

# Aleatoric Uncertainty from AI-based 6D Object Pose Predictors for Object-relative State Estimation

Thomas Jantos<sup>1</sup>, Stephan Weiss<sup>1</sup> and Jan Steinbrener<sup>1</sup>

**Abstract**—Deep Learning (DL) has become essential in various robotics applications due to excelling at processing raw sensory data to extract task specific information from semantic objects. For example, vision-based object-relative navigation relies on a DL-based 6D object pose predictor to provide the relative pose between the object and the robot as measurements to the robot’s state estimator. Accurately knowing the uncertainty inherent in such Deep Neural Network (DNN) based measurements is essential for probabilistic state estimators subsequently guiding the robot’s tasks. Thus, in this letter, we show that we can extend any existing DL-based object-relative pose predictor for aleatoric uncertainty inference simply by including two multi-layer perceptrons detached from the translational and rotational part of the DL predictor. This allows for efficient training while freezing the existing pre-trained predictor. We then use the inferred 6D pose and its uncertainty as a measurement and corresponding noise covariance matrix in an extended Kalman filter (EKF). Our approach induces minimal computational overhead such that the state estimator can be deployed on edge devices while benefiting from the dynamically inferred measurement uncertainty. This increases the performance of the object-relative state estimation task compared to a fix-covariance approach. We conduct evaluations on synthetic data and real-world data to underline the benefits of aleatoric uncertainty inference for the object-relative state estimation task.

**Index Terms**—AI-Based Methods, Deep Learning Methods, Sensor Fusion, Vision-Based Navigation, Uncertainty Estimation

## I. INTRODUCTION

Deep neural networks (DNNs) excel at computer vision tasks such as object detection, classification, and 6D object pose prediction. The latter is significant for various robotics applications as it provides metric information about the robot’s relative position and orientation with respect to the objects in the scene. Possible robotics applications include robotic manipulation and object-relative navigation with uncrewed aerial vehicles (UAV) for, e.g., infrastructure inspection. Even recent work [1], [2], however, does not include the corresponding uncertainty in this process. Indeed, a DNN that can capture its aleatoric uncertainty to account for and quantify the inherent noise in the data would enable more robust predictions, better decision-making under uncertainty, and appropriate weighting of unreliable inputs in downstream tasks.

Manuscript received: February, 27, 2025; Revised June, 25, 2025; Accepted August, 29, 2025.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers’ comments.

This work was supported by the Austrian Ministry of Climate Action and Energy (BMK), grant agreement FO999895366 (EMFLanding).

<sup>1</sup>The authors are with the Control of Networked Systems Group, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria (`{name.surname}@ieee.org`)

Digital Object Identifier (DOI): 10.1109/LRA.2025.3606700

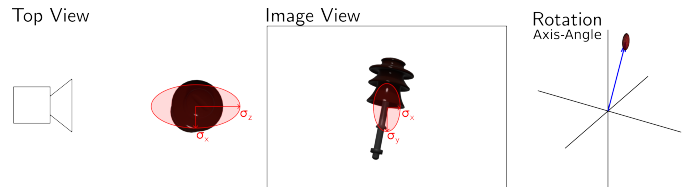


Fig. 1. Exemplary visualization of the predicted aleatoric uncertainty of the 6D pose. The network predicts the uncertainty separately for position and for the rotation. In the axis-angle representation this uncertainty can be interpreted as the uncertainty about the axis of rotation, i.e. vector direction, and the angle of rotation, i.e. length of vector. We visualize the 1- $\sigma$  bound of the 3D Gaussian uncertainty. Please note, that the visualized uncertainties are overemphasized for better understanding.

An extended Kalman filter (EKF) statistically combines information from multiple sources, e.g., sensors, to estimate the system’s state. The measurement covariance quantifies the uncertainty in sensor observations and determines how much weight the filter gives to the measurements compared to the predicted state during the update step. A higher measurement covariance reduces the influence of the measurements, while lower covariance increases their impact, ensuring that the filter adapts appropriately to the reliability of the data.

While EKF covariance tuning for optimized estimator performance is often done and possible for experts, specific single use cases, and for specific measurement sensors, it is hardly an approach scalable and resilient to non-experts under different situations. Also, such tuning becomes more sophisticated when DNNs instead of physical sensors are used to generate the measurements. Recently, Jantos et al. [1], [2] showed with such tuning that 6D object poses predicted by a DNN from raw RGB images can be fused with inertial measurement unit (IMU) data for localization of a mobile robot with respect to objects of interest. Given these advances in DNNs for object-relative navigation, quantifying the uncertainty inherent in the DNN’s prediction is more important than ever to better determine the measurement’s significance and increase the estimation performance.

In this letter, we demonstrate that aleatoric uncertainty can be utilized as measurement covariance in EKFs, combining the best of both worlds, i.e., the high accuracy of the DNN with the statistical signal weighting of an EKF, and replacing the need for any expert knowledge or manual tuning of the measurement uncertainty parameters. We show that a (pre-trained) 6D object pose predictor can be extended to additionally infer the metric, aleatoric uncertainty of the predicted poses, see Fig. 1. Replacing the fixed measurement covariance found with experts’ tuning efforts in the EKF update step with the dynamically predicted aleatoric uncertainty yields advantages from an ease-of-use point of

view and leads to performance improvements. It also allows for confidence based reference object switching and outlier rejection in multi-object relative navigation tasks.

Even though the work presented here will be discussed in the context of inspection with resource-constrained UAVs, the insights gained can be translated into other (object-relative) state estimation scenarios. Our contributions are the following:

- Extending a (pre-trained) 6D object pose predictor for aleatoric uncertainty inference. Modeling the aleatoric uncertainty as multivariate Gaussians to capture the uncertainty of the full 6D pose.
- Utilizing the predicted metric uncertainty as a dynamic measurement covariance matrix in an EKF for object-relative state estimation.
- Implementing uncertainty-based dynamic reference object switching and aleatoric uncertainty-based outlier rejection (AOR) for improved multi-object-relative state estimation.
- Validating the proposed contributions on synthetic and real-world data.

The remainder of the letter is structured as follows. In Section II, we summarize the related work regarding uncertainty prediction in deep learning and its application in state estimation. In Section III, we present our 6D object pose predictor capable of aleatoric uncertainty prediction and the integration of aleatoric uncertainty as dynamic measurement covariance in our state estimator for multi-object-relative state estimation. In Section IV, the experiments and corresponding results are discussed. In Section V, the letter is concluded.

## II. RELATED WORK

With increased prevalence of deep learning approaches across various fields, the importance of quantifying and managing uncertainties of DNN's predictions has grown significantly [3]. There are mainly two types of uncertainty: aleatoric uncertainty, i.e., noise inherent in the data, and epistemic uncertainty, describing the lack of knowledge of the model. While aleatoric uncertainty can not be reduced, epistemic uncertainty can be lowered by increasing the training data size/diversity. For aleatoric uncertainty, an underlying error distribution is assumed and the network is trained to predict these parameters [4].

In object pose prediction, uncertainty prediction refers to the prediction of a pose distribution. Bingham distributions can be used to model the orientation distribution [5]. Alternatively, non-parametric distributions are used to implicitly model the pose distribution [6] or ensembles are used to capture the pose uncertainty [7]. Merrill et al. [8] predict the 2D-pixel uncertainty in keypoint-based object pose prediction and utilize these uncertainties as a covariance for graph-based, object-relative simultaneous localization and mapping (SLAM). Instead of keypoint uncertainty, Zorina et al. [9] empirically determine a fixed 6D pose uncertainty over a test dataset to use in object-relative SLAM. However, the 6D pose covariance is simplified to use the same variance for  $x$  and  $y$  and a single variance value for the three rotational components.

Reliable and accurate state estimation, essential for deploying mobile robots in complex environments and performing various tasks, relies on combining information from multiple sources, e.g., different sensor measurements.

In D3VO, Yang et al. [10] trained a neural network for depth, pose, and pixel brightness uncertainty predictions and used them in the optimization process of a visual-inertial odometry (VIO) algorithm. NVINS [11] is a VIO framework that utilizes a DL-based camera pose regressor and IMU measurements in factor-graph optimization. Additionally, the network predicts aleatoric and epistemic uncertainty of the pose, modeled as a three-dimensional Gaussian for the translation and one-dimensional, isotropic Langevin distribution for the rotation. Combining these two uncertainties results in a four-dimensional covariance matrix, which functions as a weighting factor for the pose measurements during the optimization. Peretroukhin et al. [12] train a multi-head network for aleatoric and epistemic rotation uncertainty prediction, modeled as a three-dimensional Gaussian in the Lie algebra tangent to the predicted rotation. The predicted rotation and its corresponding uncertainty are used in graph-based optimization for visual odometry (VO). Russell and Reale [13] discuss the possibility of using DL-based, multivariate aleatoric and epistemic uncertainty as measurement noise in an EKF for two use cases. First, 3D object tracking by predicting the 3D position and uncertainty. Second, VO by predicting the change in angle and position and the corresponding two-dimensional uncertainty. CoordiNet [14] is a DNN that predicts the camera pose and aleatoric uncertainty from an input image. While the translation uncertainty is assumed to be a three-dimensional Gaussian, the network predicts a single uncertainty value for the rotation, from which the three-dimensional covariance matrix is constructed.

In contrast to the above work, our approach models the full 6D object pose uncertainty as a multivariate Gaussian with individual parameters for the components. Thus in this letter, we propose a method to extend any 6D object pose prediction network to predict aleatoric uncertainty modeled as multivariate Gaussians. To the best of our knowledge, we are the first to utilize the predicted aleatoric uncertainty of the full 6D pose as a dynamic measurement noise covariance matrix in an EKF for object-relative state estimation.

## III. METHOD

In this section, we first describe the deep-learning pose prediction framework, its extension for aleatoric uncertainty prediction, and the loss function. Second, we present the state estimation framework and the integration of the predicted aleatoric uncertainty as a dynamic measurement covariance matrix.

### A. Aleatoric Uncertainty for 6D Object Pose Prediction

A wide variety of deep learning-based approaches exist for 6D object pose prediction, differing in the input modality, availability of additional information, and possible post-processing steps. In this work, we show that a DNN that directly predicts the 6D pose of an object can be easily extended for aleatoric uncertainty prediction. As an example, we choose the open source available object pose prediction framework PoET [15]. It does not rely on any additional input information and directly predicts the relative 6D pose of each object from a single RGB image. Epistemic uncertainty will not be considered as we train the object pose predictor with synthetic data,

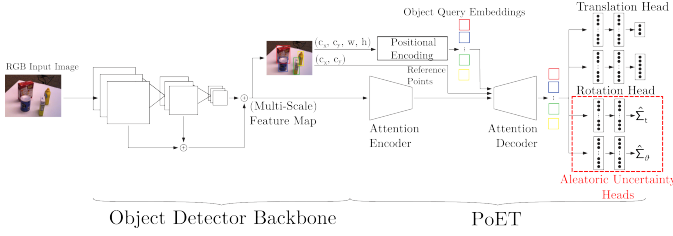


Fig. 2. Our proposed extension to the network architecture of PoET [15] for aleatoric uncertainty prediction (red box). Using the object queries, the aleatoric heads predict the corresponding uncertainties  $\hat{\Sigma}_{t,\vartheta} \in \mathbb{R}^{3 \times 3}$ .

meaning enough data can be generated to cover the object’s  $SO(3)$  space. Besides that, the additional computational load introduced, caused by multiple forward passes, would not be suitable for mobile robot navigation, potentially impeding its real-time implementation as the same authors showed in [1].

After the RGB image is passed through an object detection backbone, the detected objects and multi-scale feature maps are fed into a deformable transformer to predict object queries containing object-specific and global image context information. These object queries are passed separately to a translation and rotation head, simple multi-layer perceptrons (MLPs), to predict each object’s 3D translation and 6D rotation [16]. Using Gram-Schmidt, the 6D rotation is transformed into  $\mathbf{R} \in SO(3)$  of which we take the matrix logarithm to receive an element of its Lie algebra, which is a skew-symmetric matrix

$$\log(\mathbf{R}) = \theta[v]_{\times} \in \mathfrak{so}(3), \quad (1)$$

where  $\theta$  is the angle and  $v$  the axis of rotation. From this we can get the axis-angle representation of the rotation with  $\vartheta = \theta v \in \mathbb{R}^3$ . We extend the network by two additional MLPs to predict each object’s aleatoric uncertainty for the translation  $\hat{\Sigma}_t$  and rotation  $\hat{\Sigma}_\vartheta$ . The aleatoric uncertainty heads predict a class-specific uncertainty in a multi-class scenario. The network architecture is visualized in Fig. 2.

The main advantages of adding dedicated aleatoric uncertainty heads, i.e., being detached from the translation and rotation head, is *the possibility to extend any pre-trained object pose prediction framework for uncertainty inference* and that the heads are able to encapsulate uncertainty-specific features in the MLP’s weights. While the network is still end-to-end trainable, *the uncertainty heads can be trained individually by freezing the rest of the network*, reducing the training time drastically. Moreover, the additional computational overhead is minimal during deployment, thus making our approach suitable for edge device deployment and mobile robotics.

Following a similar idea as [12] and [13], we model the aleatoric uncertainty as being data dependent and a multivariate Gaussian distribution with the mean given by the model output. Given an input image  $x$ , our network outputs the translation  $\hat{t}$  and its covariance  $\hat{\Sigma}_t$ ,  $\mathcal{N}(\hat{t}, \hat{\Sigma}_t)$ , as well as the rotation  $\hat{\vartheta}$  and its covariance  $\hat{\Sigma}_\vartheta$ ,  $\mathcal{N}(\hat{\vartheta}, \hat{\Sigma}_\vartheta)$ , with

$$\hat{\Sigma}_t = \begin{pmatrix} \hat{\sigma}_x^2 & 0 & 0 \\ 0 & \hat{\sigma}_y^2 & 0 \\ 0 & 0 & \hat{\sigma}_z^2 \end{pmatrix} \text{ and } \hat{\Sigma}_\vartheta = \begin{pmatrix} \hat{\sigma}_{\vartheta_1}^2 & 0 & 0 \\ 0 & \hat{\sigma}_{\vartheta_2}^2 & 0 \\ 0 & 0 & \hat{\sigma}_{\vartheta_3}^2 \end{pmatrix}. \quad (2)$$

We predict the uncertainty for the individual components of the predicted translation and rotation and assume no cross-correlation between them. As for the predicted object poses,

the uncertainties are expressed with respect to the camera frame (see Fig. 1 for an explanatory example). While the translational uncertainty refers to the variance of the predicted position, the rotational uncertainty is expressed in the Lie algebra tangential to the predicted rotation.

Considering  $\hat{t}$  and  $\hat{\vartheta}$  to be independent of each other, we assume that the probability of the output  $y \in \{t, \vartheta\}$  can be modeled by a multivariate Gaussian distribution. Given that the network output  $\hat{y}$  for the input image  $x$  models the mean,  $\mathbb{E}[y|x]$ , the likelihood is then

$$p(y|x) = (2\pi)^{-3/2} |\hat{\Sigma}_y|^{-1/2} \exp\left(-\frac{1}{2}(y - \hat{y})^T \hat{\Sigma}_y^{-1} (y - \hat{y})\right). \quad (3)$$

Taking the negative log-likelihood of Eq. (3), we can define the loss function for a single object  $n$  to be

$$\mathcal{L}_{y_n} = \frac{1}{2}(y_n - \hat{y}_n)^T \hat{\Sigma}_{y_n}^{-1} (y_n - \hat{y}_n) + \frac{1}{2} \ln |\hat{\Sigma}_{y_n}| \quad (4)$$

$$= \frac{1}{2} e_n^T \hat{\Sigma}_{y_n}^{-1} e_n + \frac{1}{2} \ln |\hat{\Sigma}_{y_n}|, \quad (5)$$

where  $e_n = (e_{n,1}, e_{n,2}, e_{n,3})^T$  defines the error between components of the ground truth  $y_n$  and predicted value  $\hat{y}_n$  of object  $n$ . In combination with our independent aleatoric heads, this loss definition allows us to either simultaneously or separately train the network for predicting  $\hat{y}_n$  and the corresponding covariance  $\hat{\Sigma}_{y_n}$ . Given our assumption that  $\hat{\Sigma}_{y_n} = \text{diag}(\hat{\sigma}_{y_{n,1}}^2, \hat{\sigma}_{y_{n,2}}^2, \hat{\sigma}_{y_{n,3}}^2) \in \mathbb{R}^{3 \times 3}$ , the loss function simplifies to

$$\mathcal{L}_{y_n} = \frac{1}{2} \left( \sum_{i=1}^3 \frac{e_{n,i}^2}{\hat{\sigma}_{y_{n,i}}^2} + \ln(\hat{\sigma}_{y_{n,i}}^2) \right). \quad (6)$$

The sum of the scaled, squared error components is related to the squared Euclidean norm of a vector. For the translation, the loss function indirectly minimizes the Euclidean distance between the ground truth and predicted translation vectors. The Euclidean norm of the axis-angle is equal to the angle  $\theta$  of the rotation. Hence, the loss function indirectly minimizes the geodesic distance between the ground truth and predicted rotation. Inspired by [4], we predict  $s_{y_{n,i}} = \ln(\hat{\sigma}_{y_{n,i}}^2)$  for better numerical stability during training, resulting in

$$\mathcal{L}_{y_n} = \frac{1}{2} \left( \sum_{i=1}^3 \exp(-s_{y_{n,i}}) e_{n,i}^2 + s_{y_{n,i}} \right). \quad (7)$$

To address multiple objects in an image and arbitrary batch sizes, the loss is averaged by the number of objects  $N$ . Finally, the network is trained with a weighted multi-task loss.

$$\mathcal{L}_y = \frac{1}{N} \left( \sum_{n=1}^N \mathcal{L}_{y_n} \right) \text{ and } \mathcal{L} = \lambda_t \mathcal{L}_t + \lambda_\vartheta \mathcal{L}_\vartheta \quad (8)$$

### B. Object-relative State Estimation

In object-relative state estimation, the current state of a mobile robot, in particular the position and orientation of its propagation sensor, the IMU ( $I$ ), is estimated with respect to objects of interest ( $O_k$ ) in the world ( $W$ ) (i.e., the navigation frame) by measuring the relative 6D pose of the objects. Given two coordinate frames  $A$  and  $B$ , the transformation of frame  $B$  with respect to frame  $A$  is defined by the translation  $\mathbf{p}_{AB}$  and rotation  $\mathbf{R}_{AB}$ .  $\mathbf{I}$  and  $\mathbf{0}$  refer to the identity and the null matrix, respectively. Alternatively, a rotation is expressed

by a quaternion  $\mathbf{q}_{AB} = [\mathbf{q}_v \ q_w]^T = [q_x \ q_y \ q_z \ q_w]^T$ . While optimization-based methods usually achieve more accurate state estimates, they are computationally more demanding as they optimize over multiple sensor measurements. In contrast, filter-based methods, such as the EKF, are better suited for mobile robotics as they are computationally more efficient. Hence, we choose MaRS [17] as our state estimation framework as it was developed with mobile robotics and modularity in mind. This allows for the straightforward implementation of an object-relative pose sensor with a dynamic measurement covariance.

Building on top of previous work [1], [2], the relative pose of objects predicted from RGB images by a DL-based object pose predictor is used for object-relative state estimation. Under the assumption that  $N$  objects of interest are present in the scene, the state vector  $\mathbf{X}$  is given by

$$\mathbf{X} = [\mathbf{p}_{wI}^T, \mathbf{v}_{wI}^T, \mathbf{q}_{wI}^T, \mathbf{b}_\omega^T, \mathbf{b}_a^T, \mathbf{p}_{IC}^T, \mathbf{q}_{IC}^T, \mathbf{p}_{O_0w}^T, \mathbf{q}_{O_0w}^T, \dots, \mathbf{p}_{O_Nw}^T, \mathbf{q}_{O_Nw}^T]^T. \quad (9)$$

The core states for state propagation are the position  $\mathbf{p}_{wI}$ , velocity  $\mathbf{v}_{wI}$  and orientation  $\mathbf{q}_{wI}$  of the IMU, the gyroscopic bias  $\mathbf{b}_\omega$  and the accelerometer bias  $\mathbf{b}_a$ . We estimate the calibration between the IMU and the camera given by  $\mathbf{p}_{IC}$  and  $\mathbf{q}_{IC}$ . Additionally, we estimate the objects in the world ( $\mathbf{p}_{O_kw}, \mathbf{q}_{O_kw}$ ) to relate the object frames to the navigation frame.

In general, an EKF assumes the measurement to depend on a non-linear measurement function of the state corrupted by an additive zero-mean Gaussian noise with the measurement noise covariance matrix  $\Sigma$ , representing the uncertainty of the measurement. Modeling this measurement noise covariance matrix accurately is essential to the estimation process as it is part of the innovation covariance matrix  $\mathbf{S}$ , which captures the uncertainty of the innovation and thus quantifies the trustworthiness of the measurement in the state update step. We will refer to the measurement noise covariance matrix as measurement uncertainty for better readability throughout the manuscript. Determining  $\Sigma$  requires either information about sensor specifications from the manufacturer, an expert to conduct empirical estimation, or iteratively fine-tuning the noise parameters. Empirical analysis of the error distribution on a validation set is a non-learning-based method to determine the measurement uncertainty of a DNN.

Taking the partial derivative of Eq. (6) with respect to one of the uncertainty components  $\hat{\sigma}_{y_{n,i}}$  and given that the standard deviation is strictly positive ( $\hat{\sigma}_{y_{n,i}} > 0$ ), it can be shown that the minimum of the loss function is reached for  $\hat{\sigma}_{y_{n,i}}^2 = e_{n,i}^2$ . Therefore, during training, the aleatoric head's parameters will be optimized so that the predicted uncertainty captures the error of the network. By directly training our object pose estimation network to output a metric uncertainty of its prediction, we can eliminate the need for expert knowledge and manual fine-tuning of the measurement uncertainty in the EKF. Instead, as the aleatoric uncertainty is predicted per image and object, it varies over time and thus allows for a dynamic measurement uncertainty.

For each image, PoET detects all objects of interest and predicts their relative 6D pose to the camera ( $\mathbf{p}_{CO_k}, \mathbf{q}_{CO_k}$ ). Referring to [2], the measurement needs to be inverted to be able to derive the observation matrix  $\mathbf{H}_{O_k}$ . We treat the predicted aleatoric uncertainties, see Eq. (2), as the mea-

surement uncertainty for the translation  $\Sigma_t$  and rotation  $\Sigma_\vartheta$  measurements, respectively.

$\Sigma_t$  is a  $3 \times 3$  matrix and quantifies the measurement noise for the  $xyz$ -components. According to [18], in quaternion-based error-state Kalman filters, small rotational disturbances  $\Delta\vartheta \in \mathbb{R}^3$  can be specified in the local vector space tangent to the actual rotation  $\mathbf{R}_{CO_k} \in SO(3)$ . Hence, the measurement uncertainty  $\Sigma_\vartheta$  of these disturbances can be expressed in the Lie algebra by a regular  $3 \times 3$  matrix. Exactly as for the pose measurements, the predicted uncertainties are expressed in the camera frame and hence need to be inverted using the error propagation law.

$$\Sigma_{y, O_k C} = \mathbf{R}_{O_k C} \Sigma_{y, CO_k} \mathbf{R}_{O_k C}^T. \quad (10)$$

The full measurement noise covariance matrix  $\Sigma_{O_k} \in \mathbb{R}^{6 \times 6}$  for object  $O_k$  is constructed by diagonally stacking the inverted translation and rotation measurement noise matrices:

$$\Sigma_{O_k} = \begin{pmatrix} \Sigma_{t, O_k C} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \Sigma_{\vartheta, O_k C} \end{pmatrix}. \quad (11)$$

To render object-relative state estimation observable, it is necessary to introduce an anchor object ( $O_A$ ), whose pose in the world frame is fixed (thus anchoring the navigation frame to its pose) [2]. Dynamic measurement uncertainty, enabled through the predicted aleatoric uncertainty, allows for a stochastic informed decision to seamlessly switch the anchor object during the estimation process. By summing up the individual components of the predicted aleatoric uncertainty, Eq. (2), the object with the lowest uncertainty, i.e., the measurement the network is most certain about, is chosen as the anchor object for the current update step. We will show below, that dynamically switching the anchor object based on predicted measurement uncertainty will improve the performance of the object-relative state estimator. In case of a fixed measurement uncertainty, dynamic switching of the anchor object is not possible as the filter has no notion about the currently most certain measurement. Note that each anchor object switching process introduces a slight global drift according to the current global estimation error of the new anchor object when switched to. From an object-relative navigation perspective, this is tolerable since object relative motion is still ensured despite the global drift. In addition, switching to the currently most certain object as anchor heavily reduces outliers to be erroneously included in the estimation.

#### IV. EXPERIMENTS & RESULTS

In this section, we present the experimental setup and results. We evaluate the performance of the 6D object pose predictor on a test dataset, analyze its aleatoric uncertainties and demonstrate how these aleatoric uncertainties improve object-relative state estimation.

##### A. Dataset & Network Details

We utilize NVIDIA Omniverse IsaacSim to generate synthetic images of power poles with the insulators being automatically annotated, as they serve as our objects of interest in the object-relative state estimator. Similarly, we create a synthetic dataset with a subset of the YCB-V objects [19]. In both cases, our training and validation dataset consists of



Fig. 3. Example of synthetically generated images for the power pole (left) and the YCB-V objects (right). The numeration of the insulators used during the discussion of the results is indicated in red.

100,000 and 20,000 images, respectively. Synthetic example images are shown in Fig. 3. We choose Scaled-YOLOv4 [20] as our object detection backbone and PoET’s transformer consists of five encoder and decoder layers with 16 attention heads. The translation and rotation heads, as well as both aleatoric uncertainty heads are simple MLPs with an input layer, one hidden layer, and an output layer. Network training is performed on a NVIDIA GeForce RTX 3090.

In order to evaluate the state estimator’s performance, we also simulate physically feasible UAV trajectories with synthetic IMU data (200 Hz) and corresponding synthetic images (15 Hz). The generated trajectories include varying heights and distances to the power pole and YCB-V objects.

The additional computational load of the aleatoric heads is measured for a NVIDIA Jetson Orin AGX 64GB DevKit. Optimizing with TensorRT 8.5.2 with full computational precision, the average image processing time for PoET without aleatoric heads amounts to 106.73ms. The processing time increases to 107.43ms for PoET with aleatoric heads, thus a mere increase of 0.6%.

### B. Aleatoric Uncertainty for 6D Object Pose Prediction

In Section III-A, it was highlighted that the proposed aleatoric uncertainty heads could be either trained separately or in an end-to-end fashion. To investigate the difference, we train two different networks. First, we train PoET without the aleatoric uncertainty heads as described in [15] for 50 epochs. To evaluate the performance of the trained network, we calculate the Euclidean and geodesic distance between the ground truth and predicted translation and rotation. On the validation dataset, the trained network achieves an average translation and rotation error of 3cm and  $2^\circ$ , respectively. To calibrate the network for aleatoric uncertainty estimation, we freeze the network, add the aleatoric heads, and train with the loss function described in Section III-A for 10 epochs (Calib). As the rest of the network is frozen, the training time for an epoch decreases drastically, and the average errors do not change. Second, we train the network for simultaneous 6D object pose and aleatoric uncertainty prediction for 50 epochs (E2E). It achieves the same performance as the Calib network on the synthetic validation data.

To analyze the predicted uncertainty by these two models, we take a synthetically generated trajectory as described in Section IV-A. On this trajectory, see Fig. 4, the distance to the power pole varies between 2.7m and 3.4m, while the height follows a sinusoidal with an amplitude of 0.2m. Due to the varying distance to the power pole, the change of height of the simulated UAV and the trajectory covering  $360^\circ$ , the

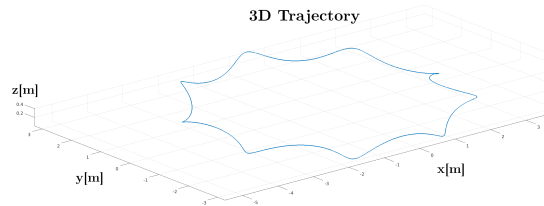


Fig. 4. Visualization of the synthetic trajectory 1 used for analysis in Section IV-B. While rotating to keep at the center located power pole in the view, the UAV also varies its height and distance to the power pole.

TABLE I

COMPARISON OF THE PICP SCORE PER INSULATOR FOR  $\alpha = 0.95$

	Calib			E2E		
	$I_0$	$I_1$	$I_2$	$I_0$	$I_1$	$I_2$
x	1.00	0.99	0.99	1.00	0.99	1.00
y	0.99	1.00	1.00	0.99	0.99	0.96
z	0.97	0.95	0.84	0.93	0.97	0.93
$\vartheta_1$	0.96	0.95	0.98	0.93	0.94	0.95
$\vartheta_2$	0.99	0.99	0.96	1.00	1.00	1.00
$\vartheta_3$	1.00	1.00	0.99	0.93	0.94	0.94

appearance of the power pole in the images covers a wide variety of viewpoints, ideal for the uncertainty analysis. Given the ground truth information from the simulation environment, it is possible to calculate the error statistics and perform the uncertainty analysis across the whole trajectory for each insulator. In addition to the translation and rotation error described above, we also calculate the absolute error component-wise for the translation and the axis-angle representation of the rotation. Similarly, the predicted aleatoric uncertainty is evaluated for each individual component. Given our assumption that the predicted 6D pose and aleatoric uncertainty describe the mean and variance of a Gaussian distribution, the quality of the uncertainty is given by the prediction interval coverage probability (PICP) [3] defined as

$$\text{PICP} = \frac{1}{D} \sum_{d=1}^D \mathbb{I} \left( y_d \in \left[ \hat{y}_d - \frac{1}{2} \Phi^{-1}(\alpha) \cdot \hat{\sigma}_d, \hat{y}_d + \frac{1}{2} \Phi^{-1}(\alpha) \cdot \hat{\sigma}_d \right] \right), \quad (12)$$

where  $D$  is the number of data points,  $\mathbb{I}$  is the indicator function,  $y_d$  is the true value,  $\hat{y}_d$  is the predicted value,  $\hat{\sigma}_d$  is the predicted standard deviation,  $\Phi^{-1}$  is the inverse of the cumulative distribution function of the standard normal distribution and  $\alpha$  is the confidence level. It describes the proportion of true values that lie within the predicted confidence intervals.

In Table I, we report the PICP metric. Assuming a confidence level of 95%, both networks achieve a PICP of above 0.9 for each component, which is a positive indicator for well-calibrated uncertainty predictions. However, PICP values close to 1 indicate that the network overestimates the uncertainty, i.e., making the predictions more uncertain than they are. As will be shown in Section IV-C, the predicted uncertainties are reliable enough to improve object-relative state estimation compared to fixed covariance values. To show that it is sufficient to calibrate a pre-trained 6D object pose predictor for aleatoric uncertainty prediction, the subsequent uncertainty analysis and the experiments for object-relative state estimation will be conducted using the Calib network.

Comparing the error distributions of one insulator across the whole trajectory to fitted normal distributions with the help of a Q-Q plot, cf. Fig. 5, underlines our assumption of a Gaussian distributed error for almost every 6D pose

TABLE II  
PERFORMANCE OF PoET ON THE SYNTHETIC YCB-V DATASET.

Object	AUC of ADD-S	Avg. TE [m]	Avg. RE [°]
cracker box	79.1	0.038	34.52
sugar box	77.3	0.043	34.34
mustard bottle	79.7	0.039	37.95
bleach cleanser	78.9	0.038	26.07
mug	84.6	0.032	49.62
power drill	80.7	0.037	25.02
scissors	66.2	0.053	46.74
All	78.1	0.040	36.06

component. While an overall good Gaussian fit is observable for the translation error components, the  $x$  error shows slight deviations at the high quantiles, indicating larger tails for the Gaussian distribution. On the other hand, slight discrepancies in the rotational components' error distribution are noticeable. Although not as pronounced as for the  $x$  error, the errors of  $\vartheta_1$  and  $\vartheta_3$  also follow a Gaussian distribution with slight deviations towards the tails. In contrast, the error distribution of  $\vartheta_2$  shows more pronounced deviations from the Gaussian distribution towards the tails and slight asymmetry of the error distribution, meaning higher positive errors are more frequent. In general, the error distributions of all 6D pose components are not exactly zero-mean, which might be attributed to the aleatoric loss function, see Eq. (6). As it was outlined in Section III-A, the network parameters are optimized such that the predicted uncertainty ( $\hat{\sigma}_{y_{n,i}}^2$ ) resembles the error ( $e_{n,i}^2$ ) of the prediction, leading to undesirable behavior of the loss function, i.e. steep gradients, when the error is tending towards zero, punishing the network for exact predictions. However, the error means are close enough to zero, so their impact on the object-relative state estimator will be minimal.

In Fig. 6, the absolute rotation error is compared componentwise to the predicted aleatoric uncertainty. We focus on  $I_0$  to observe the uncertainties' behavior in the case of a slightly varying distance to the camera. Except for the case of viewing the power pole from the side ( $t = 41s$  and  $t = 90s$ ), where the  $\vartheta_1$  uncertainty tends towards zero while the error has two succinct peaks, the rotational uncertainty closely follows the error in each component. Additionally, we investigate the influence of the distance to the insulator on the uncertainty prediction for  $I_2$ . The bottom right plot shows the strong correlation between the predicted uncertainty and the distance to the object for the camera's  $z$ -axis, which is pointing out of the image plane. Even though the uncertainty in the  $x$ -component effectively captures local variation in the distance of the  $x$ -direction, particularly at close distances, the uncertainties' overall magnitude is strongly influenced by larger distances along the  $z$ -axis. This is especially visible in the time between 80s and 100s. The distance to the object is the primary source of translation aleatoric uncertainty, which can be explained by the amount of information available regarding the object. The further away the object is, the fewer pixels capture it, providing less information to the network. Note the different scales in the plots' axes and across the plots.

For the YCB-V dataset, PoET is trained using the `Calib` scheme described above. In Table II, we report the performance of PoET on the validation dataset in terms of the AUC of ADD-S [19], average translation and rotation error. As for the power pole, the predicted uncertainty is analyzed

TABLE III  
COMPARISON OF THE PICP SCORE FOR EACH OBJECT FOR  $\alpha = 0.95$

Object	x	y	z	$\vartheta_1$	$\vartheta_2$	$\vartheta_3$
cracker box	0.987	0.995	0.866	0.979	0.905	0.954
sugar box	0.990	0.998	0.869	0.969	0.885	0.992
mustard bottle	0.994	0.999	0.842	0.975	0.961	0.993
bleach cleanser	0.996	0.998	0.997	0.924	0.922	0.988
mug	0.982	1.000	0.862	0.703	0.953	0.228
power drill	0.996	0.993	0.962	0.970	0.941	0.992
scissors	0.988	0.998	0.901	0.865	0.831	0.313

by placing each object at the center of a synthetic trajectory with a fixed distance of 1.0m. The PICP scores are reported in Table III. Except for the  $z$ -component of some objects, the predicted aleatoric uncertainty captures the errors of the translation and rotation components. The only outliers are the rotational components of the mug and the scissors, which are almost symmetrical objects. In Fig. 7, the absolute translation and rotation errors are compared to the predicted aleatoric uncertainty of the mug. While the uncertainty nicely encapsulates the translation components, the rotation components exhibit heightened error and uncertainty values for ambiguous scenarios, i.e., where the handle is not clearly visible. Given this observation, performing aleatoric uncertainty-based outlier rejection (AOR) is possible by determining suitable thresholds.

The results indicate that our proposed network can capture the metric aleatoric uncertainty of the predicted 6D object pose. As modeled in Section III-A, the translation and the rotation uncertainty can be predicted componentwise and capture the characteristics of the error, which are Gaussian distributed.

### C. Object-relative State Estimation

The in-depth analysis above shows that the predicted aleatoric uncertainty adequately represents the potential error of the predicted, relative 6D object pose. This section illustrates that these aleatoric uncertainties can be used as a dynamic measurement uncertainty in object-relative state estimation. This eliminates the need for expert knowledge and manual tuning and also enables dynamic anchor object switching benefiting the object-relative state estimation task.

Following Section IV-A, we generated five trajectories with varying distances, heights, and durations to evaluate the influence of the measurement uncertainty on the performance of the object-relative state estimator. We compare three different approaches for determining and utilizing the measurement uncertainty  $\Sigma_t$  and  $\Sigma_\vartheta$ : First, following the idea of [9] to empirically determine the measurement uncertainty, the average error on the validation dataset is calculated and the standard deviation for each component is set to the corresponding error, i.e.,  $\sigma_{x,y,z} = 0.03m$  and  $\sigma_{\vartheta_{1,2,3}} = 0.035rad$ . This measurement noise covariance is kept fixed throughout the estimation (`Fixed`). Second, the predicted aleatoric uncertainties are used to construct the measurement uncertainty as described in Section III-B. Third, dynamic anchor object switching is enabled based on the dynamic measurement noise derived from the aleatoric uncertainty. The predicted 6D object poses from the `Calib` network are used as measurements for each approach. In Table IV, the state estimator's performance is evaluated based on the root mean square error (RMSE) for position and orientation, as well as the maximum position error (PE). Compared to empirically determined uncertainty (left columns), the



Fig. 5. Componentwise Q-Q plot comparing the error distribution (red) for the translation (left) and the rotation (right) to a fitted Gaussian distribution (blue). The comparison is conducted for insulator 0 and the trajectory depicted in Fig. 4.

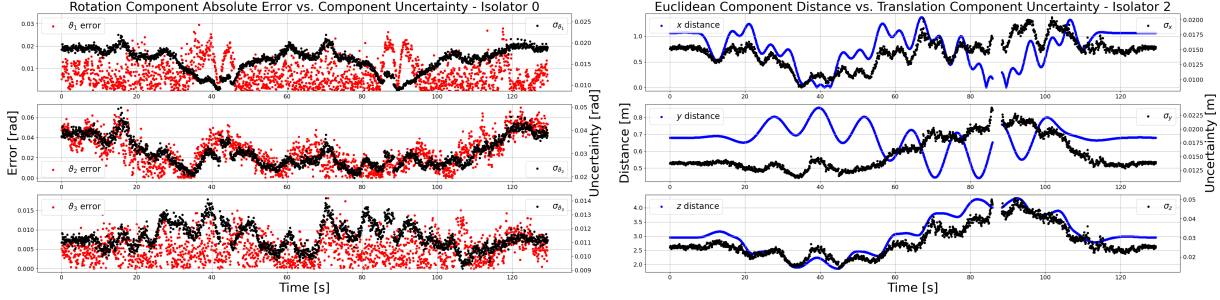


Fig. 6. Left: Comparison of the absolute rotation error (red) to the estimated aleatoric uncertainty (black) across the whole trajectory for  $I_0$ . Right: Comparison of the distance (blue) to the estimated aleatoric uncertainty (black) across the whole trajectory for  $I_2$ . The gaps in the graphs are caused by the power pole occluding the insulator. Note the different scales in the plots' axes (left and right sides of the plots) and across the plots.

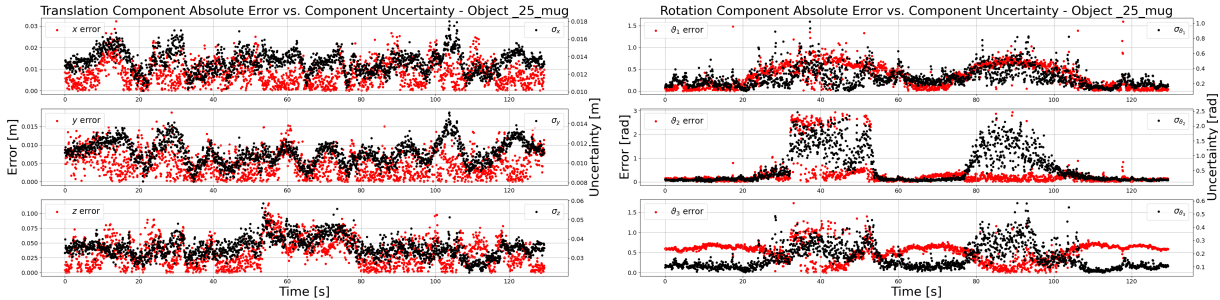


Fig. 7. Comparison of the absolute translation error (left, red) and rotation error (right, red) to the estimated aleatoric uncertainty (black) across the whole trajectory for the mug. Note the different scales in the plots' axes (left and right sides of the plots) and across the plots.

aleatoric measurement uncertainty (middle columns) achieves a comparable performance for all three metrics but without the need for prior empirical estimation. Including dynamic switching of the anchor object, enabled only through the aleatoric uncertainty prediction per measurement, drastically improves the performance for object-relative state estimation (right columns). This highlights the benefits of aleatoric uncertainty for DL-based object-relative state estimation.

To verify its real world applicability, we evaluate our approach on the real-world UAV data from [1]<sup>1</sup>. It mimics an inspection flight around a power pole by flying on a  $\pm 50^\circ$  arc at a fixed distance of 3.3m. The data includes real IMU data at 200 Hz, RGB images at 15 Hz, and ground truth data at 60 Hz from a motion capture system. The provided noise values are not changed. The results are reported in Table V. As in the case of the synthetic data, the state estimator achieves a similar performance for the fixed and aleatoric uncertainty-based measurement uncertainty, while the dynamic anchor object switching once again leads to improvements. The predicted aleatoric uncertainty of the 6D

object pose predictor, solely trained on synthetic data, also benefits the state estimation in real-world scenarios.

For the YCB-V objects, we simulate five synthetic trajectories with different object constellations. Once again, the fixed measurement noise is determined by the average error of PoET reported in Table II, i.e.,  $\sigma_{x,y,z} = 0.04\text{m}$  and  $\sigma_{\theta_{1,2,3}} = 0.628\text{rad}$ . For comparison, the aleatoric uncertainty is combined with AOR. We report the RMSE for position and orientation in Table VI. Aleatoric uncertainty and AOR improve the state estimation performance.

## V. CONCLUSION

In this letter, we presented an approach to extend a (pre-trained) DL-based 6D object pose predictor for aleatoric uncertainty prediction of the full 6D pose, modeled as a multivariate Gaussian. The predicted metric aleatoric uncertainty captures the error characteristics of the predicted 6D pose and is thus a well suited uncertainty measure for down-stream robotics tasks and can be used for outlier rejection (AOR). Concretely, we showed that utilizing the predicted aleatoric uncertainty as dynamic measurement covariance leads to improvements in the object-relative state estimation task. Besides that, it removes the need for expert knowledge or tedious empirical

<sup>1</sup>Online available at AIVIO dataset homepage.

TABLE IV

RESULTS FOR OBJECT-RELATIVE STATE ESTIMATION USING SYNTHETIC TRAJECTORIES. WE COMPARE FIXED MEASUREMENT UNCERTAINTY WITH OUR PROPOSED ALEATORIC UNCERTAINTY AND DYNAMIC ANCHOR OBJECT SWITCHING. BOLD VALUES HIGHLIGHT BEST RESULTS.

Trajectory	Fixed			Aleatoric Uncertainty (AU)			AU + Dynamic Anchor Object Switching		
	RMSE[m]	RMSE[°]	Max PE [m]	RMSE[m]	RMSE[°]	Max PE [m]	RMSE[m]	RMSE[°]	Max PE [m]
1	0.142	2.44	0.305	0.143	2.46	<b>0.302</b>	<b>0.128</b>	<b>2.26</b>	0.304
2	0.136	2.33	0.297	0.137	2.36	0.296	<b>0.130</b>	<b>2.27</b>	<b>0.261</b>
3	<b>0.118</b>	<b>2.05</b>	0.292	<b>0.118</b>	<b>2.05</b>	0.292	0.119	2.13	<b>0.284</b>
4	0.160	2.93	0.347	0.161	2.94	0.347	<b>0.121</b>	<b>2.12</b>	<b>0.257</b>
5	0.143	2.31	0.342	0.147	2.37	0.375	<b>0.130</b>	<b>2.10</b>	<b>0.295</b>
Mean	0.140	2.41	0.317	0.141	2.44	0.322	<b>0.126</b>	<b>2.18</b>	<b>0.280</b>

TABLE V

RESULTS FOR OBJECT-RELATIVE STATE ESTIMATION USING REAL-WORLD DATA [1]. WE COMPARE FIXED MEASUREMENT UNCERTAINTY WITH OUR PROPOSED ALEATORIC UNCERTAINTY AND DYNAMIC ANCHOR OBJECT SWITCHING. BOLD VALUES HIGHLIGHT BEST RESULTS.

Flight	Fixed			Aleatoric Uncertainty (AU)			AU + Dynamic Anchor Object Switching		
	RMSE[m]	RMSE[°]	Max PE [m]	RMSE[m]	RMSE[°]	Max PE [m]	RMSE[m]	RMSE[°]	Max PE [m]
1	0.176	2.47	0.350	0.175	2.40	0.359	<b>0.151</b>	<b>2.09</b>	<b>0.293</b>
2	0.177	2.00	0.372	0.193	2.02	0.381	<b>0.140</b>	<b>1.72</b>	<b>0.315</b>
3	0.180	2.52	0.356	0.187	2.61	0.373	<b>0.128</b>	<b>1.93</b>	<b>0.257</b>
4	0.163	2.00	0.314	0.170	2.05	0.335	<b>0.139</b>	<b>1.80</b>	<b>0.277</b>
5	0.165	<b>2.09</b>	0.312	0.169	2.10	0.313	<b>0.148</b>	2.18	<b>0.291</b>
6	0.178	2.27	0.375	0.187	2.19	0.394	<b>0.138</b>	<b>1.83</b>	<b>0.334</b>
7	<b>0.167</b>	<b>2.10</b>	<b>0.350</b>	0.174	<b>2.10</b>	0.352	0.180	2.52	0.355
Mean	0.172	2.21	0.347	0.179	2.21	0.358	<b>0.146</b>	<b>2.01</b>	<b>0.303</b>

TABLE VI

RESULTS FOR OBJECT-RELATIVE STATE ESTIMATION USING SYNTHETIC YCB-V DATA. BOLD VALUES HIGHLIGHT BEST RESULTS.

	Fixed		AU + AOR	
	RMSE[m]	RMSE[°]	RMSE[m]	RMSE[°]
1	0.593	34.31	<b>0.209</b>	<b>7.18</b>
2	<b>0.182</b>	20.06	0.201	<b>8.23</b>
3	0.567	18.38	<b>0.513</b>	<b>15.06</b>
4	0.516	28.94	<b>0.383</b>	<b>17.51</b>
5	0.265	25.82	<b>0.248</b>	<b>10.66</b>
Mean	0.425	25.50	<b>0.311</b>	<b>11.73</b>

experiments to determine a suitable measurement covariance. The predicted uncertainty per measurement also allowed a stochastic decision process to dynamically determine the object to which the navigation frame is anchored to. This is not possible with fix uncertainty approaches and led to significant state estimation improvements in both synthetic and real experiments. We also highlighted that the 6D object pose predictor including the aleatoric uncertainty heads can be trained solely on synthetic data with its performance translating to real-world data. Moreover, it was shown that it is sufficient to freeze the pre-trained object detector and calibrate the aleatoric uncertainty heads to achieve performance improvements for the state estimation task.

## REFERENCES

- [1] T. Jantos, M. Scheiber, C. Brommer, E. Allak, S. Weiss, and J. Steinbrener, "Aivio: Closed-loop, object-relative navigation of uavs with aided visual inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 10764–10771, 2024.
- [2] T. Jantos, C. Brommer, E. Allak, S. Weiss, and J. Steinbrener, "Ai-based multi-object relative state estimation with self-calibration capabilities," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2789–2795.
- [3] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al., "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. Suppl 1, pp. 1513–1589, 2023.
- [4] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.
- [5] H. Sato, T. Ikeda, and K. Nishiwaki, "A probabilistic rotation representation for symmetric shapes with an efficiently computable bingham loss function," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 6923–6929.
- [6] R. L. Haugaard, F. Hagelskjær, and T. M. Iversen, "Spyrpose: Se (3) pyramids for object pose distribution estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [7] K. Wursthorn, M. Hillemann, and M. Ulrich, "Uncertainty quantification with deep ensembles for 6d object pose estimation," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024.
- [8] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14901–14910.
- [9] K. Zorina, V. Priban, M. Fourmy, J. Sivic, and V. Petrik, "Temporally consistent object 6d pose estimation for robot control," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 56–63, 2025.
- [10] N. Yang, L. v. Stumberg, R. Wang, and D. Cremers, "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1281–1292.
- [11] J. Han, L. L. Beyer, G. V. Cavalheiro, and S. Karaman, "Nvins: Robust visual inertial navigation fused with nerf-augmented camera pose regressor and uncertainty quantification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.
- [12] V. Peretroukhin, B. Wagstaff, M. Giamou, and J. Kelly, "Probabilistic regression of rotations using quaternion averaging and a deep multi-headed network," *arXiv preprint arXiv:1904.03182*, 2019.
- [13] R. L. Russell and C. Reale, "Multivariate uncertainty in deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7937–7943, 2021.
- [14] A. Moreau, N. Piasco, D. Tsishkou, B. Stanculescu, and A. de La Fortelle, "Coordinet: uncertainty-aware pose regressor for reliable vehicle localization," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.
- [15] T. Jantos, M. A. Hamdad, W. Granig, S. Weiss, and J. Steinbrener, "PoET: Pose Estimation Transformer for Single-View, Multi-Object 6D Pose Estimation," in *Proceedings of the 6th Conference on Robot Learning*. PMLR, 2023.
- [16] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.
- [17] C. Brommer, R. Jung, J. Steinbrener, and S. Weiss, "Mars: A modular and robust sensor-fusion framework," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 359–366, 2020.
- [18] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
- [19] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.
- [20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 029–13 038.