2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Consistency-Based Loss for Deep Odometry Through Uncertainty Propagation

Hamed Damirchi[1], Roohollah Khorrambakht[2], Hamid D. Taghirad[1] *Senior Member, IEEE*
and Behzad Moshiri[3] *Senior Member, IEEE,*

***Abstract*— Conventionally, deep odometry networks use objective functions that only penalize short-term deviations from the true path. Since such an objective does not impose any constraints on the long-term deviations from the path, a second consistency-based loss term may be added to lower long-term drift. However, maintaining a balance between the two loss terms is challenging and often treated as a design hyperparameter. To mitigate this balancing issue, we propose to use the uncertainty over both odometry and the long-term transformations in a maximum likelihood setting and allow the network to tune the weighting between the two loss terms. To this end, we derive the odometry uncertainty alongside the pose outputs using the network itself and to derive the covariance matrix over the integrated transformation, we propose to propagate the odometry uncertainty through each iteration. This formulation provides an adaptive and statistically consistent method to weigh the incremental and integrated loss terms against each other, noting the increase in uncertainty as more steps are integrated over. We show that our approach to consistency-based losses allows the network to surpass the accuracy of the state-of-the-art visual odometry approaches. Then, the efficacy of the derived uncertainty as weighting medium is visualized and the performance benefits of uncertainty quantification are shown in a pose-graph based localization scenario.**

I. INTRODUCTION

Odometry refers to the incremental localization of a device using sensors such as cameras, IMUs, radars, etc. This method of localization has been used in both single–modal [1] and multi–modal [2] settings in various fields such as robotics [3], self-driving vehicles [4] and planetary exploration rovers [5]. Over the last decade, due to the increase in utilization of such pipelines in everyday applications, the necessity of uncertainty communication has increased for safety and reliability reasons [6]. The benefits of uncertainty quantification are not limited to uncertainty communication. In classical pose-graph based localization methods, the odometry estimates are used as constraints in between nodes of a Bayesian network where each node represents the location of the device. Although each edge is commonly given a constant covariance matrix or uses photometric errors as a heuristic for uncertainty, it has been shown [7], that estimating an uncertainty for each of the edges allows for a considerable improvement over the accuracy of the pose estimation pipeline.

[1] Advanced Robotics and Automated Systems, Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran.
[2]Department of Electrical and Computer Engineering, New York University, New York, USA.
[3]School of ECE, University College of Engineering, University of Tehran, Tehran, Iran.

Deep learning has shown to be an adequate method of learning representations from which uncertainty about a particular output can be estimated [6]. Kendall and Gal [8], categorized the total uncertainty of a network about an output into aleatoric and epistemic uncertainties where the aleatory variability of the output corresponds to the heteroscedastic noise in the data. The epistemic uncertainty is the result of imperfect training data and describes the confidence of the model about it's knowledge of a certain data point. Therefore, epistemic uncertainty can be reduced by providing the model with more task representative data, whereas uncertainties are categorized as aleatory if the model cannot reduce them using more training data. Pragmatically, Gal and Ghahramani [9] used dropout variational inference to calculate the epistemic uncertainty about the output of the network and Kendall and Gal [8] derive the aleatoric uncertainty about a datapoint through the network itself and propose to incorporate the estimated covariance matrix within a maximum likelihood setting. Finally, the total uncertainty is calculated by summing the aleatory and epistemic uncertainties together.

Although estimating the uncertainty about the pose output from an odometry network has been formulated both in end-to-end and hybrid systems, no long-term constraints are imposed on the networks trained to deliver the uncertainty estimates. In the current literature of deep odometry, the works that make use of uncertainty [10], do not consider long-term deviation from the true path leading to lower accuracy in the long run. However, the works that consider long-term consistency [11], do not use uncertainty to balance the short-term and long-term losses leading to the requirement for rigorous manual tuning of the weighting between loss terms. In this paper, we propose to use the uncertainty over the inferred odometry and the integrated odometry transformations in order to balance the short-term and long-term constraints in the objective function.

An overview of our approach is shown in Fig. 1. We implement our proposed method in a deep Visual Odometry (VO) setting where at each iteration of the algorithm, a probability distribution is first inferred over the incremental changes in the pose of the camera using a CNN-LSTM model with a pair of consecutive images of fixed size as input (blue dashed lines in Fig. 1). Then, each consecutive pair of odometry transformations are compounded and the integrated covariance matrix is computed (green dashed lines in Fig. 1). By repeating this process, we obtain the uncertainty over the integrated transformation over multiple steps which are then used in a long-term consistency constraint to mitigate the bal-

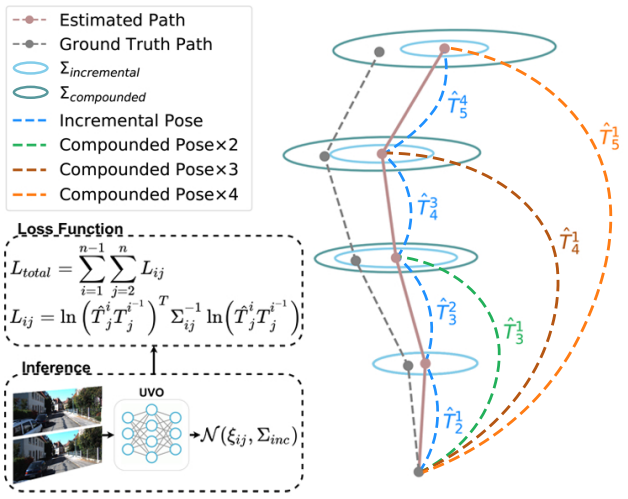


Fig. 1. An overview of the proposed method. The incremental and compounded uncertainties are visualized by projecting the covariance ellipsoid onto a 2-D plane. Consistency-based loss terms are formed using the propagated covariance matrices. Note that the overlapping windows of integration for each window size are not shown for clarity.

ancing issues between long-term and short-term losses. We compare our results against the current classical and learning-based state-of-the-art (SOTA) methods while outperforming recent work on both categories. Thereafter, we provide an in-depth analysis of the effects of the resulting covariance matrices as loss weighting medium. Finally, we utilize a loop detection algorithm to demonstrate the effectiveness of estimated odometry uncertainties in a pose-graph setup. To the best of our knowledge, propagation of uncertainty has not been proposed as a part of the loss function of an odometry network and this is the first time an approach takes accumulation of uncertainty into account in such a setting. Briefly, our contributions are as follows:

- We propose a consistency-based loss function for deep odometry algorithms based on uncertainty compounding and provide quantitative comparisons while outperforming the SOTA,
- Rigorous analysis on the effect of the compounded term on the loss value is provided,
- We embed our method into a pose-graph alongside a loop closure detection algorithm to showcase the importance of the uncertainties estimated by the network in a hybrid localization system.

II. RELATED WORKS

From an algorithmic perspective, uncertainty in odometry has been proposed in standalone deep learning [7] and hybrid algorithms [12]. Regardless of the uncertainty quantification formulation, deep learning based methods commonly take a maximum-likelihood approach to bypass the need for labels for the covariance matrix at each step. Alternatively in hybrid cases, deep learning based uncertainty estimation is utilized to estimate the error distribution of classical VO systems or used in a tightly coupled state estimation scenario [13]. We briefly discuss both categories in this section.

DeepVO [1], was the first work to formulate VO in an end-to-end fashion. This network computes the odometry without considering the long-term consistency issues and uncertainty surrounding the estimated pose. This work was later extended to ESP-VO [7] to account for the frame-to-frame uncertainties of the output poses. However, this work does not take the increase in the uncertainty of poses into account while imposing a global constraint.

CL-VO [11], proposes to integrate the odometry estimates to create a consistency-based loss term. This work does not associate uncertainty with the output poses. Due to the lack of adaptive weighting parameters for the loss terms, [11] requires manual tuning of the loss functions. Moreover, the proposed loss function in CLVO uses a handcrafted scheduling system to determine when to include the long-term error in the overall loss. In our work, apart from associating uncertainty with each output, we also propagate the uncertainty to weigh the global loss term, eliminating the need for loss tuning or scheduling.

UA-VO [10], uses a conventional CNN-LSTM architecture to estimate the odometry poses alongside their uncertainty. This work extends the previous works by including the epistemic uncertainty of the network during inference through calculation of the predictive uncertainty. UA-VO does not take the long-term consistency issues into account.

Deep Inference for Covariance Estimation (DICE) [12], infers the error distribution of an arbitrary classical odometry method using a CNN that takes as input a single image from the pair that was passed to the VO pipeline. Deeper-Dice [14], extends this method by adding the corrections from the network estimates to the VO output before modeling their distribution to account for the biases of the VO outputs. Our method does not require a separate classical pipeline to estimate the odometry and we infer the odometry itself alongside the covariance matrix using a single network.

III. PROPOSED APPROACH AND ARCHITECTURE

In this section, we first associate uncertainty with the output of the network. Then, the uncertainty compounding formulation and our loss function will be proposed. Finally, the uncertainty quantification formulation using parametric methods such as neural networks will be discussed and the architectural details of the network will be provided.

A. Odometry Uncertainty

There are several works on the association of uncertainty with pose vectors [15]–[17]. In this paper, we adopt the vector space of the SE(3) group as the output of the visual odometry network and define a PDF on this vector space which in turn allows us to induce uncertainty on the SE(3) matrices through the exponential mapping. To this end, we use noisy perturbations [17] to associate uncertainty with SE(3) matrices as follows.

$$\mathbf{T} = e^{\boldsymbol{\xi}^p} \bar{\mathbf{T}}, \boldsymbol{\xi}^p \in \mathbb{R}^6 . \tag{1}$$

Where $\bar{\mathbf{T}}$ represents the mean transformation matrix and is estimated by the network using a pair of images as input.

Meanwhile, $\boldsymbol{\xi}^p$ is the noisy perturbation in the form of a Lie algebra vector that defines the uncertainty over the camera movement as a Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}$ as follows.

$$p(\boldsymbol{\xi}^p) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \boldsymbol{\Sigma} \in \mathbb{R}^{6\times6} \tag{2}$$

The $\boldsymbol{\Sigma}$ matrix is estimated by the network alongside the pose the details of which will be presented in Section III-D.2.

B. Uncertainty Compounding

In the previous section, we provided the formulation for representing a distribution over the estimated transformation from a visual odometry network. In this section, we will use this formulation to derive the equation for calculating the propagated uncertainty over a sequence of images. To perform integration while propagating the incremental uncertainty, we use the definition from (1) on a setup where the network has estimated the odometry over two timesteps as follows.

$$e^{\boldsymbol{\xi}^p_{02}}\bar{\mathbf{T}}^{i-2}_i = e^{\boldsymbol{\xi}^p_{12}}\bar{\mathbf{T}}^{i-2}_{i-1}\, e^{\boldsymbol{\xi}^p_{01}}\bar{\mathbf{T}}^{i-1}_i \tag{3}$$

Where $e^{\boldsymbol{\xi}^p_{01}}\bar{\mathbf{T}}^{i-1}_i$ and $e^{\boldsymbol{\xi}^p_{12}}\bar{\mathbf{T}}^{i-2}_{i-1}$ represent the consecutive outputs from the network in 2 timesteps over a trajectory. In particular, $\bar{\mathbf{T}}^{i-1}_i$ represents the pose of the camera at timestep i with respect to a frame placed at the position of the device for image frame $i-1$. $\boldsymbol{\xi}^p_{01}$ is the corresponding perturbation that depicts the uncertainty over the estimated pose change between frames i and $i-1$. Moreover, $\bar{\mathbf{T}}^{i-2}_i$ represents the mean of the integrated transformation matrix with the compounded uncertainty $e^{\boldsymbol{\xi}^p_{02}}$ in the form of a noisy perturbation. To derive the formulation for calculating $e^{\boldsymbol{\xi}^p_{02}}$, we use the Baker-Campbell-Hausdorff (BCH) formula following [17], to which we refer the reader for a full interpretation. To solve (3) for the compounded covariance matrix, we move the perturbation factors to the left hand side of $\bar{\mathbf{T}}^{i-2}_{i-1}$ to get

$$e^{(\boldsymbol{\xi}^{p\wedge}_{02})} = e^{(\boldsymbol{\xi}^{p\wedge}_{12})}e^{(\overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}\boldsymbol{\xi}^p_{01})^\wedge} \tag{4}$$

where $\overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}$ is the adjoint of the SE(3) matrix $\bar{\mathbf{T}}^{i-2}_{i-1}$ and the wedge ($^\wedge$) operator is defined below

$$\boldsymbol{\xi}^{p\wedge} = \begin{bmatrix}\boldsymbol{\rho}^p\\ \boldsymbol{\phi}^p\end{bmatrix}^\wedge = \begin{bmatrix}\boldsymbol{\phi}^{p\wedge} & \boldsymbol{\rho}^p\\ \mathbf{0}_{1\times3} & 0\end{bmatrix}, \boldsymbol{\phi}^{p\wedge} = \begin{bmatrix}0 & -\phi^p_z & \phi^p_y\\ \phi^p_z & 0 & -\phi^p_x\\ -\phi^p_y & \phi^p_x & 0\end{bmatrix} \tag{5}$$

Where $\rho^p = [\rho^p_x, \rho^p_y, \rho^p_z]^T$ and $\phi^p = [\phi^p_x, \phi^p_y, \phi^p_z]^T$ represent the translation and rotation component of $\boldsymbol{\xi}^p$, respectively. By using the BCH formula on (4) while noting $\mathbb{E}[\boldsymbol{\xi}^p_{ij}] = 0$ for any i and j, we may derive the compounded covariance matrix as follows

$$\begin{aligned}\boldsymbol{\Sigma}_{02} = \mathbb{E}[\boldsymbol{\xi}^p_{02}\boldsymbol{\xi}^{p^T}_{02}] = \mathbb{E}[&\boldsymbol{\xi}^p_{12}\boldsymbol{\xi}^{p^T}_{12} + \boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'^T}_{01}\\ &+\frac{1}{12}((\boldsymbol{\xi}^{p^\wedge}_{12}\boldsymbol{\xi}^{p^\wedge}_{12})(\boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'^T}_{01}) + (\boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'T}_{01})(\boldsymbol{\xi}^{p^\wedge}_{12}\boldsymbol{\xi}^{p^{\wedge T}}_{12})\\ &\quad+(\boldsymbol{\xi}^{p'^\wedge}_{01}\boldsymbol{\xi}^{p'^\wedge}_{01})(\boldsymbol{\xi}^p_{12}\boldsymbol{\xi}^{p^T}_{12}) + (\boldsymbol{\xi}^p_{12}\boldsymbol{\xi}^{p^T}_{12})(\boldsymbol{\xi}^{p'^\wedge}_{01}\boldsymbol{\xi}^{p'^\wedge}_{01})^T)\\ &+\frac{1}{4}(\boldsymbol{\xi}^{p^\wedge}_{12}(\boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'^T}_{01})\boldsymbol{\xi}^{p^{\wedge T}}_{12})]\end{aligned} \tag{6}$$

where $\boldsymbol{\Sigma}_{02}$ is the compounded covariance matrix and $\boldsymbol{\xi}^{p'}_{01} = \overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}\boldsymbol{\xi}^p_{01}$. The curly wedge operation ($^\curlywedge$) is defined as

$$\boldsymbol{\xi}^{p^\curlywedge} = \begin{bmatrix}\boldsymbol{\rho}^p\\ \boldsymbol{\phi}^p\end{bmatrix}^\curlywedge = \begin{bmatrix}\boldsymbol{\phi}^{p^\wedge} & \boldsymbol{\rho}^{p^\wedge}\\ \mathbf{0}_{3\times3} & \boldsymbol{\phi}^{p^\wedge}\end{bmatrix} \in \mathbb{R}^{6\times6}. \tag{7}$$

Equation (6) may be broken down and written as

$$\mathbb{E}[\boldsymbol{\xi}^p_{12}\boldsymbol{\xi}^{p^T}_{12}] = \boldsymbol{\Sigma}_{12} \tag{8}$$

$$\begin{aligned}\mathbb{E}[\boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'^T}_{01}] = \boldsymbol{\Sigma}'_{01} &= \mathbb{E}[\overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}\boldsymbol{\xi}^p_{01}\boldsymbol{\xi}^{p^T}_{01}\overline{\boldsymbol{\mathcal{T}}}^{i-2^T}_{i-1}]\\ &= \overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}\boldsymbol{\Sigma}_{01}\overline{\boldsymbol{\mathcal{T}}}^{i-2^T}_{i-1}\end{aligned} \tag{9}$$

$$\begin{aligned}\mathbb{E}[\boldsymbol{\xi}^{p^\curlywedge}_{12}\boldsymbol{\xi}^{p^\curlywedge}_{12}] &= \mathbb{E}\begin{bmatrix}\boldsymbol{\phi}^{p^\wedge}_{12}\boldsymbol{\phi}^{p^\wedge}_{12} & \boldsymbol{\phi}^{p^\wedge}_{12}\boldsymbol{\rho}^{p^\wedge}_{12} + \boldsymbol{\rho}^{p^\wedge}_{12}\boldsymbol{\phi}^{p^\wedge}_{12}\\ 0 & \boldsymbol{\phi}^{p^\wedge}_{12}\boldsymbol{\phi}^{p^\wedge}_{12}\end{bmatrix}\\ &= \begin{bmatrix}(\boldsymbol{\Sigma}^{\phi\phi}_{12})^* & (\boldsymbol{\Sigma}^{\rho\phi}_{12} + \boldsymbol{\Sigma}^{\rho\phi^T}_{12})^*\\ \mathbf{0}_{3\times3} & (\boldsymbol{\Sigma}^{\phi\phi}_{12})^*\end{bmatrix}\end{aligned} \tag{10}$$

$$\mathbb{E}[\boldsymbol{\xi}^{p'^\curlywedge}_{01}\boldsymbol{\xi}^{p'^\curlywedge}_{01}] = \begin{bmatrix}(\boldsymbol{\Sigma}^{\phi\phi'}_{01})^* & (\boldsymbol{\Sigma}^{\rho\phi'}_{01} + \boldsymbol{\Sigma}^{\rho\phi'^T}_{01})^*\\ \mathbf{0}_{3\times3} & (\boldsymbol{\Sigma}^{\phi\phi'}_{01})^*\end{bmatrix} \tag{11}$$

$$\mathbb{E}[\boldsymbol{\xi}^{p^\curlywedge}_{12}(\boldsymbol{\xi}^{p'}_{01}\boldsymbol{\xi}^{p'^T}_{01})\boldsymbol{\xi}^{p^{\curlywedge T}}_{12}] = \begin{bmatrix}\mathcal{B}_{11} & \mathcal{B}_{12}\\ \mathcal{B}_{21} & \mathcal{B}_{22}\end{bmatrix} \tag{12}$$

$$\begin{aligned}\mathcal{B}_{11} = &(\boldsymbol{\Sigma}^{\phi\phi}_{12}, \boldsymbol{\Sigma}^{\rho\rho'}_{01})^* + (\boldsymbol{\Sigma}^{\rho\phi^T}_{12}, \boldsymbol{\Sigma}^{\rho\phi'}_{01})^*\\ &+ (\boldsymbol{\Sigma}^{\rho\phi}_{12}, \boldsymbol{\Sigma}^{\rho\phi'^T}_{01})^* + (\boldsymbol{\Sigma}^{\rho\rho}_{12}, \boldsymbol{\Sigma}^{\phi\phi'}_{01})^*\end{aligned} \tag{13}$$

$$\mathcal{B}_{12} = (\boldsymbol{\Sigma}^{\phi\phi}_{12}, \boldsymbol{\Sigma}^{\rho\phi'^T}_{01})^* + (\boldsymbol{\Sigma}^{\rho\phi^T}_{12}, \boldsymbol{\Sigma}^{\phi\phi'}_{01})^* \tag{14}$$

$$\mathcal{B}_{21} = \mathcal{B}^T_{12} \tag{15}$$

$$\mathcal{B}_{22} = (\boldsymbol{\Sigma}^{\phi\phi}_{12}, \boldsymbol{\Sigma}^{\phi\phi'}_{01})^* \tag{16}$$

where $\mathbf{A}^* = -tr(\mathbf{A})\mathbf{1} + \mathbf{A}$ and $(\mathbf{A}, \mathbf{B})^* = \mathbf{A}^*\mathbf{B}^* + (\mathbf{B}\mathbf{A})^*$. To derive (10), we use (7) followed by the identity $\mathbf{u}^\wedge_1\mathbf{u}^\wedge_2 = -(\mathbf{u}^T_1\mathbf{u}_2)\mathbf{1} + \mathbf{u}_2\mathbf{u}^T_1, \mathbf{u}_1, \mathbf{u_2} \in \mathbb{R}^3$. To get (11) we use $\boldsymbol{\xi}^p_{01} = \overline{\boldsymbol{\mathcal{T}}}^{i-2}_{i-1}\boldsymbol{\xi}^p_{01}$ alongside $(\boldsymbol{\mathcal{T}}\boldsymbol{\xi})^\curlywedge = \boldsymbol{\mathcal{T}}\boldsymbol{\xi}^\curlywedge\boldsymbol{\mathcal{T}}^{-1}$. Finally to derive (12)-(16), the following identity is used [18]

$$\begin{aligned}\mathbf{u}^\wedge\mathbf{A}\mathbf{v}^\wedge \equiv &\left(-\operatorname{tr}\left(\mathbf{v}\mathbf{u}^T\right)\mathbf{1} + \mathbf{v}\mathbf{u}^T\right) \times \left(-\operatorname{tr}(\mathbf{A})\mathbf{1} + \mathbf{A}^T\right)\\ &+ \operatorname{tr}\left(\mathbf{A}^T\mathbf{v}\mathbf{u}^T\right)\mathbf{1} - \mathbf{A}^T\mathbf{v}\mathbf{u}^T\end{aligned} \tag{17}$$

Where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3, \mathbf{A} \in \mathbb{R}^{3\times3}$. Overall, (8)-(16) can be used to calculate the compounded uncertainty $\boldsymbol{\Sigma}_{02}$ while the mean value of the compounded pose may be found through $\bar{\mathbf{T}}^{i-2}_i = \bar{\mathbf{T}}^{i-2}_{i-1}\bar{\mathbf{T}}^{i-1}_i$.

C. Loss Function

In this section, to derive our objective function, we factorize a likelihood over the estimated odometry and their integration. For the problem with two odometry outputs defined in (3), we have the following factorization

$$\begin{aligned}p(\boldsymbol{\xi}_{12}, \boldsymbol{\xi}_{01}, \boldsymbol{\xi}_{02}) = &\, p(\boldsymbol{\xi}_{12} \mid \mathbf{f}_{\boldsymbol{\theta}}(I_{1,2}))\\ &\times p(\boldsymbol{\xi}_{01} \mid \mathbf{f}_{\boldsymbol{\theta}}(I_{2,3}))\\ &\times p(\boldsymbol{\xi}_{02} \mid \boldsymbol{\xi}_{12}, \boldsymbol{\xi}_{01})\end{aligned} \tag{18}$$

Where I_i represents the captured frame at iteration i and $\boldsymbol{\xi}_{i,i-1} = \log(\mathbf{T}^{i-1}_i) = \log(\boldsymbol{\xi}^p_{i,i-1}\bar{\mathbf{T}}^{i-1}_i)$ represents the network estimates. Precisely, we use the network to estimate frame-to-frame transformation $\log(\bar{\mathbf{T}}^{i-1}_i)$ and the covariance

matrix associated with $\boldsymbol{\xi}^{p}_{i,i-1}$ as the outputs. Meanwhile, $\mathbf{f}_{\boldsymbol{\theta}}$ is the function representing our network with parameters $\boldsymbol{\theta}$. The negative log likelihood of (18) derives the loss minimization objective.

$$
\begin{aligned}
-\log\ p(\boldsymbol{\xi}_{12}, \boldsymbol{\xi}_{01}, \boldsymbol{\xi}_{02}) = \\
\log(e^{\boldsymbol{\xi}_{12}} \mathbf{T}_{i-1}^{i-2^{*^{-1}}})^{T} \boldsymbol{\Sigma}_{12}^{-1} \log(e^{\boldsymbol{\xi}_{12}} \mathbf{T}_{i-1}^{i-2^{*^{-1}}}) \quad (19) \\
+ \log(e^{\boldsymbol{\xi}_{01}} \mathbf{T}_{i}^{i-1^{*^{-1}}})^{T} \boldsymbol{\Sigma}_{01}^{-1} \log(e^{\boldsymbol{\xi}_{01}} \mathbf{T}_{i}^{i-1^{*^{-1}}}) \quad (20) \\
+ \log(e^{\boldsymbol{\xi}_{02}} \mathbf{T}_{i}^{i-2^{*^{-1}}})^{T} \boldsymbol{\Sigma}_{02}^{-1} \log(e^{\boldsymbol{\xi}_{02}} \mathbf{T}_{i}^{i-2^{*^{-1}}}) \quad (21) \\
+ \log(|\boldsymbol{\Sigma}_{12}|) + \log(|\boldsymbol{\Sigma}_{01}|) + \log(|\boldsymbol{\Sigma}_{02}|) \quad (22)
\end{aligned}
$$

Where $\mathbf{T}^{*}$ represents the ground truth pose. Equations (19) and (20) represent the geodesic distance between the estimated odometry and ground truth poses (frame-to-frame deviation) where each term is weighted by the covariance matrix estimated at the corresponding iteration by the network. Equation (21) represents the compounded loss term where the distance between the integrated poses and the ground truth is used as the long-term loss while being weighted by the propagated covariance matrix computed through (6).

Therefore, in the case of the losses imposed on the odometry outputs, if the network is not able to estimate odometry accurately, it can increase the uncertainty output to lower the amount of loss. However, the three terms in (22) would then act as regularizers and punish large uncertainties to create an overall balance. In case of the global loss term, if at a certain iteration along the trajectory, a pair of input frames result in a peak over the pose uncertainty (the network was not able to estimate the output accurately) the propagated uncertainty will substantially increase during the compounding process and the integrated loss will be adaptively weighted. Therefore, uncertainty estimation allows us to weigh the loss on each axis while also providing a principled way to balance the short-term and long-term losses against each other.

D. Implementation Details

In this section we provide the details of the uncertainty quantification algorithm and the architecture of our network.

1) Network Architecture: We use a CNN-LSTM architecture to derive a spatio-temporal model of the consecutive inputs. As can be seen in Fig. 2 we use the encoder section of FlowNetS [19] to derive the visual features from a pair of input frames. The visual features are then converted into a vector using global average pooling. The averaged features are then passed through two layers of Long-Short Term Memory networks to model the features temporally. Thereafter, two fully connected layers (not shown in Fig. 2) are used to estimate the output pose and uncertainty.

2) Uncertainty Quantification: The uncertainty quantification formulation should be constrained in such a way that the resulting matrix would be semi-positive definite. To this end, We process the 6 odometry uncertainty outputs into diagonal elements of the covariance matrix through $\sigma_i^2 = \exp(s_i)$ where $s_i = \log \sigma^2$ is estimated by the network. To calculate $\log(|\boldsymbol{\Sigma}_{12}|)$ and $\log(|\boldsymbol{\Sigma}_{01}|)$ from (22), the following equation may be used.

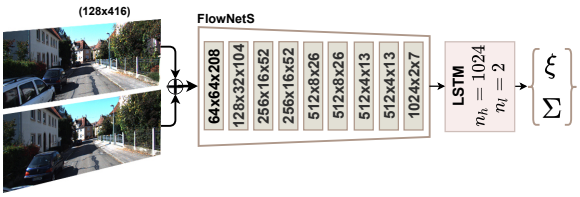


Fig. 2. Architecture of our VO network. The images are resized, concatenated along the channel dimension and passed to the network for processing.

$$
\log|\boldsymbol{\Sigma}| = \log(\prod_{i=1}^{n=6} \sigma_i^2) = \sum_{i=1}^{n=6} \log(\sigma_i^2) = \sum_{i=1}^{n=6} (\mathrm{s}_i) \quad (23)
$$

However, the term $\log(|\boldsymbol{\Sigma}_{02}|)$ regarding the compounded loss term in (22) is no longer diagonal due to the compounding, and (23) cannot be used to calculate this term. To this end, we take the Cholesky factorization of the estimated covariance matrix and calculate $\log(|\boldsymbol{\Sigma}_{02}|)$ as follows

$$
\log|\boldsymbol{\Sigma}| = \log(|\mathbf{L}\mathbf{L}^{T}|) = 2\log(|\mathbf{L}|) = 2\sum_{i=1}^{n=6} (\log \mathbf{L}_{ii}) \quad (24)
$$

where $\mathbf{L}$ is the lower triangular matrix resulting from Cholesky factorization of $\log(|\boldsymbol{\Sigma}_{02}|)$.

IV. Experiments and Analysis

We perform all the experiments on an NVIDIA P100 GPU using PyTorch and PyTorch lightning. While training, we use short segments of the training sequences with lengths of 32. The windows over which output poses are compounded have a maximum length of 5 while a batch size of 16 is used during training. Moreover, we have open–sourced our code for reproducibility purposes[1]. In the following, we discuss the dataset used for all our analyses alongside the approaches against which we compare our method.

A. Dataset and Evaluation

We use the KITTI odometry dataset to perform our experiments. This dataset consists of 22 sequences of driving a car in residential areas. We use sequences 00-07 to train and validate our network and perform tests using sequences 08-10. To quantitatively evaluate our network we use the KITTI odometry benchmark [22], where the relative translation and rotation errors of output poses are computed over segments with lengths of 100m-800m.

B. Comparisons

We compare our results against both classical and deep learning based odometry methods on the KITTI dataset. To compare with the classical methods we chose DSO [20], a SOTA direct odometry approach and the monocular variant of ORB-SLAM2 [21] as a well-known SOTA indirect odometry method. To compare against deep learning based approaches, we chose UA-VO [10], ESP-VO [7], DeepVO [1]

[1]The code will be available upon acceptance

TABLE I
QUANTITATIVE ANALYSIS

Sequence	DSO [20] t(%)/r(°)	ORB-SLAM2 [21] t(%)/r(°)	DeepVO [1] t(%)/r(°)	CLVO [11] t(%)/r(°)	ESPVO [7] t(%)/r(°)	UA-VO [10] t(%)/r(°)	UVO (ours) t(%)/r(°)
08	49.2(29.7)/**0.44**	57.2(13.7)/0.46	9.06/2.64	8.84/2.88	11.60/4.27	9.68(7.91)/3.82(2.76)	**5.12**(4.93)/**1.35**
09	67.6(17.1)/**0.52**	72.0(3.32)/0.84	10.6/4.21	8.83/3.54	11.28/3.22	10.2(11.9)/4.29(3.15)	**8.31**(7.61)/**2.63**
10	77.3(6.68)/1.43	83.0(5.51)/**0.51**	15.8/4.14	14.5/3.90	12.66/4.32	11.1(10.3)/3.86(3.49)	**10.5**(7.90)/**2.91**
Avg.	64.7(17.8)/0.80	70.7(7.51)/**0.60**	11.8/3.66	10.72/3.44	11.85/3.94	9.95(10.0)/3.93(3.13)	**7.98**(6.81)/**2.30**

and CLVO [11]. UA-VO is the current SOTA for uncertainty-based odometry approaches. The loss function proposed in this method does not include a global term that would take long-term deviations into account. ESP-VO and CLVO both include a compounding term in their loss function but do not make use of uncertainty to weigh the losses in a principled way. Finally, DeepVO is the SOTA odometry method that does not make use of uncertainty nor a global loss term.

C. Quantitative Analysis

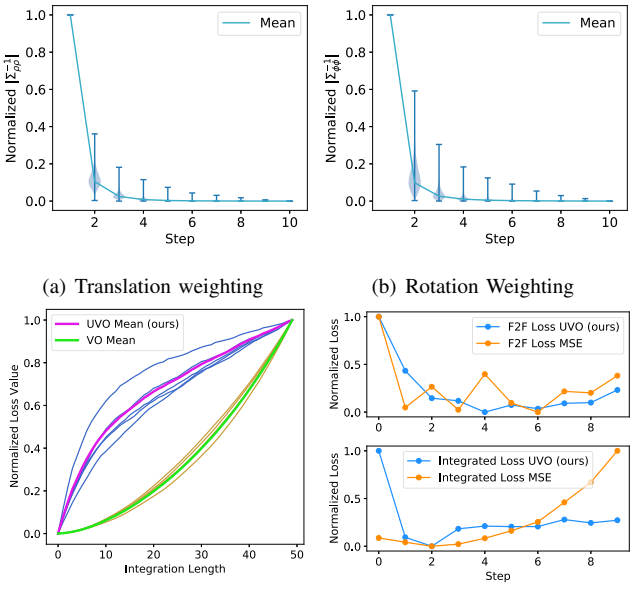
The quantitative analysis of our method is provided in Table I alongside the competing classical and deep learning based approaches. The results for the SOTA deep learning based method termed UA-VO are reported from [10]. Furthermore, The values inside the parentheses for this method represent the results of our re-implementation of UA-VO. Due to a lack of open-source code for DeepVO, CLVO and ESPVO, we implemented them based on [1], [7], [11].

Compared to deep learning based approaches, our method achieves a significantly higher accuracy both in terms of individual sequences and the overall mean. In particular, UVO obtains a 19.8% increase in translation and 41.5% in rotation accuracy over UA-VO. Among other deep learning based methods, our method achieves an increase of 32.4% over translation and 37.1% over rotation accuracy compared to DeepVO which shows the benefits of using uncertainty-based losses. Although CLVO does include a compositional loss term, the lack of adequate weighting results in a diminished accuracy compared to our approach. On the other hand, even though ESPVO does associate uncertainty with frame-to-frame outputs, the lack of such a weighting mechanism on the integrated poses degrades the performance of this network. We note that the entries for our method on Table I are the mean of top-3 runs and the range of translation and rotation accuracy over 10 runs are $8.57\%\pm0.8$ and $2.3^\circ\pm0.6$, respectively.

Compared to classical approaches, our approach outperforms DSO and ORBSLAM2 in terms of translation accuracy while these two methods achieve higher accuracy in terms of rotation. Due to the absolute scale recovery problems in classical monocular VO, we also provide the scale-corrected results for both methods inside parentheses in Table I. Scale correction does not improve the performance of DSO to a range comparable to other approaches. Compared to its own unscaled trajectory, ORB-SLAM2 obtains a 10-fold improvement after scaling. Meanwhile, scale-correction has a minimal impact on our method based on sequences 8 and 9 suggesting that our network estimates the absolute scale accurately without a need for correction while sequence 10 does exhibit improvements potentially due to distribution shifts. Overall, our method still outperforms ORB-SLAM2 in terms of averaged accuracy after scale correction.

D. Weighting Analysis

In this section, we visualize the weighting derived by the network for odometry and long-term loss values. Fig. 3(a) and Fig. 3(b) represent the normalized $|\mathbf{\Sigma}^{-1}|$ values for the translation and rotation sections of the covariance matrix, respectively. Based on these two figures, the compounding of the covariance matrices induces exponentially decaying weighting terms for both translation and rotation as the number of steps increases. The direct effect of this approach to weighting can be seen in Fig. 3(c). In this figure, the normalized loss values for uncertainty-based (ours) and uncertainty-less (mean-squared error) loss functions over each training sequence of the dataset are visualized. It can be seen that in the case of using a mean squared error as the loss function, the loss values increase exponentially as more terms are integrated. However, when using our approach, the weighting seen in Fig. 3(a) and 3(b) does not allow the loss to



(a) Translation weighting (b) Rotation Weighting

(c) Normalized compounded loss (d) Normalized frame-to-frame loss

Fig. 3. Effect of weighting on the loss values over the KITTI dataset

TABLE II

LOOP CLOSURE QUANTITATIVE ANALYSES

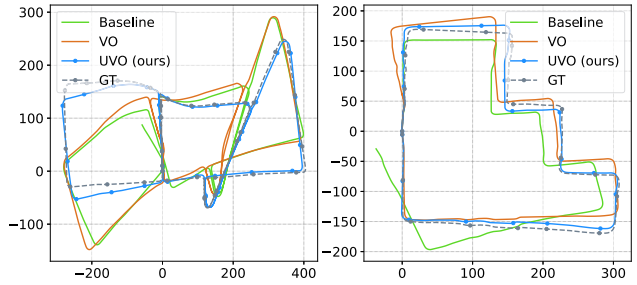
Seq.	Baseline t(%) (Aligned)	Baseline r(°) (Traj.)	VO t(%) (Aligned)	VO r(°) (Traj.)	UVO (ours) t(%) (Aligned)	UVO (ours) r(°) (Traj.)
13	5.109 (5.725)	2.210 (2.210)	8.084 (8.116)	4.125 (4.125)	**3.395** (**3.416**)	**1.390** (**1.390**)
15	14.20 (9.813)	3.465 (3.465)	10.49 (5.135)	1.706 (1.706)	**10.18** (**4.300**)	**1.330** (**1.330**)
Avg.	9.654 (7.769)	2.837 (2.837)	9.287 (6.625)	2.915 (2.915)	**6.787** (**3.858**)	**1.360** (**1.360**)

increase exponentially and the increase in the loss magnitude exhibits a less aggressive behavior. A case study over a 10-step window is also provided in Fig. 3(d). It can be seen that odometry loss for both uncertainty-based and uncertainty-less approaches for this sequence are highly correlated. However, while the MSE loss increases exponentially with the introduction of integration, the uncertainty-based loss does not and rather, the precision term in the compounded loss causes a decrease in the global loss term due to the large amount of uncertainty in the first step of the algorithm. This shows that the balanced weighting for the global and incremental loss terms in our approach requires no manual tuning or dataset-specific changes.

E. UVO and Loop Closure

In this section, we use the incremental pose and uncertainty outputs of the network as the edges of a pose-graph to showcase the benefits of uncertainty estimation in a realistic scenario. We use DBoW3 [23], to detect loops in the provided trajectory. When a loop is detected, an edge connects the corresponding nodes of images in the graph that are in the neighborhood of each other. Then, the pose and uncertainty of this edge are derived by passing this pair of frames to the network itself. By solving this graph in different scenarios we may quantify the effectiveness of using uncertainty in such a setting. To form a baseline, we perform the same experiment once without any loops (termed baseline) and once with fixed uncertainty (termed VO) where the pose matrices are the network outputs. To perform this experiment we use sequences 13 and 15 of the KITTI dataset. Since the ground truth is not provided for these sequences, we used the stereo variant of ORB-SLAM2 [21], which obtains an accuracy of 1.15% on translation and 0.27° on rotation based on the KITTI odometry benchmark, as a reasonably accurate proxy for ground-truth.

The results from this experiment are provided in Table II. Quantitative results are reported in two scenarios. One where the output paths are untouched and one where the paths are scale-corrected. Based on the results from sequence 15, with the addition of loop closure, both uncertainty-based and uncertainty-less approaches provide a significant increase of 28.3% and 26.1% in translation accuracy over the untouched trajectories, respectively. Meanwhile, the scaled paths show that the accuracy increase for uncertainty-based estimates



(a) Sequence 13 (b) Sequence 15

Fig. 4. Loop closure results on the KITTI dataset (scale corrected)

is 8.51% larger compared to that of the uncertainty-less study. Based on the results from sequence 13, not using the estimated uncertainty *degrades* the prediction accuracy by 58.2% on translation and 86.6% on rotation and using the estimated uncertainty allows for an increase in accuracy by 33.5% on translation and 37.1% on rotation. This is because the uncertainty-less experiment incorporates overconfident factors into the graph, while the uncertainty-based method balances the weights of the added factors.

The resulting trajectories from this experiment are visualized in Fig. 4. In the case of sequence 15, at the start of the path (position (0, 50)) the outputs experience a large deviation from the ground-truth while the UVO outputs are able to track the true trajectory accurately. The results on sequence 13 show that the estimated trajectory is able to closely follow the ground-truth trajectory especially in areas where loops are detected (where $x > 0$ in Fig. 4(b)) while uncertainty-less loop closure causes degradation in the estimated trajectory.

V. CONCLUSION

This paper introduces a consistency-based loss function for deep odometry by compounding the estimated SE(3) pose and uncertainties. The compounded terms are then used in a negative log-likelihood objective function where the precision matrices weighting the global loss term are based on the integrated uncertainty. Quantitative results against the SOTA in a visual odometry setting show that the addition of the proposed loss term allows our approach to significantly outperform the recently proposed SOTA methods. Then, the weighting resulted from the estimated precision matrices is visualized and the loss values from UVO are compared to the commonly used mean-squared error loss to show the appropriate balancing of the loss in our approach. Finally, the efficacy of the estimated uncertainties is shown in a loop closure scenario where the constraints between the nodes are the pose and uncertainty estimates from our method. This analysis showed that the uncertainty estimates allow for a significant increase in accuracy while not using the estimated uncertainty to formulate the factors in the graph leads to a diminished accuracy.

REFERENCES

[1] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050.

[2] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, "Selective sensor fusion for neural visual-inertial odometry," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 534–10 543.

[3] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[4] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3946–3952.

[5] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1. IEEE, 2005, pp. 903–910.

[6] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *arXiv:2011.06225*, 2020.

[7] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[8] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[10] G. Costante and M. Mancini, "Uncertainty estimation for data-driven visual odometry," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1738–1757, 2020.

[11] M. R. U. Saputra, P. P. de Gusmao, S. Wang, A. Markham, and N. Trigoni, "Learning monocular visual odometry through geometry-aware curriculum learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3549–3555.

[12] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1436–1443.

[13] C. Li and S. L. Waslander, "Towards end-to-end learning of visual inertial odometry with an ekf," in *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 190–197.

[14] A. De Maio and S. Lacroix, "Simultaneously learning corrections and error models for geometry-based visual odometry methods," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6536–6543, 2020.

[15] G. S. Chirikjian, *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*. Springer Science & Business Media, 2011, vol. 2.

[16] S. Su and C. Lee, "Uncertainty manipulation and propagation and verification of applicability of actions in assembly tasks," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2471–2476 vol.3.

[17] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[18] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[20] 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.

[21] 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.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.

[23] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.