

Can Transformer Attention Spread Give Insights Into Uncertainty of Detected and Tracked Objects?

Felicia Ruppel^{1,2}, Florian Faion¹, Claudius Gläser¹ and Klaus Dietmayer²

Abstract—Transformers have recently been utilized to perform object detection and tracking in the context of autonomous driving. One unique characteristic of these models is that attention weights are computed in each forward pass, giving insights into the model’s interior, in particular, which part of the input data it deemed interesting for the given task. Such an attention matrix with the input grid is available for each detected (or tracked) object in every transformer decoder layer. In this work, we investigate the distribution of these attention weights: How do they change through the decoder layers and through the lifetime of a track? Can they be used to infer additional information about an object, such as a detection uncertainty? Especially in unstructured environments, or environments that were not common during training, a reliable measure of detection uncertainty is crucial to decide whether the system can still be trusted or not.

I. INTRODUCTION

Object detection and tracking are essential tasks in a perception pipeline for autonomous and automated driving. Only with knowledge about surrounding objects, downstream tasks, such as prediction and planning, are possible. In such a system, where the cascading effects of perception errors can be detrimental, it is very important to be able to quantify the reliability of the detection and tracking output. In object detection, uncertainty can stem from two sources [1]: *Epistemic* uncertainty is caused by uncertainty of the model, e.g. when an observation is made that was not present in the training dataset. Unstructured and dynamic environments can also cause such an uncertainty, as their versatility can hardly be captured in a training dataset. Second, *aleatoric* uncertainty stems from sensor noise, and also encompasses uncertainty caused by low visibility and increased distance from the sensor [1].

While state-of-the-art object detection methods have been based on deep learning for many years, both with image input [2] as well as on point clouds [3], [4], it is a recent phenomenon that deep learning based models are also used for joint tracking and detection [5], [6], [7]. Such trackers aim to utilize the detector’s latent space to infer additional information about a tracked object, rather than relying on low-dimensional bounding boxes as input. However, they have the drawback that they are unable to output an uncertainty, as a conventional method would, e.g. tracking based on a Kalman filter [8]. While deep learning based detectors and trackers usually output a confidence score or class probability

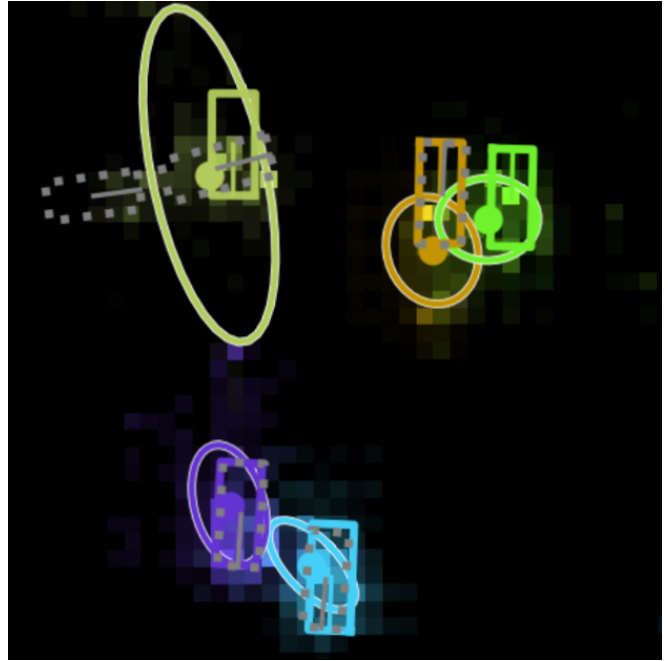


Fig. 1. Example of estimated bounding boxes with their respective attention covariance matrices, pictured as ellipses. Ground truth boxes are denoted by dotted grey lines, while estimated boxes, attention weights, attention mean and ellipses are colored. Excerpt from the birds-eye-view grid at a distance of 30 to 50 meters from the ego vehicle.

score per estimated object, these generally can not be used as a reliable uncertainty measure, but additional measures are necessary to capture uncertainty [9].

One approach towards joint object detection and tracking is the usage of transformer models [10], which were able to achieve state-of-the-art results in some domains [11], [6]. Transformers are based on attention, i.e. the interaction between input tokens, which is why these models allow for a unique insight into their reasoning: One can visualize the attention matrices that are computed in each model forward pass and investigate which part of the input data was used to generate a certain output. In previous work, we developed a transformer based model for detection and tracking [7] in the context of autonomous driving that operates on (lidar) point clouds. An example of visualized attention weights from the tracking model are pictured in Figure 1. In empirical observations, a more focused attention tends to lead to a more accurate detection. Therefore, we investigate whether the attention weight distribution can give insights into a detection uncertainty in this paper. An uncertainty indicator would be very valuable towards the ability to use transformer based

¹Robert Bosch GmbH, Corporate Research, 71272 Renningen, Germany, {firstname.lastname}@de.bosch.com

²Institute of Measurement, Control and Microtechnology, Ulm University, Germany, {firstname.lastname}@uni-ulm.de

methods in safety-critical use cases, such as autonomous driving.

The main contributions of this work are:

- We propose a new metric, namely attention spread, to quantify the distribution of attention weights.
- We analyze whether attention spread is an indicator for uncertainty by comparing it to the observed detection accuracy in terms of Intersection-over-Union (IoU) with ground truth bounding boxes.
- We analyze the spatial and temporal dependencies of the observed attention weight matrices, both in the context of object detection as well as tracking.

II. BACKGROUND

A. Attention in Transformers

The transformer model was first introduced in [10] in the context of natural language processing, but has since been applied in many fields, such as computer vision [12], [13]. The original model consists of an encoder and a decoder. In the encoder, a sequence of tokens is input, which can be encoded words, pixels, or grid cells, depending on the usage. These input tokens \mathbf{x}_i (feature vectors), $i = 1, \dots, N$, can interact with one another through *self-attention*. For this, each of them is transformed into three unique vectors via learnt mappings: a query \mathbf{q}_i , a key \mathbf{k}_i and a value \mathbf{v}_i . Now, attention weights are computed by comparing each of the queries with each of the keys in terms of their dot product, obtaining scalar weights $w_{ij} = \mathbf{q}_i^\top \mathbf{k}_j$, which result in an *attention weight matrix* of size $N \times N$. The output of the self-attention layer is obtained by a weighted sum of the values $\sum_{j=1}^N w_{ij} \mathbf{v}_j$, $i = 1, \dots, N$, whereas computations are commonly performed in matrix form to increase efficiency and a softmax is applied to the weights to normalize their sum to 0.

In the transformer decoder on the other hand, it is common to input two sequences: the encoder’s output of length N , as well as a sequence of query vectors of length M . During *self-attention* in the decoder, the queries attend to one another, while *cross-attention* layers allow an interaction between the two sequences: In the attention module, the keys and values are now computed from one sequence, while the queries stem from the other. The attention weight matrices of size $N \times M$ during cross-attention in the transformer decoder are the focus of this paper.

B. Transformers for Object Detection and Tracking

The detection transformer (DETR) [12] was one of the first models to utilize a transformer for object detection. It has been adapted and extended in multiple ways, for example for joint tracking and detection [6]. Our model from previous work [7], which is the subject of the analysis in this paper, is based on the aforementioned approaches, but with a focus on applicability to large point clouds, such as those, which are commonly measured with automotive lidar sensors. The model is introduced in the following.

An overview of the joint detection and tracking model is pictured in Figure 2, with the detector on the left (a).

This detector can either be used as a standalone model, i.e. without tracking, or it serves as a track initializer on the first frame of a sequence. The input point cloud is processed through a backbone, Pointpillars [3]. This could be replaced by any pretrained backbone that encodes the input into a grid (or sequence) of feature vectors. A positional encoding is added before the encoded input is passed to the transformer decoder as an unordered sequence. Note that the transformer *encoder* is left out in this model, allowing for a comparably smaller GPU memory requirement. Therefore, only the backbone is available to provide context-encoding functionality. The second sequence that is passed to the decoder is denoted anchor-based object queries. M queries are generated, which is more than the number of objects that are expected to appear in one frame. Each of them serves as a slot for a possible object and is able to access the input data through cross-attention, as well as interact with the other object queries during self-attention. Following [14], [15], the query encoding is computed from prior locations, which are sampled from the input point cloud using farthest point sampling. This is meant to achieve an even spread of queries over the birds-eye-view grid, while not placing queries in areas where no lidar data is available (e.g. behind a large building). The object queries are transformed through K decoder layers. To the resulting feature vectors, a regression and classification head is applied to obtain bounding box parameters of detected objects, whereas some are assigned to the ‘no-object’ class, since there are generally more queries than objects in the scene.

The focus of this paper are the cross-attention matrices in the decoder, of size $N \times M$. Since we operate on a square grid that is output by the backbone, we reshape the matrices to $\sqrt{N} \times \sqrt{N} \times M$, so that the first two dimensions correspond to indices on the birds-eye-view grid. For each object that is output by the model, one can obtain K attention matrices of shape $\sqrt{N} \times \sqrt{N}$ (one per decoder layer), giving an insight, which input data the respective query accessed to detect this object.

In Figure 2 (b), the object tracker [7] is pictured. In addition to the object queries, track queries are passed to the decoder, which serve as slots for continued tracks. These stem from the model’s output at the previous timestep and contain information about an object in feature space. They are transformed through an ego-motion compensation module (EMC) to correct the ego-motion between frames. For these track queries, the aforementioned attention weight matrices can be obtained as well.

III. PROPOSED METHOD

A. Modelling the attention distribution

The cross-attention weights between one object or track query and the input birds-eye-view grid in a given decoder layer are reshaped to a matrix of shape $\sqrt{N} \times \sqrt{N}$, as introduced above. It contains attention weights w_{pq} , $1 \leq p, q \leq \sqrt{N}$, where p and q correspond to indices on the birds-eye-view grid. The largest K weights are selected, i.e. $p, q \in S^K$. By definition of the birds-eye-view grid, each

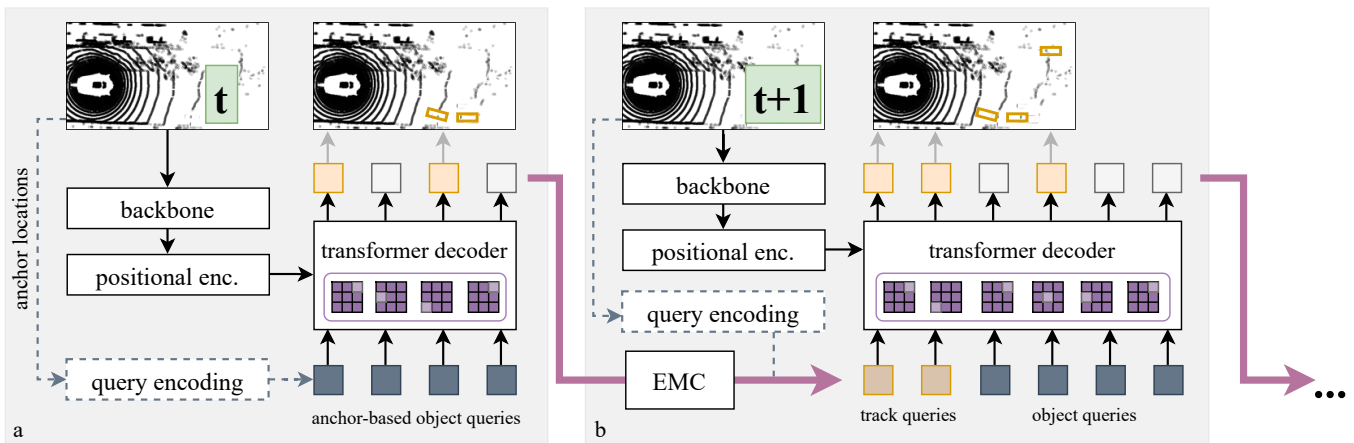


Fig. 2. Detection and tracking model overview. The model design stems from previous work [7]. The focus of this paper lies with the analysis of the *attention matrices* within the transformer decoder, which are illustrated in purple. a. Object detection model that can either be used as a standalone detector, or as track initializer in the first frame of a sequence. The input point cloud is processed through a backbone and a positional encoding is added. M locations are sampled from the input point cloud and an object query is generated from each. In the transformer decoder, the queries can each aggregate information about one object, through cross-attention with the input feature map. b. Tracking model. Those feature vectors from the previous frame that belong to an object to be tracked are passed into the transformer decoder as track queries, in addition to the object queries. To compensate ego-motion between frames, they are transformed in feature space through an ego-motion compensation module (EMC).

horizontal index q is associated with a location in x -direction x_q , and each index p with one in y -direction, y_p . We define the mean of top- K attention as

$$\mu_K = \frac{1}{W} \sum_{p,q \in S^K} w_{p,q} \begin{pmatrix} x_q \\ y_p \end{pmatrix}, \quad (1)$$

where $W = \sum_{p,q \in S^K} w_{p,q}$. We propose to quantify attention covariance as follows:

$$C_K = \frac{1}{W} \sum_{p,q \in S^K} w_{p,q} \begin{bmatrix} x_q \\ y_p \end{bmatrix} - \mu_K \begin{bmatrix} x_q \\ y_p \end{bmatrix} - \mu_K^\top. \quad (2)$$

This covariance matrix can be reduced to a single value, namely the proposed attention spread (AS)

$$AS = \det C_K. \quad (3)$$

It is difficult to quantify 'ground truth' uncertainty of a deep learning model. In this work, we quantify detection accuracy in terms of intersection-over-union (IoU) between estimated boxes and ground truth. IoU has been shown to correlate with epistemic uncertainty [1]. Aleatoric uncertainty, on the other hand, arises with low visibility and object distance [16], among other causes.

B. Attention shift and focus

In the detector, an object query passes through K decoder layers. Before the first layer, it is assigned an anchor location, from which its encoding is computed and that serves as a prior location for object detection. One would expect that the attention between the query and the input data in the first layer is quite broad, since it might move away to find a nearby object, while still located around the prior location. As the object query continues to focus on one object through the following layers, the resulting attention is expected to be more focused around that object. We propose to test

these assumptions by collecting the aforementioned attention distribution parameters for each decoder layer.

Another related question of interest is how the attention focus changes during the lifetime of a track.

IV. RESULTS

For this analysis, we set $M = 100$, $\sqrt{N} = 128$, $K = 6$, $k = 100$, and use the pretrained detection and tracking model as introduced in previous work [7]. The experiments are performed on the nuScenes `val` dataset and only the class 'car' is considered. The model is either used in detection mode, i.e. without passing temporal information to the following frame, or in tracking mode, by passing track queries to the model in addition to the object queries. Only bounding boxes are considered that the model itself classified as relatively confident, with a detection score above 0.8, since it is common in this model that many queries are idle.

A. Detection and tracking accuracy

In Figure 3, attention spread is plotted in terms of intersection-over-union (IoU). For this, the IoU between the estimated bounding boxes and their closest ground truth object was computed. A large IoU corresponds to a more accurately detected object. IoU values of 0 (no overlap) were removed. The IoU values were grouped into 10 bins of width 0.1, and the corresponding attention spread obtained from the last decoder layer. The median attention spread per bin was plotted, as well as 25th and 75th percentiles as error bars. The plots are very similar for object detection (top) and joint detection and tracking (bottom). The observed median attention spread sinks with growing IoU. This means that low attention spread can be an indicator for high IoU and vice-versa. Therefore, attention spread may also indicate epistemic uncertainty [1], however the ranges between 25th and 75th percentiles are quite large.

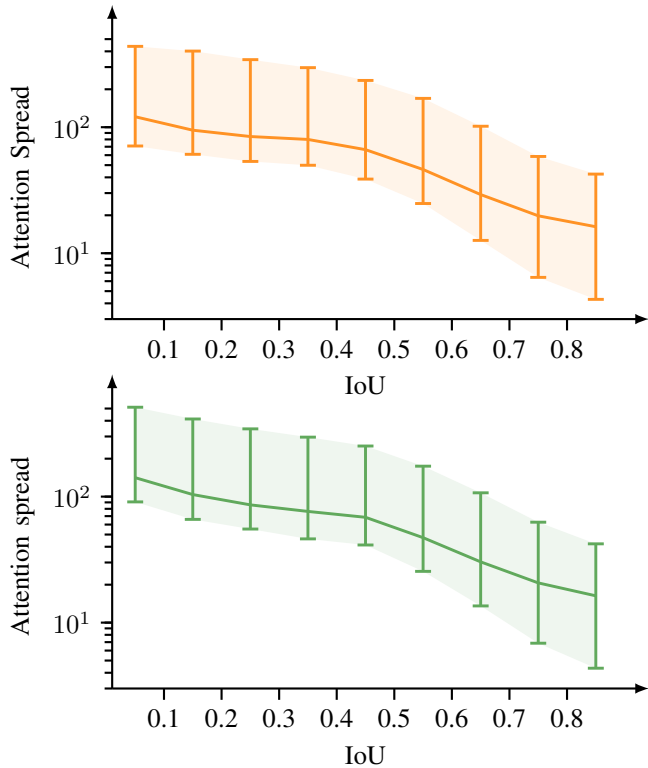


Fig. 3. Median attention spread in terms of intersection-over-union between the estimated bounding box and the closest ground truth box, with 25th and 75th percentiles as error bars. IoU values were grouped into bins of width 0.1, while those smaller than 0 were excluded. *Top*: Object detection *Bottom*: Joint detection and tracking.

B. Object distance to the ego vehicle

To see whether attention spread changes with the distance of estimated bounding boxes to the ego vehicle, the birds-eye-view is divided into 20×20 bins. For each bin, estimated bounding boxes located there, as well as their respective attention spread in the last layer of the detector, are aggregated. Their mean per bin over all frames in the nuScenes `val` dataset is pictured in Figure 4. It is observable that mean attention spread increases with distance to the ego vehicle. In this regard, it behaves similarly to aleatoric uncertainty [16].

C. Attention through multiple decoder layers

To investigate how attention spread changes through the decoder layers, its values observed during object detection are collected per layer for each observed object. Median attention spread per layer is pictured in Figure 5, as well as its 25th and 75th percentiles. The mean attention spread in the last layer is smaller than in the first, which is in line with the expectations. However, the value is quite small in the second layer and rises again in the following, which poses the question whether there exists a specific reason for this early focusing and later broadening of the cross-attention.

D. Attention during a track’s lifetime

Observing many tracked objects, we find that the attention spread of a tracked object changes over time. One exemplary

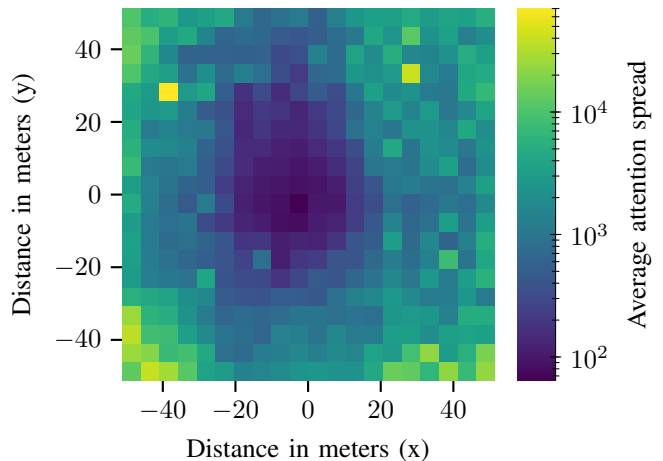


Fig. 4. Attention spread in terms of the locations of estimated bounding boxes in birds-eye-view. Locations were grouped into 20×20 bins and their respective attention spreads averaged.

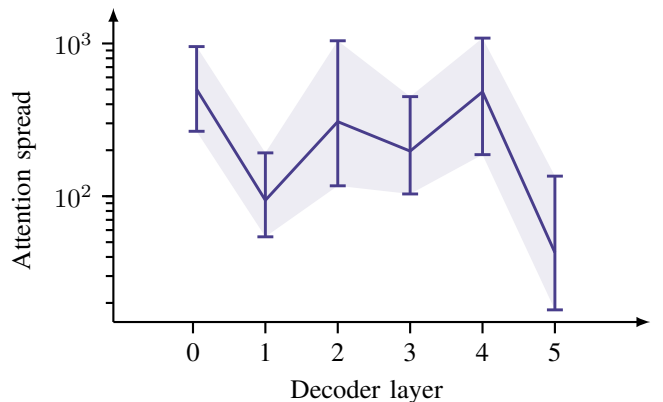


Fig. 5. Median attention spread per layer in the transformer decoder. The error bars denote 25th and 75th percentiles.

visualization of the initialization phase of a track is pictured in Figure 6. When the lime green track is initialized at a distance of ca. 50 meters, it barely overlaps with the field of view. Its attention spread is quite large (displayed in terms of an ellipse that is derived from the covariance matrix). As it moves closer over time, its attention spread decreases.

In Figure 7 (left), the median attention spread in terms of elapsed time since track initialization is pictured. For this, the first 3.5 seconds of all tracks, as output by the model on the nuScenes `val` dataset, were considered if the total length of the respective track was at least 7 seconds. In Figure 7 (right), the median attention spread in terms of the remaining time until track finalization is pictured. In both figures, error bars denote the 25th and 75th percentiles.

We observe that both in the track initialization phase as well as in the finalization phase a change in median attention is visible. Multiple factors may influence this behavior: Firstly, tracks are often initialized and finalized at a large distance to the ego vehicle, when an object first enters and last exists the field of view, respectively. In that regard, the observations are similar to those in Figure 4. Second, tracks

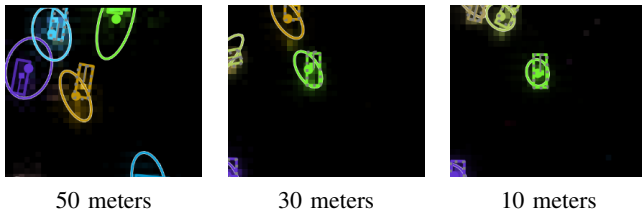


Fig. 6. Exemplary attention during the lifetime of one track (light green). The track is initialized at a distance of ca. 50 meters and then moves closer towards the ego vehicle.

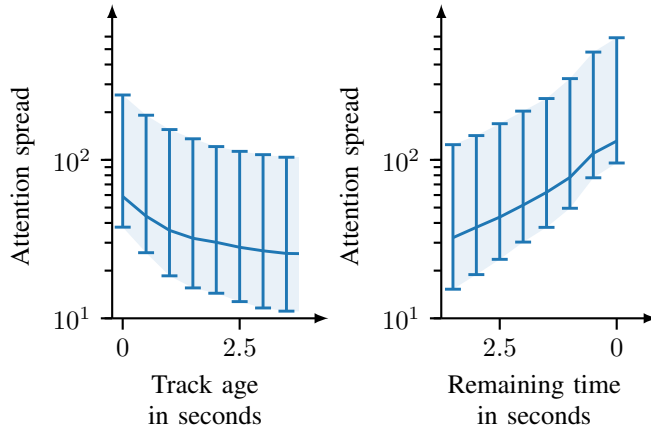


Fig. 7. Median attention spread in terms of track age, with 25th and 75th percentiles as error bars. *Left*: Track initialization phase, consisting of the first 3.5 seconds per track. *Right*: Track finalization phase, considering the last 3.5 seconds per track. Note that only tracks, which reached a final duration of at least 7 seconds, were included.

may increase in confidence after a certain initialization time, since the model can utilize past observations, encoded in the track queries, for its reasoning.

V. CONCLUSION

In this paper, we investigated the distribution of the attention weight matrices in the transformer decoder in the context of object detection and tracking and proposed a new metric to quantify it. We found that median attention spread decreases with larger IoU, while it increases with larger distance to the ego vehicle. This indicates that attention spread may be able to offer insights into both aleatoric and epistemic uncertainty, whereas it is not possible to differentiate between the two from attention spread alone. Besides this, attention spread observed during the lifetime of tracks changes over time, especially in the initialization and finalization phase.

Our findings open up questions for further research: Is attention spread the best way to describe the attention weight distribution? Can a measure with less variance be found? Can the model design be improved or the model behavior be better understood based on the knowledge about attention spread per layer?

We conclude that the attention matrices available in transformer models have the potential to give insights into detection and tracking uncertainty and that further research is promising.

REFERENCES

- [1] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection," in *IEEE Int. Conf. Intell. Transp. Syst.*, 2018, pp. 3266–3273.
- [2] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, December 2015.
- [3] A. Lang *et al.*, "Pointpillars: Fast Encoders for Object Detection from Point Clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 697–12 705.
- [4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [5] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3569–3577.
- [6] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "TrackFormer: Multi-Object Tracking with Transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 8844–8854.
- [7] F. Ruppel, F. Faion, C. Gläser, and K. Dietmayer, "Transformers for Multi-Object Tracking on Point Clouds," in *IEEE Intell. Veh. Symp.*, 2022, pp. 852–859.
- [8] H.-K. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3D Multi-Object Tracking for Autonomous Driving," *arXiv:2001.05673*, 2020.
- [9] A. Loquercio, M. Segu, and D. Scaramuzza, "A General Framework for Uncertainty Estimation in Deep Learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.
- [10] A. Vaswani *et al.*, "Attention is All You Need," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Int. Conf. Learn. Represent.*, 2021.
- [12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 213–229.
- [13] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Int. Conf. Learn. Represent.*, 2021.
- [14] I. Misra, R. Girdhar, and A. Joulin, "An End-to-End Transformer Model for 3D Object Detection," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 2906–2917.
- [15] F. Ruppel, F. Faion, C. Gläser, and K. Dietmayer, "Transformers for Object Detection in Large Point Clouds," in *IEEE Int. Conf. Intell. Transp. Syst.*, 2022.
- [16] S. Wirges, M. Reith-Braun, M. Lauer, and C. Stiller, "Capturing object detection uncertainty in multi-layer grid maps," in *IEEE Intell. Veh. Symp.*, 2019, pp. 1520–1526.