2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

DL-PoseNet: A Differential Lightweight Network for Pose Regression over SE(3)

Wenjie Li[1], Jia Liu[1], Yanyan Wang[2], Wei Hao[1], Dayong Ren[1], Lijun Chen[1,*]

***Abstract*— Accurate pose estimation over $SE(3)$ is fundamentally crucial for numerous perception tasks, including camera re-localization. While existing learning-based methods estimated from a series of RGB images have significantly improved the accuracy of pose, the majority of models still face one or two limitations. First, few representations on $SE(3)$ are smooth and differential, making them difficult to apply in deep learning frameworks. Second, they often require high computational resources due to complex deep network designs. We in this paper propose the DL-PoseNet to address these issues. Specifically, we present a novel representation for $SE(3)$ which follows the property of smoothness of the pose. We then design a lightweight neural network to regress the pose by developing a differential pose layer. Finally, we introduce a novel loss function and gradient descent method to better supervise the proposed lightweight pose network. Extensive experiments on the camera re-localization task on the Cambridge Landmarks and 7-Scenes datasets demonstrate the superior predictive accuracy and benefits of our method in comparison with the state-of-the-art.**

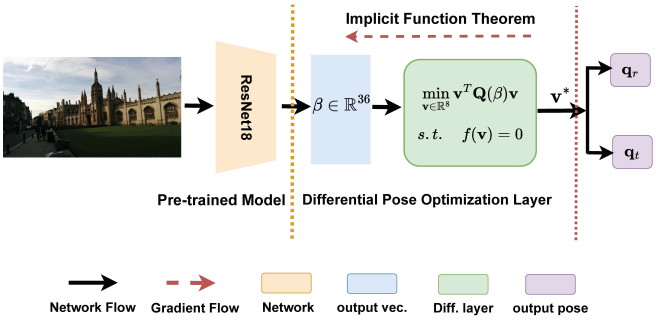

Fig. 1. The framework of the proposed differential lightweight network. Overall, the proposed network comprises two parts: (1) The RestNet part and (2) the differential pose optimization part. In the first part, we utilize the pretained ResNet-18 as our backbone where input images are required to resize to 224×224. Then the pose is regressed by the second part that builds a differential optimization layer. During the training stage, the implicit function theorem is leveraged to optimize the entire network.

I. INTRODUCTION

Estimating the 6-DoF pose from a single RGB image comprises the primary challenge in the fields of robotics [1], computer vision [2], [3], and aerospace engineering [4], [5]. Recently, deep learning technologies have provided us with an effective vehicle to estimate the pose, and a large number of deep models have been presented, which made impressive progress toward the accuracy of the pose.

Despite some advancements, the majority of learning-based methods combined with traditional pose representation (*i.e.* Euler angles, unit quaternions, or transformation matrices) endure one or two limitations. First, the singularities or discontinuities may arise in certain angles in the rotation part (e.g., the Gimbal Lock issue with Euler angles [6]). Second, the mainstream homogeneous transformation matrices endure the problem of overparameterization, which sometimes easily leads to memory inefficiency [7]. In addition, dual quaternions involving two quaternions are another vehicle to denote the pose but that have the nonsmooth feature due to the discontinuity property of the quaternions [8].

To address these deficiencies, recent 5D and 6D smooth representations of the rotation have been proposed [8] due to their properties of continuity and smoothness. Moreover, a symmetric matrix [9] also has been presented to represent the rotation over $SO(3)$ which is proved to be smooth and differential in learning-based rotation regression tasks. However, these approaches limit their scope to the rotation group $SO(3)$ and do not consider the translation part, making them difficult to leverage in the field $SE(3)$.

Moreover, many existing models, despite their advancements in pose accuracy, may consume more computing resources due to their complex deep convolutional layers, compared to traditional geometry-based methods. For instance, the PoseNet [10] is developed by integrating the GoogLeNet as the backbone, which is inevitable to be trained with multiple GPUs.

We in this paper propose a lightweight neural network to tackle these challenges. Different from existing work, our method equips two competitive leading edges. First, our pose representation coupling with the rotation part and translation part admits the property of smoothness that can improve convergence in learning-based models. Second, we convert the geometry-based pose optimization problem into a data-driven differential optimization layer. In this way, the solution can be regressed by a lightweight neural model that fuses the differential layer into a pre-trained CNN model that is shown in Fig. 1. To further perform the deep pose regression tasks, we also present a novel loss function and gradient descent strategy to supervise the whole network. Extensive experiments on the camera re-localization task on the Cambridge Landmarks and 7-Scenes datasets remarkably demonstrate the superior predictive accuracy and benefits of our method in comparison with the state-of-the-art.

In summary, our work makes the following contributions:

[1] The authors with the Department of Computer Science and Technology, Nanjing University, China. Email: {wenjielee,hw,rdy}@smail.nju.edu.cn, {jialiu,chenlj}@nju.edu.cn.
[2] The author with the School of Computer Science, Hohai University, China. Email: yanyan.wang@hhu.edu.cn.
* Corresponding author

- We propose a lightweight neural network to regress the pose over $SE(3)$. Our pose representation not only admits the property of smoothness making it suitable for learning-based frameworks, but also successfully develop a lightweight neural model integrating with the differential pose layer.
- We supervise the deep pose regression tasks by proposing a novel loss function. In addition, an implicit function theorem is also provided as the backpropagation procedure to optimize the whole networks.
- We conduct extensive experiments on the camera relocalization task on the Cambridge Landmarks dataset and 7-Scenes dataset to show the benefits of our method, the final results demonstrate superior localization accuracy compared with the state-of-the-art.

II. Related Work

A. Pose Estimation

Early related research focuses on geometry-based approaches due to their reliability in some static environments. The majority of these techniques follow the same pipeline: keypoints are extracted using several mathematic tools such as the SURF [11], ORB [12], then a quadratic error function is derived which is generally solved using the Gauss-Newton method [13], Levenberg-Marquardt method [14] and their variants. And several marvelous works have been proposed such as the ORB-SLAM [15], ORB-SLAM2 [16], DVO [17], RSC [18] and GLM [19]. However, these approaches cannot function well in some textureless scenarios. In light of this fact, deep neural networks are performing as an alternative vehicle to regress the pose in recent years. DeepVO [20] brings the deep convolutional network into visual odometry to accurately locate the robot from pair consecutive images. PoseNet [10] introduces the deep neural network to automatically localize the camera from RGB images without manually extracting features. Inspired by these two pioneering works, more fabulous learning-based works have been proposed in this area such as PixLoc [21], DBN [2], GNN-Pose [22], AtLoc [23] and DPVO [24]. However, most learning-based methods seldom consider the fact that the directly regressed pose lying in a Euclidean space is generally not smooth and differential [8], [9], leading to several issues in capturing a more accurate result. Furthermore, a large number of models suffer from high computational overhead due to various deep complicated network designations. In this work, we aim to address this deficiency by presenting a lightweight neural model to regress the pose over $SE(3)$.

B. Pose Parameterization

The pose is traditionally parametrized as elements belonging to $SE(3)$ [25], in which the rotation part is diversely denoted by Euler angles, unit quaternions, or rotation matrices, while the translation part is separately marked as the three-dimensional vector [26]. However, Li et al. [7] pointed out that these representations occasionally incur some issues on the rotation part. For example, Euler angles enduring the singularities and discontinuities may lead to the Gimbal Lock problem [6]. The 4×4 homogeneous transformation matrix is overparameterized with a large of redundancy that may cause memory inefficiency. Unit quaternions are antipodally symmetric but not smooth according to [8]. Additionally, dual quaternions [27] are alternatively provided to model the pose, but similarly bear the nonsmooth feature.

Zhou et al. [8] proposed that a continuous representation of the pose is helpful for training deep neural works, and further demonstrated that a continuous representation is possible if the embedding space is greater than five. In this manner, the 6D and 5D representations have been provided to regress the rotation on $SO(3)$ in learning-based tasks. Besides, another continuous and smooth matrix was proposed to regress the rotation [9] which is parameterized as a ten-dimensional vector. Yet these methods only focus on the smooth representation of the rotation part that make them have difficulty in being used in $SE(3)$.

In this work, we similarly aim to provide a smooth representation of the pose but differ from previous works in three aspects: (1) our smooth solution involving the rotation part and translation part is the first to relax the limitation from $SO(3)$ to $SE(3)$, (2) we show that our representation is smooth and continuous on $SE(3)$ by expanding the concept of *continuity* on $SO(3)$, (3) we introduce a novel loss function and an implicit function theorem to better supervise the whole lightweight pose network.

III. Problem Formulation

In this section, we introduce the necessary geometry-based mathematical formulation of pose estimation over $SE(3)$ using dual quaternions. To fully understand the follow-up sections, we first give a short description of dual quaternions. Then the pose estimation formulation is derived with the form of the quadratically-constrained quadratic program.

A. Dual Quaternion

Dual quaternions comprising two quaternions are viewed as an alternative tool for representing the pose on $SE(3)$ [27]. Before presenting the dual quaternion, it is necessary to give a short introduction to the quaternion.

Quaternion. In this work, a quaternion $\mathbf{q}$ is defined as $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, the $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ is the standard basis of the three-dimensional Euclidean space $\mathbb{R}^3$. For convenience, we bring the vector $\mathbf{q} = [q_0, \mathbf{q}_{vec}] \in \mathbb{R}^4$ to denote the quaternion. The multiplication between two arbitrary quaternions can be done with a matrix-vector form that is given by

$$\mathbf{p} \odot \mathbf{q} = \mathbf{L}_p\mathbf{q} = \mathbf{R}_q\mathbf{p}, \tag{1}$$

where $\mathbf{L}_p$ and $\mathbf{R}_q$ refer to the left side and right side multiplication respectively, which are expressed by

$$\mathbf{L}_p = \begin{bmatrix} p_0 & -\mathbf{p}_{vec}^T \\ \mathbf{p}_{vec} & \mathbf{p}_{vec}^{\times} + p_0\mathbf{I}_3 \end{bmatrix}, \mathbf{R}_q = \begin{bmatrix} q_0 & -\mathbf{q}_{vec}^T \\ \mathbf{q}_{vec} & -\mathbf{q}_{vec}^{\times} + q_0\mathbf{I}_3 \end{bmatrix},$$

where $[\mathbf{a}]^{\times}$ denotes the skew-symmetric matrix formed from the vector $\mathbf{a}$, and $\mathbf{I}$ refers to the identity matrix.

The norm of a quaternion is defined as $\sqrt{\mathbf{q} \odot \mathbf{q}^*}$, with $\mathbf{q}^* = [q_0, -\mathbf{q}_{vec}]$ being the conjugate of $\mathbf{q}$. And quaternions with a unit norm are called unit quaternions, which are used to denote the pure rotation on the unit hypersphere $\mathbb{S}^3 \subset \mathbb{R}^4$.

Dual Quaternion. The formulation of a dual quaternion is given by

$$\mathbf{v} = \mathbf{q}_r + \epsilon \mathbf{q}_d, \epsilon \neq 0, \epsilon^2 = 0, \tag{2}$$

where ϵ refers to the dual number, $\mathbf{q}_r$ is the unit quaternion for indicating the rotation, and $\mathbf{q}_d$ is the dual part quaternion for representing the composition of the rotation quaternion and translation quaternion $\mathbf{q}_t = [0, t_x, t_y, t_z]^T$, with $\mathbf{q}_d = 0.5\mathbf{q}_t \odot \mathbf{q}_r \in \mathbb{R}^4$.

Since the dual part $\mathbf{q}_d$ is orthogonal to the real part $\mathbf{q}_r$ on the hypersphere space $\mathbb{S}^3$, we further get the unit dual quaternion manifold $\mathbb{DH}_1 := \{[\mathbf{q}_r^T, \mathbf{q}_d^T]^T | \|\mathbf{q}_r\| = 1, \mathbf{q}_r \in \mathbb{S}^3, \mathbf{q}_r^T \mathbf{q}_d = 0\} \subset \mathbb{R}^8$. Furthermore, the translation vector $\mathbf{t} = [t_x, t_y, t_z]$ can be recovered from $\mathbb{DH}_1$ according to [7], which can be written as

$$\mathbf{t} = 2[\mathbf{R}_{qr}]_{1:3}^T \mathbf{q}_d, \tag{3}$$

where $[\mathbf{R}_{qr}]_{1:3}$ refers to the last three columns of the right multiplication matrix $\mathbf{R}_{qr}$.

Multiplication of Dual Quaternions. Given two dual quaternions $\mathbf{v}_1$ and $\mathbf{v}_2$, the multiplication is expressed by

$$\mathbf{v}_1 \diamond \mathbf{v}_2 = \mathbf{q}_{r1} \odot \mathbf{q}_{r2} + \epsilon(\mathbf{q}_{r1} \odot \mathbf{q}_{d2} + \mathbf{q}_{d1} \odot \mathbf{q}_{r2}), \tag{4}$$

where $\diamond$ denotes the multiplication of two dual quaternions.

Then we rewrite Eq. (4) with a matrix-vector form,

$$\mathbf{v}_1 \diamond \mathbf{v}_2 = \mathbf{Q}_{\mathbf{v}_1}^L \mathbf{v}_2 = \mathbf{Q}_{\mathbf{v}_2}^R \mathbf{v}_1, \tag{5}$$

where $\mathbf{Q}_{\mathbf{v}_1}^L$ and $\mathbf{Q}_{\mathbf{v}_2}^R$ are left and right multiplication of $\mathbf{v}_1$ and $\mathbf{v}_2$ accordingly, which are defined by

$$\mathbf{Q}_{\mathbf{v}_1}^L = \begin{bmatrix} \mathbf{L}_{\mathbf{q}_{r1}} & \mathbf{0}_{4\times4} \\ \mathbf{L}_{\mathbf{q}_{d1}} & \mathbf{L}_{r1} \end{bmatrix}_{8\times8}, \mathbf{Q}_{\mathbf{v}_2}^R = \begin{bmatrix} \mathbf{R}_{\mathbf{q}_{r2}} & \mathbf{0}_{4\times4} \\ \mathbf{R}_{\mathbf{q}_{d2}} & \mathbf{R}_{\mathbf{q}_{r2}} \end{bmatrix}_{8\times8}.$$

B. Pose Optimization Problem

Many optimization problems are formally formulated as a quadratic cost function, and the pose estimation on $SE(3)$ is no exception. Theoretically, given a set of related vector measurements $\{\mathbf{m}_i, \mathbf{n}_i\}_{i=1}^N \in \mathbb{R}^8$, the general pose transformation from $\mathbf{n}_i$ to $\mathbf{m}_i$ is given by

$$\mathbf{m}_i = \mathbf{v} \diamond \mathbf{n}_i \diamond \mathbf{v}^{-1}. \tag{6}$$

Subsequently, we rewrite Eq. (6) as follows, note that we omit the subscript i for convenience,

$$\begin{aligned} &\mathbf{v} \diamond \mathbf{n} - \mathbf{m} \diamond \mathbf{v} = 0 \\ &\Rightarrow \mathbf{Q}_{\mathbf{n}}^R \mathbf{v} - \mathbf{Q}_{\mathbf{m}}^L \mathbf{v} = 0 \\ &\Rightarrow (\mathbf{Q}_{\mathbf{n}}^R - \mathbf{Q}_{\mathbf{m}}^L)\mathbf{v} = 0 \\ &\Rightarrow \mathbf{v}^T \mathbf{Q} \mathbf{v} = 0, \end{aligned} \tag{7}$$

where the cost weight matrix $\mathbf{Q}$ is symmetric and positive semidefinite [28], with $\mathbf{Q} = (\mathbf{Q}_{\mathbf{n}}^R - \mathbf{Q}_{\mathbf{m}}^L)^T \mathbf{G}^T \mathbf{G} (\mathbf{Q}_{\mathbf{n}}^R - \mathbf{Q}_{\mathbf{m}}^L)$, $\mathbf{G} = diag(\sqrt{g_1}, ..., \sqrt{g_8})$.

However, we cannot strictly ensure Eq. (7) holds due to the existence of the noise. Hence, we in this work aim to convert the pose optimization problem into a Quadratically-Constrained Quadratic Program (QCQP).

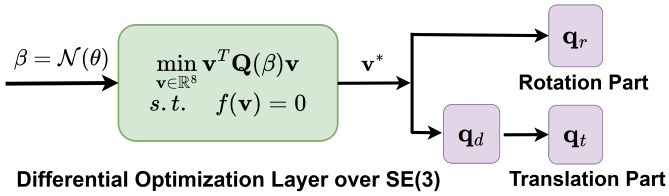

Fig. 2. The framework of proposed differential pose optimization layer. The input vector $\boldsymbol{\beta}$ is regressed by the neural networks $\mathcal{N}(\boldsymbol{\theta})$, and the final pose is predicted from the proposed differential layer via the eigendecomposition.

Definition 3.1 (Pose Optimization as the QCQP): Let matrix $\mathbf{Q} \in \mathbb{R}^{8\times8}$ be a symmetric matrix, the solution of the pose optimization problem is to find the dual quaternion $\mathbf{v} \in \mathbb{R}^8$ satisfies the following formulation

$$\begin{aligned} &\min_{\mathbf{v}} \mathbf{v}^T \mathbf{Q} \mathbf{v} \\ &s.t. \quad f(\mathbf{v}) = \begin{pmatrix} \|\mathbf{q}_r\| - 1 \\ \mathbf{q}_r^T \mathbf{q}_d \end{pmatrix} = 0. \end{aligned} \tag{8}$$

IV. The Analysis of Symmetric Matrix $\mathbf{Q}$

As aforementioned, we formulate the pose optimization with the QCQP problem, where the data matrix $\mathbf{Q}$ constitutes the main challenge. In this section, we provide a rigorous analysis of $\mathbf{Q}$. First, we show how the problem arises as a differential QCQP problem (DQCQP). Then, we introduce the solution to the DQCQP and give the smoothness proof of $\mathbf{Q}$. Finally, the relationship between the DQCQP and deep learning is given.

A. Differential QCQP

Traditionally, a set of algorithms (e.g., Lagrange method [29] and its variants) are well-chosen to solve the Problem (3.1), where the matrix $\mathbf{Q}$ constitutes the main challenge. Conversely, we in this work propose to address this problem by means of a data-driven manner. To this end, a generalized differential QCQP is crucially significant.

Recall that the matrix $\mathbf{Q} \in \mathbb{R}^{8\times8}$ is real symmetric and positive semidefinite, we try to parametrize the matrix $\mathbf{Q}$ using the vector $\boldsymbol{\beta} \in \mathbb{R}^{36}$. In this way, the relationship can be built between the data-driven approach and the QCQP problem by regressing the vector $\boldsymbol{\beta}$.

Definition 4.1 (Differential QCQP): Let $\boldsymbol{\beta} \in \mathbb{R}^{36}$ be a vector predicted from neural networks, the data quadratic matrix $\mathbf{Q}(\boldsymbol{\beta}) \in \mathbb{R}^{8\times8}$ then can be derived and the QCQP problem can be drawn as follows,

$$\begin{aligned} &\min_{\mathbf{v}} \mathbf{v}^T \mathbf{Q}(\boldsymbol{\beta}) \mathbf{v} \\ &s.t. \quad f(\mathbf{v}) = \begin{pmatrix} \|\mathbf{q}_r\| - 1 \\ \mathbf{q}_r^T \mathbf{q}_d \end{pmatrix} = 0. \end{aligned} \tag{9}$$

The Problem (4.1) is issued by choosing a pair of optimal unit dual quaternions $\pm\mathbf{v}^*$[1]. Inspired by [30], we tackle this problem by implementing the eigendecomposition of the real

[1]Due to the antipodal symmetry feature of the unit quaternion $\mathbf{q}_r$, *i.e.*., $\mathbf{q}_r = -\mathbf{q}_r$, we then have $\mathbf{v} = -\mathbf{v}$ for unit dual quaternions.

symmetric matrix $\mathbf{Q}$, and the closed-form solution $\widetilde{\mathbf{v}} \in \mathbb{R}^8$ is shown to be the eigenvector associated with the simple minimal eigenvalue, which can be expressed by

$$\widetilde{\mathbf{v}} = \min_{\lambda_i} eig(\mathbf{Q}), \quad i = 1, 2, ..., 8, \tag{10}$$

where λ_i denotes the eigenvalues of the matrix $\mathbf{Q}$.

However, the directly acquired solution $\widetilde{\mathbf{v}}$ that we call the pseudo-pose from Eq. (10) seems infeasible since it fails to lie on the unit dual quaternion manifold $\mathbb{DH}_1$. In light of the fact, we adopt the strategy $\mathcal{S}(\cdot)$ by normalizing the first four elements $\widetilde{\mathbf{v}}_{1:4}$, and taking an orthogonalization option between $\widetilde{\mathbf{v}}_{1:4}$ and $\widetilde{\mathbf{v}}_{5:8}$. Then the solution $\mathbf{v}^*$ to the Problem (4.1) can be drawn as follows,

$$\pm\mathbf{v}^* = \mathcal{S}(\min_{\lambda_i} eig(\mathbf{Q})), \quad i = 1, 2, ..., 8. \tag{11}$$

and the whole procedure is visually depicted in Fig. 2.

B. A Smooth Representation over SE(3)

In learning-based regression tasks, it is often required to work with numerous representations of the same space, a smooth and differential representation for $SE(3)$ is helpful for training deep neural networks. In this section, we present a smooth pose representation $\mathbf{Q}$ over $SE(3)$.

According to [8], the representation is said to be smooth provided that any intermediate representations in $\mathbb{R}^n$ regressed by the network can be mapped to the original space $\mathbb{X}$, that is, $h : \mathbb{R} \to \mathbb{X}$. Conversely, the mapping from $\mathbb{X}$ to $\mathbb{R}$ holds, $g : \mathbb{X} \to \mathbb{R}$, and for each element $x \in \mathbb{X}$, we have $h(g(x)) = x$, we say that representation is smooth and continuous. In this work, we extend this concept to the $SE(3)$ space.

Remark 4.2: Note that the mapping from real vector[2] $\boldsymbol{\beta} \in \mathbb{R}^{36}$ to the homogeneous transformation matrix $\mathbf{T} \in SE(3)$, $h : \mathbb{R}^{36} \to SE(3)$, where $\mathbf{T}$ can be converted from the unit dual quaternion $\mathbf{v} \in \mathbb{DH}_1$ that is acquired by Eq. (11). We admit that a smooth mapping exists from $SE(3)$ to $\mathbb{R}^{36}$, $g : SE(3) \to \mathbb{R}^{36}$, and have $h(g(\mathbf{T})) = \mathbf{T}$.

C. Relation to Deep Learning

As mentioned above, a smooth representation of $SE(3)$ is beneficial to regressing the pose. In this section, we build the relationship between the smooth representation $\mathbf{Q}$ and deep learning methods. To better predict the target $\mathbf{Q}$, we first introduce a gradient descent method for the optimization of the neural network. Then a well-designed loss function is proposed which is also critical for learning-based tasks.

1) Gradient Descent Strategy: Usually, the neural networks can be viewed as a mapping $\mathcal{N}(\cdot)$ that from the input image data into the target pose, which is expressed by

$$\mathbf{v}^* = \mathcal{N}(\mathbf{Q}(\boldsymbol{\beta}(\boldsymbol{\theta}(\mathbf{I})))), \tag{12}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{36}$ is the vector form of $\mathbf{Q}$, the $\boldsymbol{\theta}$ refers to the parameters of the neural network, and $\mathbf{I}$ denotes the input images.

[2]Literately, the symmetric matrix $\mathbf{Q} \in \mathbb{R}^{8\times8}$ can be simplified by the vector $\boldsymbol{\beta} \in \mathbb{R}^{36}$.

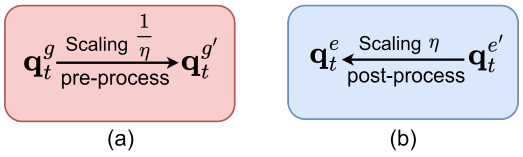

Fig. 3. The pre-process and post-process procedure of translation part. (a) The ground truth translation part is required to scale $1/\eta$ before feeding ino the network. (b) The output translation part from the network is necessary to scale η.

Such a neural network is designed to convert the pose estimation problem into a convex optimization problem (*i.e.* DQCQP layer). Moreover, the DQCQP layer can be seamlessly integrated into any existing deep learning architecture due to its smoothness feature. In this paper, we utilize the lightweight ResNet-18 as our backbone for feature extraction when considering the RGB input mode.

Additionally, to better optimize the designed network, a well-defined gradient descent strategy is of utmost significance. Different to [29], [31] that addressed the quadratic problem by constructing a differential Lagrange layer, we rather in this work introduce the implicit function theorem to build the optimization layer which is inspired by [9], [32]. Regarding the real symmetry of the matrix $\mathbf{Q} \in \mathbb{R}^{8\times8}$, the derivative of $\mathbf{v}^*$ exists with respect to $\boldsymbol{\beta} \in \mathbb{R}^{36}$ if and only if the minimum eigenvalue λ_{min} of $\mathbf{Q}$ is simple, and the gradient is computed as follows,

$$\frac{\partial \mathbf{v}^*}{\partial \boldsymbol{\beta}} = \mathbf{v}^* \otimes (\lambda_{min}\mathbf{I} - \mathbf{Q})^+, \tag{13}$$

where $\otimes$ denotes the Kronecker product, $(\cdot)^+$ refers to the Moore-Penrose pseudo-inverse, and $\boldsymbol{\beta}$ denotes the vector form of $\mathbf{Q}$ which is regressed by the neural network.

2) The Loss Function: Unlike the previous work [10], [33] which directly makes a difference with the rotation and translation parts, respectively. In this part, we introduce a new loss function to supervise the whole training process, which is formulated by

$$\begin{aligned}\mathcal{L}(\mathbf{v}) = \mathcal{L}\left(\begin{bmatrix}\mathbf{q}_r \\ \mathbf{q}_d\end{bmatrix}\right) = &\min(\|\mathbf{q}_r^g - \mathbf{q}_r^e\|_2, \|\mathbf{q}_r^g + \mathbf{q}_r^e\|_2) + \\ &\gamma \min(\|\mathbf{q}_d^g - \mathbf{q}_d^e\|_2, \|\mathbf{q}_d^g + \mathbf{q}_d^e\|_2),\end{aligned} \tag{14}$$

where $\mathbf{q}_r^e$ and $\mathbf{q}_d^e$ are the predicted real part and dual part, while $\mathbf{q}_r^g$ and $\mathbf{q}_d^g$ are the labeled real part and dual part. Furthermore, the weight parameter γ is a trade-off between the difference of the real part and the dual part, accordingly.

V. EXPERIMENTS

In this section, we conduct extensive experiments to demonstrate the accuracy and benefits of our method, which we perform the camera re-localization task on both indoor and outdoor datasets, namely, the Cambridge Landmark [10] and 7-Scenes datasets [34].

We set the ResNet-18 as the backbone where the input images are resized to 224×224, and the parameters of

TABLE I

THE EVALUATION RESULTS ON THE 7-SCENES DATASET IN COMPARISON WITH THE STATE-OF-THE-ART. THE RESULTS ARE MEASURED BY THE MEDIAN TRANSLATION ERROR(M) AND THE MEDIAN ROTATION ERROR(°). THE BEST RESULTS ARE IN **BOLD**.

Scene	Trainable Para.	Chess	Fire	Heads	Office	Pumpkin	RedKitchen	Stairs
PoseNet	5M	0.32m/8.12°	0.47m/14.4°	0.29m/12.0°	0.48m/7.68°	0.47m/8.42°	0.59m/8.64°	0.47m/13.8°
BPN	5M	0.37m/7.24°	0.43m/13.7°	0.31m/12.0°	0.48m/8.04°	0.61m/7.54°	0.58m/7.54°	0.48m/13.1°
Dense PoseNet	5M	0.32m/6.60°	0.47m/14.0°	0.30m/12.2°	0.48m/7.24°	0.49m/8.12°	0.58m/8.34°	0.48m/13.1°
MapNet	63M	0.08m/3.25°	0.27m/11.69°	0.18m/13.2°	0.17m/5.15°	0.22m/4.02°	0.23m/4.93°	0.30m/12.08°
MapNet++	63M	0.10m/**3.17**°	0.20m/9.04°	0.13m/11.1°	0.18m/5.38°	0.19m/**3.92**°	0.20m/5.01°	0.30m/13.4°
GPoseNet	6M	0.20m/7.1°	0.38m/12.3°	0.21m/13.8°	0.28m/8.8°	0.37m/6.92°	0.35m/8.1°	0.37m/12.4°
UBN	1.5M	0.10m/4.97°	0.27m/12.87°	0.12m/14.05°	0.20m/7.52°	0.23m/7.11°	0.19m/8.25°	0.28m/13.1°
MBN-MB	1.5M	0.10m/4.35°	0.28m/11.86°	0.12m/12.76°	0.19m/6.55°	0.22m/6.9°	0.21m/8.08°	0.31m/9.98°
AtLoc	2M	0.10m/ 4.1°	0.25m/11.4°	0.16m/11.8°	0.17m/5.3°	0.21m/4.4°	0.23m/5.4°	0.26m/10.5°
Ours	1M	**0.02m**/4.1°	**0.04m/8.0**°	**0.02m/7.0**°	**0.03m/4.0**°	**0.05m**/4.1°	**0.04m/4.8**°	**0.02m/5.0**°

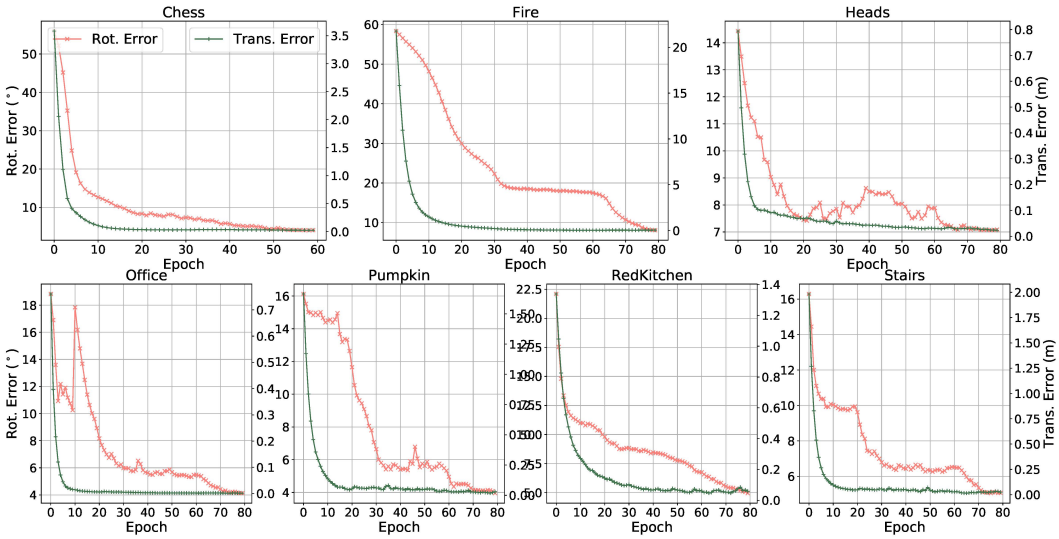

Fig. 4. The rotation and translation median error curves on the 7-Scenes dataset on all 80 epochs. The left y axis denotes the rotation median error, and the right y axis represents the translation median error. Moreover, the red lines are the rotation median errors on all 7 scenes, while the green lines are the translation median errors on all 7 scenes.

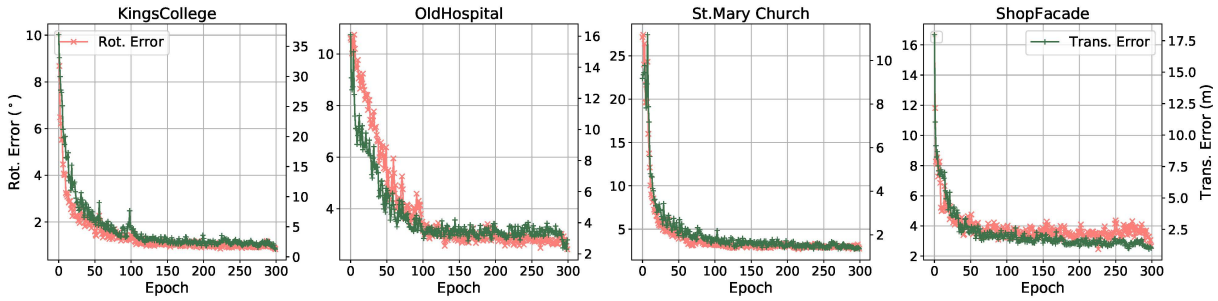

Fig. 5. The rotation and translation median error curves on the Cambridge Landmark dataset on all 300 epochs. The left y axis denotes the rotation median error, and the right y axis represents the translation median error. Moreover, the red lines are the rotation median errors on all 4 scenes, while the green lines are the translation median errors on all 4 scenes.

ResNet are initialized from pre-trained ImageNet weights. Then two fully connected layers are added to replace the ResNet's activations for predicting the vector $\boldsymbol{\beta}$ that afterward is used as the input of the differential QCQP layer shown in Fig. 2, the details of the network structure can be found in Fig. 1. During the training stage, Eq. (13) is applied to the gradient descent optimization. Moreover, we set the trade-off parameter $\gamma = 400$ in the 7-Scenes and $\gamma = 100$ in the Cambridge dataset, accordingly. It is important to note that the learning parameters of the model are confined to two

TABLE II

THE EVALUATION RESULTS ON THE CAMBRIDGE LANDMARKS DATASET IN COMPARISON WITH THE STATE-OF-THE-ART. THE RESULTS ARE MEASURED BY THE MEDIAN TRANSLATION ERROR(M) AND THE MEDIAN ROTATION ERROR(°). THE BEST RESULTS ARE IN **BOLD**.

Scene	Trainable Para.	Kings College	Hospital	ShopFacade	St.Mary Church
PoseNet	5M	1.92m/5.40°	2.31m/5.38°	1.46m/8.08°	2.65m/8.48°
BPN	5M	1.74m/4.06°	2.57m/5.12°	1.25m/7.54°	2.11m/8.38°
Dense PoseNet	5M	1.66m/4.86°	2.62m/4.90°	1.41m/7.18°	2.45m/7.96°
MapNet	63M	1.07m/1.89°	1.94m/3.91°	1.49m/4.22°	2.0m/4.53°
GPoseNet	6M	1.61m/2.30°	2.62m/3.91°	1.14m/5.70°	2.93m/6.50°
UBN	1.5M	0.88m/1.77°	1.93m/3.71°	0.8m/4.74°	1.84m/6.19°
MBN-MB	1.5M	0.83m/2.08°	2.16m/3.64°	0.92m/4.93°	1.37m/6.03°
GNN-Pose	13M	**0.48m**/1.00°	**1.14m**/2.50°	**0.48m/2.50**°	1.52m/3.20°
Ours	1M	1.10m/**0.80**°	1.90m/**2.30**°	0.87m/**2.50**°	**1.30m/2.60**°

fully connected layers and a differential layer. As a result, our model can be efficiently trained on an RTX 2080 GPU platform.

Additionally, the predicted pose $\mathbf{v}^*$ derived from the unit minimum eigenvector $\widetilde{\mathbf{v}}$ in Eq. (11) cannot always be practical since the norm of the translation vector in some scenarios is sometimes considerably large. To address this issue, we propose a scale factor η for the translation vector which can be found in Fig. 3. Specifically, the labeled translation vector $\mathbf{q}_t^g$ encoded by the dual quaternion is required to scale $\frac{1}{\eta}$ its original. Conversely, the final predicted translation vector $\mathbf{q}_t^e$ computed from Eq. (3) is recovered by scaling η. Note that we set $\eta = 300, 200$ in the 7-Scenes dataset and the Cambridge Landmarks dataset respectively.

To demonstrate the accuracy of our lightweight model, we take a comparison with the state-of-the-art including the PoseNet and DensePoseNet [10], MapNet and MapNet++ [35], BPN [36], UBN and MBN-MB [2], AtLoc [23], GPoseNet [37], GNN-Pose [22] in terms of final evaluation results on two common datasets.

A. 7-Scenes Dataset

The model is trained for 80 epochs using a batch size of 8, and the learning rate is 2×10^{-5}. Figure 4 illustrates the inferred median rotation error curve and median translation error curve in 80 epochs, where the left axis denotes the rotation median errors, and the right axis refers to the translation median errors. Obviously, our model is remarkably effective and can converge after 80 epochs. To further exhibit the performance of our method, we then make the comparison with the state-of-the-art that is depicted in Table I, where the best results are denoted by bold font. Our method enables the lowest error on both rotation part and translation part in the scenes of *Fire, Heads, Office, RedKitchen, Stairs*. Besides, our method has a competitive advantage on the translation part in the scenes of *Chess* and *Pumpkin*.

B. The Cambridge Landmark Dataset

Next, we continue the camera re-localization task on the outdoor dataset to test the effectiveness of our approach. We train the proposed network for 300 epochs using a batch size of 8, and the learning rate is 3×10^{-5}. Likewise, we plot the inferred rotation and translation accuracy curves on all scenes. Fig. 5 shows that our method is able to converge after about 150 epochs. To further verify the accuracy of our method, we also provide the comparison with state-of-the-art methods in Table II. Our method performs best on the *Saint Mary Church* with the lowest rotational median error of $2.60°$ and the lowest translational median error of 1.30m. Besides, our method also performs marginally best on the rotation part of the remaining scenes and has a competitive advantage on the translation part. Furthermore, in comparison to existing models in Table I and Table II, our model boasts the smallest number of trainable parameters of about 1M. This efficiency could result in a reduction in the necessary computational resources.

Therefore, the proposed lightweight model is proved to be effective and accurate on the camera re-localization task in the 7-Scenes dataset and the Cambridge Landmark dataset.

VI. CONCLUSIONS

In this paper, we present a lightweight differential neural network to regress the pose over $SE(3)$. First, we present a differential and smooth representation over $SE(3)$ making it suitable for deep pose regression tasks. Unlike existing continuous representations that focus only on the rotation part, our representation provides an alternative vehicle for denoting pose over $SE(3)$ by regressing the vector $\boldsymbol{\beta}$. Then we develop a novel lightweight neural network to convert the pose optimization into a convex optimization problem considering the smoothness of the proposed differential layer. Next, we introduce a novel loss function to supervise the whole pose regression tasks, and an implicit function theorem is provided for the backpropagation procedure. To show the accuracy of our approach, we conduct extensive experiments on the camera re-localization task on the Cambridge Landmarks and 7-Scenes datasets in which final results demonstrate the superior predictive accuracy of our method in comparison with the state-of-the-art.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (Nos. 62072231, 62332013, and 62102079), the Fundamental Research Funds for Central Universities and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual slam algorithms," *Robotics*, vol. 11, no. 1, 2022.

[2] H. Deng, M. Bui, N. Navab, L. J. Guibas, S. Ilic, and T. Birdal, "Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation," *ArXiv*, vol. abs/2012.11002, 2020.

[3] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari, "Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6738–6748.

[4] J. Song, D. Rondao, and N. Aouf, "Deep learning-based spacecraft relative navigation methods: A survey," *Acta Astronautica*, vol. 191, pp. 22–40, 2022.

[5] J. Lucas, T. Kyono, M. Werth, N. Gagnier, Z. Endsley, J. Fletcher, and I. McQuaid, "Estimating satellite orientation through turbulence with deep learning," in *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2020.

[6] E. G. Hemingway and O. M. O'Reilly, "Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments," *Multibody System Dynamics*, vol. 44, no. 1, pp. 31–56, 2018.

[7] K. Li, F. Pfaff, and U. D. Hanebeck, "Unscented dual quaternion particle filter for se(3) estimation," *IEEE Control Systems Letters*, vol. 5, pp. 647–652, 2021.

[8] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5738–5746, 2019.

[9] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly, "A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty," in *Proceedings of Robotics: Science and Systems*, Jul. 12–16 2020.

[10] A. Kendall, M. K. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," *2015 IEEE International Conference on Computer Vision*, pp. 2938–2946, 2015.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 404–417.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[13] J. V. Burke and M. C. Ferris, "A gauss—newton method for convex composite optimization," *Mathematical Programming*, vol. 71, no. 2, pp. 179–194, 1995.

[14] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis: proceedings of the biennial conference held at Dundee, June 28–July 1, 1977*. Springer, 2006, pp. 105–116.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[17] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[18] Y. Dai, H. Li, and L. Kneip, "Rolling shutter camera relative pose: Generalized epipolar geometry," *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4132–4140, 2016.

[19] N. Jiang, Z. Cui, and P. Tan, "A global linear method for camera pose registration," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 481–488.

[20] S. Wang, R. Clark, H. Wen, and A. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, pp. 513 – 542, 2018.

[21] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3247–3257.

[22] M. O. Turkoglu, E. Brachmann, K. Schindler, G. J. Brostow, and A. Monszpart, "Visual camera re-localization using graph neural networks and relative pose supervision," in *2021 International Conference on 3D Vision*. IEEE, 2021, pp. 145–155.

[23] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "Atloc: Attention guided camera localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 06, 2020, pp. 10 393–10 401.

[24] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," *arXiv preprint arXiv:2208.04726*, 2022.

[25] J. McCarthy, *An Introduction to Theoretical Kinematics*. Cambridge,MA: MIT Press, 1990.

[26] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[27] G. Leclercq, P. Lefèvre, and G. Blohm, "3d kinematics using dual quaternions: theory and applications in neuroscience," *Frontiers in behavioral neuroscience*, vol. 7, p. 7, 2013.

[28] M. Horn, T. Wodtko, M. Buchholz, and K. C. J. Dietmayer, "Online extrinsic calibration based on per-sensor ego-motion using dual quaternions," *IEEE Robotics and Automation Letters*, vol. 6, pp. 982–989, 2021.

[29] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 136–145.

[30] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[31] S. Gould, R. Hartley, and D. Campbell, "Deep declarative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 3988–4004, 2022.

[32] J. R. Magnus, "On differentiating eigenvalues and eigenvectors," *Journal of Optimization Theory and Applications*, 1985.

[33] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," in *Conference on Robot Learning*. PMLR, 2021, pp. 1761–1772.

[34] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013.

[35] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, "Geometry-aware learning of maps for camera localization," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2616–2625.

[36] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation*, 2016, pp. 4762–4769.

[37] M. Cai, C. Shen, and I. D. Reid, "A hybrid probabilistic model for camera relocalization," in *British Machine Vision Conference*, 2018.