

# Variational Voxel Pseudo Image Tracking

Illia Oleksiienko\*, Paraskevi Nousi†, Nikolaos Passalis†, Anastasios Tefas† and Alexandros Iosifidis\*

\**DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Denmark*

†*Department of Informatics, Aristotle University of Thessaloniki, Greece*

{io, ai}@ece.au.dk    {paranous, passalis, tefas}@csd.auth.gr

**Abstract**—Uncertainty estimation is an important task for critical problems, such as robotics and autonomous driving, because it allows creating statistically better perception models and signaling the model’s certainty in its predictions to the decision method or a human supervisor. In this paper, we propose a Variational Neural Network-based version of a Voxel Pseudo Image Tracking (VPIT) method for 3D Single Object Tracking. The Variational Feature Generation Network of the proposed Variational VPIT computes features for target and search regions and the corresponding uncertainties, which are later combined using an uncertainty-aware cross-correlation module in one of two ways: by computing similarity between the corresponding uncertainties and adding it to the regular cross-correlation values, or by penalizing the uncertain feature channels to increase influence of the certain features. In experiments, we show that both methods improve tracking performance, while penalization of uncertain features provides the best uncertainty quality.

**Index Terms**—3D Single Object Tracking, Point Cloud, Uncertainty Estimation, Bayesian Neural Networks, Variational Neural Networks, Machine Learning for Embedded Devices

## I. INTRODUCTION

3D Single Object Tracking (3D SOT) is the task of tracking an object in a 3D scene based on the given initial object position. This task combines challenges from both 3D Object Detection, as objects have to be accurately located in space, and 3D Multiple Object Tracking, as the object of interest has to be distinguished from similar objects. There is a variety of sensors that can be used for 3D SOT, including single or double camera setups, Lidar and Radar. While the camera setups are the cheapest option, they capture images which lack valuable depth information for 3D SOT, which can be provided by Lidar sensors. Lidars generate point clouds, which are sets of 3D points detected as the positions in the 3D scene of light beam reflections. The explicit depth information makes Lidar the most common choice for many 3D perception methods, including 3D SOT. The SOT is performed by predicting the offset of the object’s position with respect to its previous known position. This has been approached by using correlation filters [1], [2], deep learning methods to directly predict the object’s offset [3], or by using Siamese methods which search for the position with the highest similarity score [4]–[8]. Since 3D perception methods are often used in critical fields, such as robotics or autonomous driving, it is important to provide accurate predictions and confidence estimations to avoid costly damages.

Uncertainty estimation in neural networks allows for using the network’s outputs to better indicate the confidence in its predictions and to improve their statistical qualities, leading to better performance. The practical applications of uncertainty estimation are studied for several perception tasks, including 3D Object Detection [9]–[11], 3D Object Tracking [12], [13], 3D Human Pose Tracking [14], and Steering Angle Prediction [15]. These methods provide an improvement in perception and control by using an uncertainty estimation process. However, most of these methods adopt single deterministic approaches to estimate different types of uncertainty, or use Monte Carlo Dropout (MCD) [16] as an approach to estimate epistemic uncertainty. According to experiments in [17] on the uncertainty quality of different types of Bayesian Neural Networks (BNNs), MCD achieves the worst uncertainty quality.

Utilization of uncertainty estimation for perception can be divided into two categories: to improve accuracy of perception and to use uncertainty to change the decision of the control system. The first approach is easier to implement with existing datasets and tools, which is the reason why we chose it in our work. This approach is based on the assumption that statistically better models and processes lead to better generalization and more robust models. Both uncertainty estimation and utilization processes theoretically lead to an improvement in the aforementioned metrics. The decision-based approach uses uncertainty measurements from the perception model and utilizes that to either switch control from the model to a human expert (such as in autonomous driving or critical system control) or to a more accurate but resource-demanding model. This approach is harder to implement and test, and is a direction for future work.

In this paper, we introduce a Variational Neural Network (VNN) [18] based version of the fastest 3D SOT method called Voxel Pseudo Image Tracking (VPIT) [8] and propose two ways, i.e., the uncertainty similarity approach and the penalization approach, to utilize the estimated uncertainty and improve the tracking performance of the model. The similarity-based approach computes a similarity between the estimated uncertainties to serve as an additional similarity score, while the penalization approach focuses on certain features by penalizing the feature values corresponding to high uncertainties. We train a VNN version of PointPillars for 3D Object Detection to serve as backbone for the proposed Variational VPIT (VVPIT) method. We, then, train the whole

network following the VPIT’s training procedure, but use the uncertainty-aware cross-correlation function and multiple samples of the Variational Feature Generation Network to compute uncertainty in the produced features. In experiments, we show that the use of uncertainty leads to an improvement in the model’s tracking performance, and the choice of the penalty-based uncertainty utilization strategy leads to the highest improvement in Success and Precision metrics [19].

The remainder of the paper is structured as follows. Section II describes related and prior work. In Section III we describe the proposed approach, including the Variational TANet model and its training and the proposed uncertainty-aware AB3DMoT. Section IV outlines the experimental protocol and provides experimental results. Section V concludes this paper<sup>1</sup>.

## II. RELATED WORK

Gawlikowski et al. [20] define four main categories of uncertainty estimation methods, based on the strategies they use to estimate the uncertainty of the model. Deterministic Methods [12], [21] use a single deterministic network and either predict its uncertainty by using an additional regression branch, or estimate it by analyzing the output of the model. Bayesian Neural Networks (BNNs) [22], [23] consider a distribution over weights of the network and compute the outputs of multiple model samples for the same input. The variance in the network’s outputs expresses the estimated uncertainty, while the mean of outputs is used as the prediction value. Ensemble Methods [24], [25] consider a categorical distribution over the weights of the network and train multiple models at once. Test-Time Data Augmentation methods [26]–[28] apply data augmentations commonly used in the training phase during the inference to pass distorted inputs to a single deterministic network and compute the variance in the model’s outputs.

Variational Neural Networks [18], [29] are similar to BNNs, but instead of considering a distribution over weights, they place a Gaussian distribution over the outputs of each layer and estimate its mean and variance values by the corresponding sub-layers. All types of uncertainty estimation methods, except those in the Deterministic Methods category, use multiple model passes to compute the variance in the network’s outputs. This means the Deterministic Methods generally have the lowest computational impact on the model, but they usually perform worse than other methods. The single deterministic network approach can be improved by considering the Bayesian alternative, as it can be seen as a case of BNNs with the simple Dirac delta distribution over weights, which places the whole distributional mass on a single weight point.

The 3D SOT task is usually approached by using point-based Siamese networks, which consider a pair of target and search regions, predict a position of the target region inside the search region and compute the object offset relative to the previous object position. P2B [30], BAT [31], Point-Track-Transformer (PTT) [32], [33] and 3D-SiamRPN [4] use point-wise Siamese networks and predict object positions based

on the comparison of target and search point clouds. 3D Siam-2D [34] uses one Siamese network in a 2D Birds-Eye-View (BEV) space to create fast object proposals and another Siamese network in 3D space to select the true object proposal and regress the bounding box. Voxel Pseudo Image Tracking (VPIT) [8] uses voxel pseudo images in BEV space and deploys a SiamFC-like module [5] to extract and compare features from target and search regions. Instead of using different scales, VPIT uses a multi-rotation search to find the correct vertical rotation angle.

Bayesian YOLO [35] is a 2D object detection method that estimates uncertainty by combining Monte Carlo Dropout (MCD) [16] with a deterministic approach and predicts aleatoric uncertainty with a special regression branch, while computing the epistemic uncertainty from the variance in MCD model predictions. Feng et al. [9] use a Lidar-based 3D object detection method and estimate the uncertainty in the predictions of the model in a similar way to Bayesian YOLO, by using a partially MCD model for the epistemic uncertainty estimation and using a separate regression branch for the aleatoric uncertainty estimation. LazerNet [10] predicts the uncertainty of a 3D bounding box using a single deterministic network and utilizes the predicted uncertainty during the non-maximum suppression process. This approach is further improved by estimating the ground truth labels’ uncertainty based on the IoU between the 3D bounding box and the convex hull of the enclosed point cloud, and using the provided uncertainties during the training process [11].

Zhong et al. [12] perform 3D Multiple Object Tracking (MOT) by using a single deterministic network for 3D Object Detection to predict the uncertainty in outputs and providing the estimated uncertainties to the tracker by replacing the unit-Gaussian measurement noise in Kalman filter [36] with the predicted uncertainties. Uncertainty-Aware Siamese Tracking (UAST) [37] performs 2D single object tracking by using a single deterministic network and computing the distribution over the outputs by quantizing over the specific range of values and predicting the softmax score for each quantized value. The final regression value is computed as an expectation of the corresponding quantized distribution, and the distributions are used to estimate better confidence scores and select the best box predictions.

To the best of our knowledge, there are no methods that utilize uncertainty for 3D Single Object Tracking. Moreover, the estimation of uncertainty for related tasks, such as 2D Single Object Tracking, 3D Multiple Object Tracking or 3D Object Detection, is based on single deterministic networks or MCD, despite the fact that the statistical quality of single deterministic networks can be improved by using a Bayesian alternative, and that MCD tends to produce the worst quality of uncertainty between BNNs [17].

## III. METHODOLOGY

We select the Voxel Pseudo Image Tracking (VPIT) [8] method as a base model for uncertainty estimation in 3D single object tracking, as it achieves the highest inference speed on

<sup>1</sup>Our code is available at [gitlab.au.dk/maleci/opendr/vnn\\_vpitr\\_opendr](https://gitlab.au.dk/maleci/opendr/vnn_vpitr_opendr)

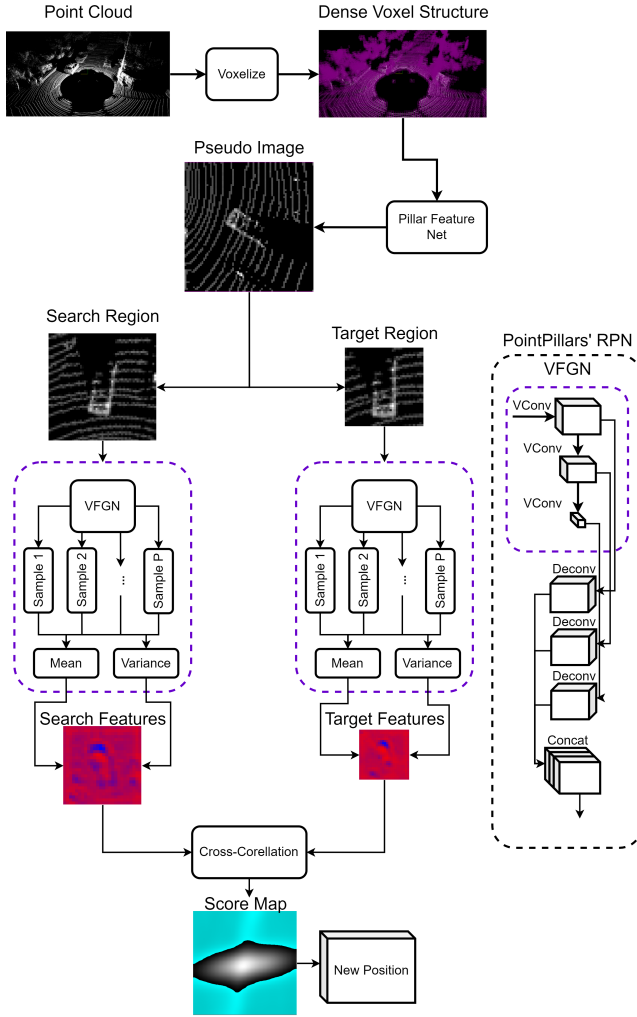


Fig. 1: Variational Voxel Pseudo Image Tracking structure.

an outdoor KITTI tracking dataset [38]. Additionally, VPIT suffers less speed loss when running on embedded devices, compared to the powerful desktop GPUs, which are the target devices for applying 3D perception methods. The difference in speed loss between VPIT and other real-time 3D SOT methods [30], [32] is explained by the specific architecture of such devices, where RAM and GPU memory are shared, meaning that transfer of data between CPU and GPU is instant, and VPIT is more suitable for such memory architecture. The speed of the model is an important factor because the addition of uncertainty estimation usually results in an increase in computational requirements, which forces the use of more lightweight models or devices with higher computational capabilities, but the addition of uncertainty estimation may result in a difference between a costly or dangerous outcomes, e.g., a crash of an autonomous car or robot, and an in-time prevented perception loss by either the model itself, or the decision-based control system. Voxel Pseudo Image Tracking uses PointPillars [39] as a backbone to create voxel pseudo images and to process them with a Feature Generation Network (FGN),

which consists of the convolutional part of the PointPillars' Region Proposal Network. The search and target features are compared with a convolutional cross-correlation function that calculates a pixel-wise similarity map. The highest value in this similarity map is used to determine the object position offset between frames.

We train a Variational VPIT (VVPIT) model by replacing the FGN subnetwork with a Variational Neural Network (VNN) [18], [29] based version of it, i.e., we create a Variational FGN (VFGN). The structure of the proposed Variational VPIT model is present in Fig. 1. The input point cloud is voxelized and processed by a small Pillar Feature Network to create a Pseudo Image of the region of interest. The search and target sub-images are processed with the same VFGN network that uses Variational Convolutional (VConv) layer to create outputs of multiple network samples for the same input, which are then processed to compute mean and variance for the output features. The number of samples can be dynamic and is not required to be the same during training and inference. For each of the target and search regions, VFGN produces a set of outputs in the form  $Y = \{y_i, i \in [1, \dots, P]\}$  which correspond to the outputs of  $P$  sampled VFGN models, with  $Y^s = \{y_i^s, i \in [1, \dots, P]\}$  corresponding to the search region output set and  $Y^t = \{y_i^t, i \in [1, \dots, P]\}$  to the target region output set. The number of samples  $P$  can be different for each set, but for simplicity, we use the same number of samples for both target and search regions. The means and variances of the outputs are computed as follows:

$$y_m^s = \frac{1}{P} \sum_i^P y_i^s, \quad y_v^s = \text{diag} \left( \frac{1}{P} \sum_i^P (y_i^s - y_m^s)(y_i^s - y_m^s)^T \right),$$

$$y_m^t = \frac{1}{P} \sum_i^P y_i^t, \quad y_v^t = \text{diag} \left( \frac{1}{P} \sum_i^P (y_i^t - y_m^t)(y_i^t - y_m^t)^T \right),$$
(1)

where  $y_m^s, y_v^s$  and  $y_m^t, y_v^t$  are the mean and variance values of search and target output feature sets, respectively, and  $\text{diag}(\cdot)$  is a function that returns the main diagonal of a matrix. Fig. 2 shows an example of the mean and variance values of features generated by the VFGN for a search region with a car in the center. The background pixels have mostly high certainty, as all sampled models agree on them being irrelevant. The high magnitude features at the top part of the car have the highest uncertainty, as different model samples can disagree on the details in the appearance of the object.

The proposed VVPIT method can utilize the predicted uncertainties in different ways. The simplest way is to entirely ignore the uncertainty values and process the mean outputs only with the regular cross-correlation function  $g(a, b)$ , defined as a 2D convolution  $\text{conv}2D_{\omega=b}(a)$  with  $\omega$  being the kernel weights. This still leads to a statistically better model which can provide better predictions, but it can be further improved by utilizing the predicted uncertainties in the cross-correlation module. Since most 3D SOT methods compare region features in a similarity manner, we focus on similarity-based approaches to use the uncertainty values, instead of

applying distance-based approaches. We propose a double similarity-based process to utilize uncertainty, which treats mean and variance values as separate feature sets and uses the convolutional similarity function  $g(a, b)$  on both of them independently. The final similarity value  $\hat{g}_{\text{double}}$  is obtained by linearly the similarities of the mean and variance of the outputs as follows:

$$\hat{g}_{\text{double}}(y_m^s, y_m^t, y_v^s, y_v^t) = g(y_m^s, y_m^t) + \lambda g(y_v^s, y_v^t), \quad (2)$$

where  $\lambda$  is a variance weight hyperparameter. This approach is based on the idea that positions with similar uncertainties should be prioritized, as there is a high chance of them representing the same object. Humans can also treat uncertainties as separate features. Let us consider a task of classifying triangle and circle images, where some objects are rounded triangles. Based on the deformation degree, people will have different values of aleatoric uncertainty in their predictions, as they will have harder time classifying in their predictions, as they will have harder time classifying rounded triangles as only one of the two classes. If a person is asked to track these objects, the aleatoric uncertainty in predictions may be the only feature needed to distinguish between objects, given that size, thickness and other features are identical. This is achieved by describing the tracked objects as “definitely a circle”, “triangle with some curves”, “in between the circle and the triangle”, which leads to low chances of mixing up these objects during tracking. The same principle can be applied for Lidar-based 3D SOT task. However, there are many different sources of uncertainty, considering the varying point cloud density, possible occlusions and object rotation. Some parts of the object of interest may have uncertain features, and this uncertainty is likely to be preserved during the tracking process.

In addition to the above approach, we also define an uncertainty penalization process which places focus on features with higher certainty and penalizes the uncertain feature values. This is achieved by dividing each mean feature value during the convolutional process by the corresponding normalized variance score, as follows:

$$\begin{aligned} \forall c, v_n^c(v) &= (\rho - 1) \frac{v^c - \min(v^c)}{\max(v^c) - \min(v^c)} + 1, \\ \forall p_x, \forall p_y, \hat{g}_{\text{pen}}(y_m^s, y_m^t, y_v^s, y_v^t)^{p_x, p_y} &= \\ &= \frac{2y_m^s{}^{p_x, p_y} y_m^t{}^{p_x, p_y}}{v_n(y_v^s)^{p_x, p_y} + v_n(y_v^t)^{p_x, p_y}}, \end{aligned} \quad (3)$$

where the  $v_n(v)$  function is used to normalize the variance predictions by the channel-wise minimum and maximum values to be in  $[1, \rho]$  range, with a hyperparameter  $\rho$  that defines how much the uncertain predictions are penalized,  $v_n^c(v)$  implements the normalization procedure for a single channel  $c$ . For an input  $j$ , the notation  $\tilde{j}$  denotes the tensor with convolutional patches of  $j$ , and  $j^{p_x, p_y}$  corresponds to the values of  $j$  at position  $(p_x, p_y)$ .

We follow the VPIT’s training protocol and initialize a VVPIT model based on the VNN version of PointPillars for 3D Object Detection. After the initialization, the model is

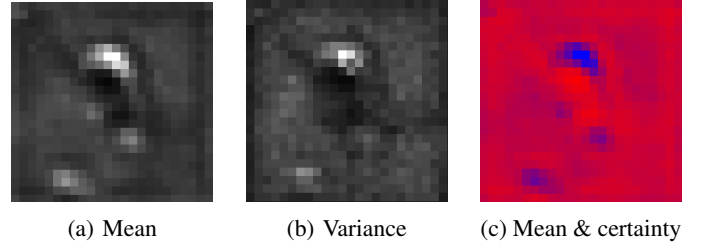


Fig. 2: An example of (a) mean, (b) variance and (c) mean and certainty features of a search region with a car in the center. Lighter color in the mean and variance images corresponds to higher values. Red color channel represents certainty in the corresponding pixel values, and blue color channel represents the mean feature values. The purple color indicates that the feature values and the certainty in those values is equally high, while the blue pixels signal features with high values and low certainty.

trained with the Binary Cross-Entropy (BCE) loss between the ground truth and the predicted score maps. Multiple VFGN samples are used during both training and inference to compute the mean and the variance in the target and search region features, which are later combined by using an uncertainty-aware cross-correlation module using one of the processes described above.

#### IV. EXPERIMENTS

We use the KITTI [38] tracking dataset to train and test models. Following the standard protocol, we use KITTI tracking training subset for both training and testing, as the test subset does not provide the initial ground truth positions. The tracks  $[0, \dots, 18]$  are used for training and validation, and tracks 19 and 20 are used to test the trained models. Model performance is computed using the Precision and Success [19] metrics, which are based on the predicted and ground truth objects’ center difference and 3D Intersection Over Union, respectively. VPIT uses a pre-trained PointPillars network to initialize its pseudo image generation and FGN modules. To follow the same procedure, we train a VNNs version of PointPillars on the KITTI [38] detection dataset, use it to initialize the VPIT model and train the corresponding model for 64,000 steps with different number of training VFGN samples per step in  $[1, \dots, 20]$  range. For the double similarity uncertainty utilization method, we test the variance weight hyperparameter  $\lambda$  in the range of  $\{0.1, \dots, 1\}$  with step 0.1 and for the uncertainty penalization approach, the  $\rho$  value is tested for the range of  $\{2, \dots, 5\}$  with step 1. The best models are reported for each uncertainty estimation method.

Table I contains the evaluation results of regular VPIT and the Variational VPIT (VVPIT) models with different ways to utilize the predicted uncertainty. We report the best-performing models for each uncertainty utilization process, which are obtained by using 20 samples of the VFGN module. By computing the average of predictions and discarding the variances, VVPIT achieves higher tracking performance compared to the

TABLE I: Precision and Success values on the KITTI single object tracking experiments for VPIT and Variational VPIT (VVPIT) models.

Method	Uncertainty	Success	Precision
VPIT	-	50.49	64.53
VVPIT	averaging	51.97	66.69
VVPIT	double similarity	52.62	66.56
VVPIT	uncertainty penalization	<b>53.30</b>	<b>67.79</b>

VPIT model. By utilizing uncertainties, the Success and Precision values are further improved. Both double similarity and uncertainty penalization processes lead to better models, but the penalization process leads to a better tracking performance.

## V. CONCLUSIONS

In this paper, we proposed a method to utilize uncertainty in 3D Single Object Tracking which uses a Variational Neural Network (VNN) based version of the VPIT 3D Single Object Tracking method to estimate uncertainty in target and search features and combines these features with an uncertainty-aware cross-correlation module. We proposed two ways to utilize uncertainty in cross-correlation, i.e., by double similarity which adds a similarity in uncertainties to the regular cross-correlation, and by uncertainty penalization which penalizes uncertain features to shift focus to the more reliable feature channels. Additionally, we tested the model's performance without exploiting the estimated uncertainties, as it still leads to a statistically better model compared to regular VPIT. The use of VNNs improves the tracking performance of VPIT in all cases, with the uncertainty penalization leading to the best Success and Precision values. The estimated uncertainty can also be used to alert a decision system whether the model should be trusted, or a human expert or a more accurate but a computationally-heavier model should be deployed to avoid failures. This approach to uncertainty estimation is a direction for future work and can be built on top of the proposed architecture.

## ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

## REFERENCES

- [1] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, 2010, pp. 2544–2550.
- [2] Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.
- [3] David Held, Sebastian Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," *1604.01802*, 2016.
- [4] Zheng Fang, Sifan Zhou, Yubo Cui, and Sebastian Scherer, "3d-siamrpn: An end-to-end learning method for real-time 3d single object tracking using raw point cloud," *Sensors*, vol. 21, no. 4, pp. 4995–5011, 2021.

- [5] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr, "Fully-convolutional siamese networks for object tracking," *arXiv:1606.09549*, 2016.
- [6] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018, pp. 8971–8980.
- [7] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," *arXiv:1812.11703*, 2018.
- [8] Illia Oleksiienko, Paraskevi Nousi, Nikolaos Passalis, Anastasios Tefas, and Alexandros Iosifidis, "Vpit: Real-time embedded single object 3d tracking using voxel pseudo images," *arXiv:2206.02619*, 2022.
- [9] Di Feng, Lars Rosenbaum, and Klaus Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *ITSC*, 2018, pp. 3266–3273.
- [10] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *CVPR*, 2019, pp. 12677–12686.
- [11] Gregory P. Meyer and Niranjana Thakurdesai, "Learning an uncertainty-aware object detector for autonomous driving," in *IROS*, 2020, pp. 10521–10527.
- [12] Yuanxin Zhong, Minghan Zhu, and Huei Peng, "Uncertainty-aware voxel based 3d object detection and tracking with von-mises loss," *arXiv:2011.02553*, 2020.
- [13] Jianren Wang, Siddharth Ancha, Yi-Ting Chen, and David Held, "Uncertainty-aware self-supervised 3d data association," in *RSJ*, 2020, pp. 8125–8132.
- [14] Ben Daubney and Xianghua Xie, "Tracking 3d human pose with large root node uncertainty," in *CVPR*, 2011, pp. 1321–1328.
- [15] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza, "A general framework for uncertainty estimation in deep learning," *RA-L*, vol. 5, no. 2, pp. 3153–3160, 2020.
- [16] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *JMLR*, 2016, vol. 48, pp. 1050–1059.
- [17] Ian Osband, Zheng Wen, Mohammad Asghari, Morteza Ibrahimi, Xiyuan Lu, and Benjamin Van Roy, "Epistemic Neural Networks," *arXiv:2107.08924*, 2021.
- [18] Illia Oleksiienko, Dat Thanh Tran, and Alexandros Iosifidis, "Variational neural networks," *arxiv:2207.01524*, 2022.
- [19] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [20] Jakob Gawlikowski, Cedric Rovile Njoutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu, "A survey of uncertainty in deep neural networks," *arxiv:2107.03342*, 2021.
- [21] Murat Sensoy, Lance Kaplan, and Melih Kandemir, "Evidential deep learning to quantify classification uncertainty," in *NeurIPS*, 2018, pp. 3183–3193.
- [22] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight Uncertainty in Neural Networks," in *JMLR*, 2015, vol. 37, pp. 1613–1622.
- [23] Martin Magris and Alexandros Iosifidis, "Bayesian learning for neural networks: an algorithmic survey," *Artificial Intelligence Review*, 2023.
- [24] Ian Osband, John Aslanides, and Albin Cassirer, "Randomized prior functions for deep reinforcement learning," in *NeurIPS*, 2018, vol. 31, pp. 8626–8638.
- [25] Matias Valdenegro-Toro, "Deep sub-ensembles for fast uncertainty estimation in image classification," *arxiv:1910.08168*, 2019.
- [26] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren, "Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation," in *BrainLes*. 2018, vol. 11384, pp. 61–72, Springer.
- [27] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019.
- [28] Ibrahem Kandel and Mauro Castelli, "Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset," *Health Inf. Sci. Syst.*, vol. 9, no. 1, pp. 33, 2021.

- [29] Illia Oleksiienko, Dat Thanh Tran, and Alexandros Iosifidis, “Variational neural networks implementation in pytorch and jax,” *Software Impacts*, vol. 14, pp. 100431, 2022.
- [30] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao, “P2b: Point-to-box network for 3d object tracking in point clouds,” *arXiv:2005.13888*, 2020.
- [31] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, and Shuguang Cui, “Box-aware feature enhancement for single object tracking on point clouds,” in *ICCV*, 2021, pp. 13199–13208.
- [32] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui, “Ptt: Point-track-transformer module for 3d single object tracking in point clouds,” *arXiv:2108.06455*, 2021.
- [33] Shan Jiayao, Sifan Zhou, Yubo Cui, and Zheng Fang, “Real-time 3d single object tracking with transformer,” *IEEE Trans Multimedia*, 2022.
- [34] Jesus Zarzar, Silvio Giancola, and Bernard Ghanem, “Efficient bird eye view proposals for 3d siamese tracking,” *arXiv:1903.10168*, 2020.
- [35] Florian Kraus and Klaus Dietmayer, “Uncertainty estimation in one-stage object detection,” in *ITSC*, 2019, pp. 53–60.
- [36] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.
- [37] Dawei Zhang, Yanwei Fu, and Zhonglong Zheng, “UAST: Uncertainty-aware siamese tracking,” in *ICML*, 2022, vol. 162, pp. 26161–26175.
- [38] Andreas Geiger, Philip Lenz, and Raquel Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *CVPR*, 2012, pp. 3354–3361.
- [39] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom, “PointPillars: Fast Encoders for Object Detection from Point Clouds,” in *CVPR*, 2019.