain, in which a sequence-to-sequence architecture with an attention model is used for machine translation with astonishing results. Recently, the Transformer [25] model has attracted great attention, in which the authors used self-attention mechanisms to replace recurrent models in the neural language processing problems, showing superiority in translation quality while being more parallelizable and requiring significantly less time to train. Self-attention has been used in visual processing [26]. Furthermore, in order to solve the problem of excessive computation caused by the self-attention mechanism in visual tasks, the external attention mechanism is proposed by Guo et al. [27] showing comparable or superior performance toward self-attention mechanisms with lower computational and memory costs.

III. DEEP VIO

In this section, we discuss our proposed deep-learning-based VIO model in detail. The framework of our approach is shown in Fig. 1. Our framework mainly includes feature encoders, an external memory attention-based feature fusion module, and a pose regressor. The network inputs a pair of images and a sequence of IMU data and outputs pose transformation. To effectively fuse visual and inertial features, we introduce an attention mechanism with memory unit assistance, unlike the previous works commonly using LSTM [4], [7], [8].
A. Feature Encoders

1) Visual Feature Encoder: In the visual feature encoder, we aim to extract geometry-meaningful features from a combination of two images, so that FlowNet [28] is adopted to provide features relevant to optical flow and motion estimation. There are six convolutional layers, and each convolutional layer is followed by a ReLu activation layer. Two consecutive frames are concatenated channel-wise and input to the network. The first image can be defined as reference image frame $P_{\text{ref}}$ and the second image as target image frame $P_{\text{tar}}$. The front-end convolution part of FlowNetS is used to extract features from the combined images to form the feature representation, after which a pooling layer to downsample the high-dimensional feature representation and a fully connected layer maps it into a visual feature vector $F_V$.

$$F_V = \text{FlowNet}([P_{\text{ref}}; P_{\text{tar}}]).$$  \hspace{1cm} (1)

2) Inertial Feature Encoder: Instead of using LSTM to process inertial data, we adopt a WaveNet style module to extract inertial features [29]. Here, we construct an inertial feature encoder based on dilated causal ConvNets. As shown in Fig. 2, in the dilated causal convolutional network, the prediction of all time steps can be performed in parallel. And the training process of the network is faster and more efficient because the circular connection structure of the recurrent model is eliminated. The inertial data between adjacent image frames are as input, we use a causal convolutional network to process a sequence of inertial data to form inertial feature vectors. The entire convolutional network has four layers, and each convolutional layer is followed by a gate activation unit

$$z = \text{tanh}(W_{f,k} \ast x) \odot \sigma(W_{g,k} \ast x)$$ \hspace{1cm} (2)

where $x$ represents the output of each causal convolutional layer, $W$ denotes the weight matrix of the convolution kernel, $g$ and $f$ represent the gate and the filter, respectively, $k$ is the layer index, $\ast$ denotes the convolution operation, $\odot$ denotes the elementwise multiplication operation, $\sigma(\cdot)$ denotes the elementwise sigmoid activation function, and tanh denotes the hyperbolic tangent activation function. The inertial feature is obtained from the sum of the result of each layer.

$$F_I = \text{WaveNet}(I).$$ \hspace{1cm} (3)

B. Memory Attention-Based Feature Fusion

We combine the feature vectors from two feature encoders in their channel dimension $[F_V; F_I]$ and then selectively fuse the feature vectors using the attention mechanism to obtain the fused feature vector $F_{\text{att}}$

$$F_{\text{att}} = \text{Attention}([F_V; F_I]).$$ \hspace{1cm} (4)

In recent years, the attention mechanism becomes a hot topic in both NLP and computer vision. The attention mechanism can be used as a good solution to associate information.
Our proposed memory attention-based feature fusion is based on nonlocal self-attention with introduced memory variables to further enhance its capacity to fuse features.

1) Self-Attention: In the self-attentive mechanism, the input feature matrix \( F \) is converted to a query matrix \( Q \), a key matrix \( K \), and a value matrix \( V \) through a linear layer according to the following equation:

\[
Q = W_q F, \quad K = W_k F, \quad V = W_v F
\]

where \( W_q \), \( W_k \), and \( W_v \) are the transform matrix of linear layers.

Then we calculate the self-attention matrix \( A \) via the following equation, which represents the attention weights of the neural network at the corresponding positions of the \( A \) matrix:

\[
A = \text{Softmax}(QK^T)V.
\]

After that, the residual connection is performed according to the following equation (7) to connect the attention matrix with the original feature matrix to obtain the feature matrix after reweighting by the attention mechanism, which can be further used for the subsequent positional decoding and pose regression:

\[
F_{\text{att}} = \alpha(A) + F
\]

where the linear layer \( \alpha(A) = W_a F \) denotes a learnable scaled self-attention vector.

2) External Memory-Aided Self-Attention: Inspired by the external attention mechanism, we introduce the memory variables into the self-attention mechanism to form our attention-based fusion module as shown in Fig. 3. Based on the self-attention mechanism, two memory units \( M_1 \) and \( M_2 \) are added to the channels of calculating the \( Q \) and \( V \) matrix of the self-attention. These two external learnable memory variables can be implemented by two linear layers. Therefore, these two memory variable parameters can be updated during backpropagation

\[
Q_{\text{att}} = \text{Norm}(QM_{Q1}^T)M_{Q2}
\]

\[
K_{\text{att}} = \text{Norm}(KM_{K1}^T)M_{K2}
\]

where \( M_{Q1}, M_{Q2}, M_{K1}, \) and \( M_{K2} \) are linear matrix to update \( Q \) and \( K \). Also, since these two variables are not determined by a particular sample, but are bounded by the samples in the whole dataset, they act as a regularizer for the samples in the whole dataset, which in turn improves the generative ability of the attention mechanism and the pose estimation accuracy.

The introduced memory units can be viewed similarly to the hidden states of the recurrent model that store and maintain the history information of a sequence. With our proposed fusion mechanism, we are able to replace the LSTM network commonly used in previous works. Our experiments show that this mechanism further improves the model performance and meanwhile reduces computational and memory requirements.

C. Pose Regressor With Geometric Multistate Constraint

After feature fusion, a pose regressor is designed to map the fused features to a relative pose vector between two frames, that is, 3-D position displacement \( \mathbf{t} \) and 3-D rotation \( \mathbf{\psi} \). It consists of two linear layers with a ReLu activation function between them to perform pose regression

\[
(t, \psi) = f_{\text{pose}}(F_{\text{att}}).
\]

The geometric relation between poses is important information to pose update, as illustrated in Fig. 4. Thus, our pose regressor further exploits this geometric multistate constraint into pose optimization.

The position displacement and rotation vectors \((t, \psi)\) between two states are used to form transformation matrix...
\( \mathbf{T} \) representing motion transformation

\[
\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}
\]

(11)

where \( \mathbf{R} \) is the rotation matrix that can be derived from Euler angle-based rotation vector \( \psi \).

In our model, the length of image sequence is chosen as \( n \), and the pose transformation from the \( i \)th frame to the \( j \)th frame is expressed as \( \mathbf{T}_{i \rightarrow j} \), \( \mathbf{T} \in \text{SE}(3) \). The pose translation from the first frame to the last frame of the sequence can be expressed as \( \mathbf{T}_{1 \rightarrow n} \). The relative poses between frames have to satisfy the following multistate constraint:

\[
\mathbf{T}_{1 \rightarrow 2} \cdot \mathbf{T}_{2 \rightarrow 3} \cdot \ldots \cdot \mathbf{T}_{(n-1) \rightarrow n} = \mathbf{T}_{1 \rightarrow n}.
\]

(12)

This multistate geometric constraint will be incorporated into the loss function to refine pose estimation.

**D. Loss Function and Optimizer**

Finally, a loss function is defined to optimize the entire framework to estimate poses. Our loss function \( L_{\text{total}} \) consists of a per-frame loss \( L_{\text{frame}} \) and a sequence loss \( L_{\text{seq}} \)

\[
L_{\text{total}} = L_{\text{frame}} + L_{\text{seq}}.
\]

(13)

Based on this loss function, we adopt the ADAM optimizer to recover the optimal parameters of the neural network framework.

1) **Per-Frame Loss**: The per-frame loss represents the loss of estimated poses and ground-truth poses between two frames. Here, we use the Euler angle \( \psi \) to denote the rotation between two image frames and the relative translations in metric units \( \mathbf{t} \). And the root mean square error (RMSE) is adopted

\[
L_{\text{frame}} = \text{RMSE}(\mathbf{t}_{\text{frame}} - \hat{\mathbf{t}}_{\text{frame}}) + \lambda_1 \cdot \text{RMSE}(\psi_{\text{frame}} - \hat{\psi}_{\text{frame}})
\]

where \((\mathbf{t}, \hat{\mathbf{t}})\) is the ground-truth pose, and \( \lambda_1 \) is a scale-factor to balance the per-frame loss between translation and rotation.

2) **Sequence Loss**: The sequence loss exploits the multistate geometric constraint to define a loss function between the ground truth and the predicted values of the whole sequence. Following (12), the sequence transformation matrix representing the relative motion from the first frame to the last frame of a sequence can be acquired, and the translation and rotation \( (\mathbf{t}_{\text{seq}}, \psi_{\text{seq}}) \) of a sequence can be derived correspondingly. The ground truth of a sequence motion transformation denotes \( (\mathbf{t}_{\text{seq}}, \psi_{\text{seq}}) \)

\[
L_{\text{seq}} = \text{RMSE}(\mathbf{t}_{\text{seq}} - \hat{\mathbf{t}}_{\text{seq}}) + \lambda_2 \cdot \text{RMSE}(\psi_{\text{seq}} - \hat{\psi}_{\text{seq}})
\]

(15)

where RMSE denotes the root mean square error function and \( \lambda_2 \) denotes the scale factor between the displacement error and the rotation error.

**IV. EXPERIMENTS**

In this section, we first evaluated our proposed deep VIO model on the KITTI dataset, which is publicly available, and commonly used as a benchmark to evaluate both classical and learning-based positioning algorithms. Then, we conducted the experiment with our own robot platform to collect data and evaluate our proposed model in challenging environments. The framework of our method was implemented with Pytorch and trained using an NVIDIA RTX3090.

**A. Dataset**

1) **KITTI Dataset**: The KITTI dataset is a dataset with data collections on car driving. There are ten sequences including camera, IMU data, and ground-truth pose. The frequency of IMU data in the synchronized sequence is only 10 Hz. Instead, we use the inertial data from the unsynchronized data folder, in which IMU data is 100 Hz, and synchronize data by hand. We selected Sequence 00–08 (03 is not available) as the training set to train our model and used Sequence 09 and 10 as the test set. The frequency of images is 10 Hz, and the ground-truth pose is provided at the same timestep as the images.

2) **Our Robot Dataset**: We used a ground vehicle platform equipped with a monocular red, green, blue (RGB) camera and an MEMS IMU, as shown in Fig. 5, to collect images in 10 Hz and inertial data in 100 Hz. Meanwhile, to capture the ground-truth pose, an integrated navigation system with high-precision fiber IMU and global navigation satellite system (GNSS) was adopted. The pose accuracy provided by the integrated navigation system (INS)/GNSS system is around 0.1 m. The experiment was conducted on overcast days with relatively poor lighting conditions and a large amount of water on wet ground that will cause light reflections, as shown in Fig. 6. This scene is difficult for traditional VIOs to extract and match visual features. Our collected dataset includes seven sequences (00–06). We used two of them as the test set and the rest as the training and validation sets.

**B. Positioning Results on the KITTI Dataset**

We first evaluated our proposed EMA-VIO on the KITTI dataset and compare the pose results of our EMA-VIO with two traditional VIO methods (i.e., VINS-Mono [1], open keyframe-based visual inertial SLAM (OKVIS) [30]) and two learning-based pose estimation methods (i.e., VInet [4] and [12]). VInet extracts visual features using FlowNet, but uses LSTM to extract inertial features and fuses visual and inertial features with LSTM. From Fig. 7, clearly, the trajectories of our proposed EMA-VIO on Sequence 09 and 10 of the KITTI dataset are closer to the ground-truth trajectories of these two sequences.

Furthermore, as demonstrated in Table I, the official evaluation metric of the KITTI dataset is adopted to compare the
TABLE I

| Seq. | OKVIS | VINS-Mono | Zhan et al. | VINet | EMA-VIO (Ours) |
|------|-------|-----------|-------------|-------|----------------|
|      | t_{rel}(%) | r_{rel}(%) | t_{rel}(%) | r_{rel}(%) | t_{rel}(%) | r_{rel}(%) |
| 09   | 9.77 | 2.97 | 41.47 | 2.41 | 11.92 | 3.60 | 11.83 | 3.00 | **8.68** | **1.54** |
| 10   | 17.30 | 2.82 | 20.35 | 2.73 | 12.62 | 3.43 | 8.60 | 4.39 | **7.46** | **2.26** |
| average | 13.51 | 2.90 | 30.91 | 2.57 | 12.27 | 3.52 | 10.22 | 3.69 | **8.07** | **1.90** |

*_{t_{rel}} and r_{rel} indicate the averaged translation and rotation drift of all subsequences with length of (100, ..., 800 m).

quantitative results of three VIO models, which is the average translational RMSE drift (%) and the average rotational RMSE drift (°/100 m). It can be found that our method outperforms VINS-mono and VINet in terms of both translation and rotation. Compared with VINet, our proposed learning-based VIO significantly reduces the translation drift by around 20% and the rotation drift by 48%. Both VINet and our EMA-VIO show advantages over the VINS-Mono, as cameras and IMU are not strictly calibrated and time-synchronized on the KITTI dataset, which causes trouble to the traditional hand-designed VIO system. It can be seen that the introduced external memory attention effectively integrates visual and inertial features, which contributes to building potential relationships among different features and adaptively adjusting the feature weights at a different place according to the training data. Moreover, the introduced memory unit improves the ability of the attention module to deal with the feature with time sequence information, so that it can be used to replace the decoder based on LSTM in the previous learning-based VIO.

C. Positioning Results on Our Robot Dataset

Then, an outdoor robot navigation experiment was conducted to collect data for the evaluation of our proposed VIO model. As the height of our ground-truth pose from the GNSS/IMU system is not accurate enough, we mainly evaluate positioning results on the plane. Fig. 8 shows the trajectories of our proposed EMA-VIO and a baseline, that is, VINet. Both of them can perform well in these challenging scenes. We find that VINS-Mono fails to extract and track feature points in the scenes of our dataset, as poor lighting conditions of the overcast day and the water on the ground with light reflections cause trouble to the hand-designed feature detectors. As shown in Fig. 9, the extracted visual features of VINS are not enough to perform the positioning algorithm. This indicates that deep neural networks are more powerful to extract features for localization. The quantitative results are shown in Table II. The evaluation metric we used here is the horizontal position error (HPE) according to the following equation:

\[
\text{Error} = \sqrt{\frac{1}{n} (p_i - \hat{p}_i)^2}
\]
where \( n \) is the length of the trajectory sequence, \( \mathbf{p}_i \) denotes the planar localization result \((x_i, y_i)\) of the estimated value of the algorithm, and \( \hat{\mathbf{p}}_i \) denotes the planar localization value \((\hat{x}_i, \hat{y}_i)\) of the ground truth. Clearly, our EMA-VIO outperforms VINS-net in positioning results. This indicates the effectiveness of our introduced external memory attention and geometric constraints. Our method shows good robustness in the challenging navigation scene. Though the traditional VIO method performs well in scenes with rich texture and demonstrates good localization results, the hand-crafted features are hard to be applied in challenging navigation scenes (e.g., with sparse features and poor light conditions), and thus traditional VIO approaches usually perform not well due to the lack of effective feature tracking. The deep-learning-based VIO approach continues to show good performance in challenging experimental scenarios due to the learned features.

**D. Ablation Study**

In order to validate the performance improvement of each module in our proposed approach, we also conducted an ablation study of our EMA-VIO model on the KITTI dataset. The results are shown in Table III. VIO indicates the baseline learning-based VIO with LSTM for multimodal fusion and inertial data processing, while EA means replacing LSTM with external memory attention for feature fusion, WaveNet means replacing LSTM with WaveNet for inertial data processing, and MC means adding multistate geometric constraints. Meanwhile, due to further utilization of the transformation relationship between successive image frames, the convergence speed of the model is increased by introducing the multistate constraint in the training process.

It can be seen that our introduced external attention, WaveNet, and multistate geometric constraints further improve the positioning performance base model. Though the WaveNet degrades the rotation slightly, the introduction of WaveNet still improves positioning results and improves the model efficiency that will be discussed below, so we view it as a tradeoff to use WaveNet.

**E. Efficiency Analysis**

Finally, we come to analyze the efficiency of our proposed EMA-VIO. Two important terms are selected to compare with VINS-net. One is the number of parameters that represent the memory consumption, and the other one is multiply-accumulate operations (MACs) that represent the computation requirement. As FlowNet in both our EMA-VIO and VINS-net occupies large resources of memory and computation, we exclude FlowNet in our calculation, considering that this visual extraction module can be replaced with a new technique in the future that is not the main point in this work. Our EMA-VIO method shows great advantages not only in the number of parameters, but also in MACs, compared with VINS-net as shown in Fig. 10. Our model reduces the number

### Table II

| Test Sequence | VINS-mono | VINS-net | EMA-VIO (ours) |
|---------------|-----------|----------|----------------|
| 01            | Fail      | 2.9080 m | 2.3401 m       |
| 02            | Fail      | 4.3956 m | 4.2437 m       |
| Average       | Fail      | 3.6318 m | 3.2919 m       |

### Table III

| VIO | EA | WaveNet | MC | Seq09 | Seq10 | Avg |
|-----|----|---------|----|-------|-------|-----|
|     |    |         |    | \( t_{rel} (%) \) | \( r_{rel} (%) \) |     |
| ✔️  | ✔️ | ✔️      | ✔️ | 11.83 | 3.00   | ✔️  |
| ✔️  | ✔️ |         | ✔️ | 10.23 | 1.33   | ✔️  |
| ✔️  | ✔️ | ✔️      | ✔️ | 9.34  | 1.09   | ✔️  |
| ✔️  | ✔️ | ✔️      | ✔️ | 8.68  | 1.54   | ✔️  |

*\( t_{rel} \) and \( r_{rel} \) indicate the averaged translation and rotation drift of all subsequences with length of \((100, ..., 800 m)\).*
of the parameters from 35.695 to 1.868 M and reduces the MACs from 219.406 to 11.631 M. This is due to the fact that we replace LSTM with external memory attention for feature fusion and replace LSTM with WaveNet for inertial data processing.

V. CONCLUSION

In this work, we proposed a novel learning-based VIO framework with external memory attention for visual and inertial sensor fusion. Additionally, the multistate geometric constraint is introduced into our framework to further optimize the pose estimation. Real-world experiments on the public dataset and our own robot platform show that our proposed EMA-VIO outperforms both traditional and learning-based VIO models. Moreover, our model is more efficient and reduces the memory and computational requirement. However, our model still requires to be trained on the GPU, limiting its further applications in microrobotic systems. Besides, the stability of our proposed model needs to be verified in an open environment.

REFERENCES

[1] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” IEEE Trans. Robot., vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 3, pp. 611–625, Mar. 2017.

[3] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM,” IEEE Trans. Robot., vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[4] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “VINet: Visual-inertial odometry as a sequence-to-sequence learning problem,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 3995–4001.

[5] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, “D3 VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1281–1292.

[6] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” Int. J. Robot. Res., vol. 32, no. 6, pp. 690–711, 2013.

[7] C. Chen, S. Rosa, C. Xiaoxuan Lu, B. Wang, N. Trigoni, and A. Markham, “Learning selective sensor fusion for states estimation,” 2019, arXiv:1912.13077.

[8] P. Wei, G. Hua, W. Huang, F. Meng, and H. Liu, “Unsupervised monocular visual-inertial odometry network,” in Proc. 29th Int. Joint Conf. Artif. Intell., Jul. 2020, pp. 2347–2354.

[9] X. Yao, Z. Zhu, C. Kang, S.-H. Wang, J. M. Gorriz, and Y.-D. Zhang, “AdaD-FNN for chest CT-based COVID-19 diagnosis,” IEEE Trans. Emerg. Topics Comput. Intell., early access, Jun. 1, 2022.

[10] S. Wang, R. Clark, H. Wen, and N. Trigoni, “DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2017, pp. 2043–2050.
Changhao Chen (Member, IEEE) received the B.Eng. degree from Tongji University, Shanghai, China, in 2014, the M.Eng. degree from the National University of Defense Technology Changsha, China, in 2016, and the Ph.D. degree from the University of Oxford, Oxford, U.K., in 2020. He is a Lecturer with the College of Intelligence Science and Technology, National University of Defense Technology. His research interests include robotics, computer vision, and cyberphysical systems.

Xianfei Pan received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently a Vice Professor with the College of Intelligence Science and Technology, National University of Defense Technology. His current research interests include GPS/integrated navigation systems (INSs) and indoor navigation systems.

Ruochen Liu received the B.S. degree from the National University of Defense Technology, Changsha, China, in 2020, where he is currently pursuing the Ph.D. degree. His current research interests include multisensor integrated navigation and air-ground co-navigation.

Jiarui Cui received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include inertial navigation algorithm, and multisensor integrated navigation theory and application.

Jun Mao received the B.S. degree in electronic science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the M.S. and Ph.D. degrees in control science and engineering from the National University of Defense Technology, Changsha, China, in 2014 and 2019, respectively. He is currently a Lecturer with the National University of Defense Technology. His current research interests include vision-based navigation and bio-inspired navigation.