InterTrack: Interaction Transformer for 3D Multi-Object Tracking

John Willes  Cody Reading  Steven L. Waslander
University of Toronto Robotics Institute
{johnwilles, codyreading}@gmail.com, steven.waslander@robotics.utias.utoronto.ca

Abstract—3D multi-object tracking (MOT) is a key problem for autonomous vehicles, required to perform well-informed motion planning in dynamic environments. Particularly for densely occupied scenes, associating tracks to new detections remains challenging as existing systems tend to omit critical contextual information. Our proposed solution, InterTrack, introduces the Interaction Transformer for 3D MOT to generate discriminative object representations for data association. We extract state and shape features for each track and detection, and efficiently aggregate global information via attention. We then perform a learned regression on each track/detection feature pair to estimate affinities, and use a robust two-stage data association and track management approach to produce the final tracks. We validate our approach on the nuScenes 3D MOT benchmark, where we observe significant improvements, particularly on classes with small physical sizes and clustered objects. As of submission, InterTrack ranks 1st in overall AMOTA among methods using CenterPoint [1] detections.

I. INTRODUCTION

3D multi-object tracking (MOT) is a vital task for many fields such as autonomous vehicles and robotics, enabling systems to understand their environment in order to react accordingly. A common approach is the tracking-by-detection paradigm [2], [1], [3], [4], in which trackers consume independently generated 3D detections as input. Existing tracks are matched with new detections at each frame, by estimating a track/detection affinity matrix and matching high affinity pairs. Incorrect associations can lead to identity switching and false track initialization, confusing decision making in subsequent frames.

Affinities are estimated by extracting track and detection features, followed by a pairwise comparison to estimate affinity scores. Feature extraction can simply extract the object state [2], [3] or use a neural network to extract features from sensor data [5], [6]. For effective association, features should be discriminative such that feature comparison results in accurate affinity estimation [7].

3D MOT methods often perform feature extraction independently for each track and detection. Independent methods, however, tend to suffer from high feature similarity, particularly for densely clustered objects (see Figure 1). Similar features are non-discriminative, and lead to ambiguous data association as the correct match cannot be distinguished from the incorrect match hypotheses.

Feature discrimination can be improved through learned feature interaction modeling [7]. Interaction modeling allows individual features to affect one another, adding a capacity for the network to encourage discrimination between matching and non-matching feature pairs. Previous methods [7], [5] have adopted a graph neural network (GNN) to model feature interactions, but limit the connections to maintain computational efficiency. We argue all interactions are important, as for example, short range interactions can be helpful for differentiating objects in dense clusters, while long range interactions can assist with fast-moving objects with large motion changes.

Additionally, most prior 3D MOT works [1], [6] lack duplicate track filtering. Duplicate tracks can be an issue for trackers that ingest detections [1] with high recall and many duplicate false positives, leading to reduced performance.

To resolve the identified issues, we propose InterTrack, a 3D MOT method that introduces the Interaction Transformer to 3D MOT, in order to generate discriminative affinity estimates in an end-to-end manner, and introduces a track rejection module. We summarize our approach with three contributions.

![Figure 1: 3D tracking visualization. (a) When object interactions are not considered, the highlighted track produces multiple similar affinity scores leading to an incorrect match at $t = 1$. The track is killed at $t = 2$, resulting in an identity switch (shown as new color). (b) Interaction modeling encourages a single highest affinity for the correct detection resulting in an uninterrupted track.](image-url)
(1) Interaction Transformer. We introduce the Interaction Transformer for 3D MOT, a learned feature interaction mechanism leveraging the Transformer model. The Interaction Transformer models spatio-temporal interactions between all object pairs, leveraging attention layers [8] to maintain high computational efficiency. Through complete interaction modeling, we encourage feature discrimination between all object combinations, leading to increased overall feature discrimination and tracking performance shown in Section IV-C.

(2) Affinity Estimation Pipeline. We design a novel method to estimate track/detection affinities in an end-to-end manner. Using detections and LiDAR point clouds, our method extracts full state and shape information for each track and detection, and aggregates contextual information via the Interaction Transformer. Each track/detection feature pair is used to regress affinity scores. Our learned feature extraction offers increased representational power over simply extracting the object state and allows for interaction modeling.

(3) Track Rejection. We introduce a duplicated track rejection strategy, by removing any tracks that overlap greater than a specified 3D intersection-over-union (IoU) threshold. We validate the track rejection module leads to performance improvements as demonstrated in Section IV-C.

InterTrack is shown to rank 1st on the nuScenes 3D MOT test benchmark [9] among methods using public detections, with margins of 1.00%, 2.20%, 4.70%, and 2.60% AMOTA on the Overall, Bicycle, Motorcycle, and Pedestrian categories, respectively.

II. RELATED WORK

Transformer. The Transformer [8] has become the standard model of choice in natural language processing [10], [11], [12]. Dependencies between sentence elements are modelled through an attention mechanism, leading to improved computational efficiency and performance. Transformers have seen use in computer vision [13], [14], [15], which model interactions between pixel-level image features. Transformers have also seen use in 3D prediction [16] wherein interaction modelling between objects leads to improved prediction. In this work, we leverage the Transformer to model the interactions between all tracks and detections for improved tracking.

Image Feature Matching. Image feature matching consists of matching points between image pairs, often used in applications such as simultaneous localization and mapping (SLAM) and visual odometry. Feature descriptors are extracted from local regions surrounding each point, in order to match points with the highest feature similarity. Feature extraction can be performed by hand-crafting features [17], [18], [19] or with a learned convolutional neural network (CNN) [20], [21]. SuperGlue [22] and LoFTR [15] introduce a GNN and Transformer module respectively, as a means to incorporate both local and global image information during feature extraction. We follow the Transformer design of LoFTR [15] to incorporate global object information in our feature extraction.

Multi-Object Tracking. MOT is a key focus in 2D [23] and 3D [2] settings. Tracking methods either follow a tracking-by-detection approach [24], [25], wherein object detection and track estimation are performed separately, or perform joint detection and tracking in a single stage [26], [27], [28], [29].

In 2D MOT, recent methods [27], [30] leverage the Transformer for joint detection and tracking. A drawback is the lack of explicit affinity estimates, resulting in less interpretability often required by safety-critical systems.

In 3D MOT, the tracking-by-detection paradigm [31], [6] is dependent on high-quality, high-frequency detections. These trackers solve an association problem [32], [33], either matching detections to existing tracks or used to birth new tracks. Similar to feature matching methods, tracking-by-detection methods extract object features and form matches with the highest feature similarity [34], [35]. Affinity matrices are estimated that represent the feature similarity between tracks and detections, and are fed to either a Greedy or Hungarian [36] matching algorithm to generate matches. Based on how data association is performed, trackers can be divided into two categories: heuristic and learned association methods.

Heuristic-based methods compute affinity scores using a heuristic similarity or distance metric. The metric is based on the object state, with some metrics including 3D IoU, Euclidean distance, or Mahalanobis distance [37]. AB3DMOT [2] uses 3D IoU as the similarity metric and a Kalman filter for track prediction and update. Introduced by CenterPoint [1], methods [38], [39] can compute affinity using Euclidean distance, and replace the Kalman filter prediction with motion prediction based on learned velocity estimates. EagerMOT [3] performs separate association stages for LiDAR and camera streams, and modifies the Euclidean distance to include orientation difference. Without contextual information, heuristic-based affinity estimation leads to many similar affinity scores and ambiguous data association, notably for clustered objects.

Learning-based methods use a neural network to extract object features from detection and sensor data, followed by either a distance metric [31] or a learned scoring function [7], [6] to estimate pairwise affinities. GNN3DMOT [7] and PTP [5] utilize a graph neural network (GNN) to model object interactions, with object features represented as GNN nodes and edge features used to estimate affinities. Sparse edges construction is required for computational efficiency. Chiu et al. [6] combine both a learned scoring function with the Mahalanobis distance [37] to generate affinity scores. AlphaTrack [31] improves the CenterPoint [1] detector by
fusing LiDAR and image information, and extracts object appearance information for data association. Current learned association methods either extract features independently or with sparse interaction modeling, limiting feature discrimination and region of influence that is considered during matching.

InterTrack directly addresses the limitation of interaction modeling for data association, by introducing the Interaction Transformer to 3D MOT. Doing so allows for complete interaction modeling, leading to improved object feature discrimination. InterTrack only leverages the Transformer for data association to provide an interpretable solution for safety-critical applications in autonomous vehicles.

III. METHODOLOGY

InterTrack learns to estimate affinity matrices for data association, followed by a 3D tracking module to estimate tracks. An overview of InterTrack is shown in Figure 2.

A. Affinity Learning

Our network learns to estimate discriminative affinity matrices for effective data association. Taking object detections and LiDAR point clouds as input, we extract object-level features for both existing tracks and new detections. Interaction information is embedded in the track/detection features with a Transformer model, followed by an affinity estimation module to regress affinity scores.

Object Feature Extraction. We extract features for both tracks and detections, using a shared network to extract both state and shape features for each object. The inputs to the feature extractor are either track or detection states, \( T_{t-1} \in \mathbb{R}^{M \times 11} \) or \( D_t \in \mathbb{R}^{N \times 11} \), and frame corresponding LiDAR point clouds, \( P_{t-1} \in \mathbb{R}^{P \times 5} \) or \( P_t \in \mathbb{R}^{P \times 5} \). We extract state directly as \((x, y, z, l, w, h, \theta, \dot{x}, \dot{y}, c, s)\) including 3D location \((x, y, z)\), dimensions \((l, w, h)\), heading angle \(\theta\), velocity \((\dot{x}, \dot{y})\), class \(c\), and confidence \(s\), while shape information is extracted from the LiDAR point cloud, \(P\). We adopt the point cloud preprocessing and 3D feature extractor used in SECOND [40] (Voxel Backbone in Figure 2) to generate bird’s-eye-view (BEV) features, \(B\), where multiple LiDAR sweeps are combined and each point is represented as \((x, y, z, r, \Delta t)\) including 3D location \((x, y, z)\), reflectance \(r\), and relative timestep \(\Delta t\). 3D object states are projected into the BEV grid, \(B\), and used with ROI Align to extract shape features for each object. Let \(G(\cdot)\) represent a feed-forward network (FFN) consisting of 2 Linear-BatchNorm-Relu blocks, used to modify the number of feature channels. We apply a state FFN, \(G_{st}\), and a shape FFN, \(G_{sh}\) (State and Shape Layers in Figure 2) to modify the number of channels to \(C/2\), and then concatenate to form track or detection features, \(T_{t-1}^{\text{feat}} \in \mathbb{R}^{M \times C}\) or \(D_t^{\text{feat}} \in \mathbb{R}^{N \times C}\).

Interaction Transformer. The track and detection features, \(T_{t-1}^{\text{feat}}\) and \(D_t^{\text{feat}}\), are used as input to the Interaction Transformer (see Figure 2), producing interaction-aware track and
detection features, $\tilde{T}_{t-1}^{\text{feat}} \in \mathbb{R}^{M \times C}$ and $\tilde{D}_{t}^{\text{feat}} \in \mathbb{R}^{N \times C}$. We adopt the transformer module from LoFTR [15] and interleave the self and cross attention blocks $N_c$ times. In this work, we retain the value of $N_c = 4$. Self-attention blocks perform feature attenuation within a frame, therefore embedding spatial relationships through intra-frame interactions. Conversely, cross-attention blocks perform feature attenuation across frames, embedding temporal relationships through inter-frame interactions. Leveraging both self and cross attention provides a complete interaction modeling solution.

**Affinity Head.** The interaction-aware track and detection features, $\tilde{T}_{t-1}^{\text{feat}}$ and $\tilde{D}_{t}^{\text{feat}}$, are used to generate an affinity matrix, $A_t \in \mathbb{R}^{M \times N}$. Every feature pair is concatenated $R \times R \in (\text{Channel and G})$ $D \in A \times R$ times. In $\tilde{F} \in t \tau \in A \text{and rows/columns}$, that minimizes the $\hat{D}^{(3)}(\tilde{T}_t, \hat{A})$, is matched to $\hat{A}$ is the affinity matrix label. To generate the affinity matrix for pairs $1$ is the predicted velocity of a detection and $\dot{p}$ is the predicted velocity estimate from the detection as follows:

$$p_t = p_{t-1} + d_t \nu_{t-1},$$

where $p_t = (x_t, y_t)$ is the track centroid, $\nu_{t-1} = (\dot{x}_{t-1}, \dot{y}_{t-1})$ is the predicted velocity of a detection and $d_t$ is the timestep duration at time $t$. Discarding the Kalman prediction equations removes the covariance predictions required for the Kalman update equations. Therefore, we simply assign the detection state as the latest measurement.

**Track Overlap Rejection.** Due to the permissive structure of the two stage data association framework of EagerMOT, we note that duplicate tracks occur regularly in the output stream. Duplicate tracks often occur when ingesting detections [1] with many duplicates with high recall, often with low confidence scores. Detections with low confidence could be filtered to remove most duplicates, but would result in reduced track coverage due to a loss of true positive detections. Rather, we detect duplicates by computing the 3D IoU between all tracks at each frame, and flag any pairs of the same class with a 3D IoU greater than a rejection threshold, $\tau_{\text{rej}}$. For each pair, we reject the track with the lower age in order to reduce instances of identity switching.

**C. Training Loss**

We treat the affinity matrix estimation as a binary classification problem where each element represents the probability of a match between any given track, $T_{t-1}^m$, and detection, $D_{t}^n$. We use the binary focal loss [41] as supervision in order to deal with the positive/negative imbalance:

$$L = \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} \text{FL}(A_t(m, n), \hat{A}_t(m, n))$$

where $\hat{A}_t$ is the affinity matrix label. To generate the affinity label, each track, $T_{t-1}^m$, and detection, $D_{t}^n$, is matched to ground truth boxes via 3D IoU to retrieve an identity label. Only matches greater than a threshold, $\tau_{\text{gt}}$, are considered valid. We set $\hat{A}_t(m, n) = 1$ for pairs $(T_{t-1}^m, D_{t}^n)$ with the same identity and $\hat{A}_t(m, n) = 0$ for all remaining pairs.
Table I: 3D MOT results on the nuScenes test set [9], reporting methods that use CenterPoint [1] detections. We indicate the best result in bold and improvements relative to the next best method EagerMOT [3].

| Method               | AMOTA↑ | AMOTP↓ | Bicycle | Bus | Car | Motor | Ped | Trailer | Truck |
|----------------------|--------|--------|---------|-----|-----|-------|-----|---------|-------|
| AB3DMOT [2]          | 15.1   | 1.501  | 0.0     | 40.8| 27.8| 8.1   | 14.1| 13.6    | 1.3   |
| CenterPoint [1]      | 63.8   | 0.555  | 32.1    | 71.1| 82.9| 59.1  | 76.7| 65.1    | 59.9  |
| ProbTrack [6]        | 65.5   | 0.617  | 46.9    | 71.3| 83.0| 63.1  | 74.1| 65.7    | 54.6  |
| EagerMOT [3]         | 67.7   | 0.550  | 58.3    | 74.1| 81.0| 62.5  | 74.4| 63.6    | 59.7  |
| **InterTrack (Ours)**| **68.7**| **0.560**| **60.5**| **72.7**| **80.4**| **67.2**| **77.0**| **62.9**| **60.0** |
| **Improvement**      | +1.00 | +0.01  | +0.20   | -1.40| -0.60| -0.70 | +2.60| +0.70   | +0.30 |

IV. EXPERIMENTAL RESULTS

A. nuScenes Dataset Results

We use the official nuScenes 3D MOT evaluation [9] for comparison, reporting the AMOTA and AMOTP metrics. Table I shows the results of InterTrack on the nuScenes test set [9], compared to 3D MOT methods using CenterPoint [1] detections for fair comparison. We observe that InterTrack outperforms previous methods by a large margin on overall AMOTA (+1.00%) with a similar AMOTP (+0.01m).

InterTrack greatly outperforms previous methods on classes with smaller physical sizes, with considerable margins of +2.20%, +4.70%, and +2.60% on the Bicycle, Motorcycle, and Pedestrian classes. We attribute the performance increase to the dense clustering of smaller objects that exists in the nuScenes dataset. Heuristic metrics often have trouble distinguishing between clustered objects (see Figure 1), while our Interaction Transformer produces features with more discrimination leading to correct association. We reason heuristic metrics are sufficient on larger objects due to increased separation, as InterTrack lags behind prior methods on the Bus, Car, and Trailer classes with margins of -1.40%, -0.60%, and -0.70%, and outperforms the Truck class (+0.30%). We note InterTrack achieves 5 first place

D. Data Augmentation

As the 3D object detector and InterTrack are trained on the same training splits, the quality of detections encountered at training time will be higher than at test time. Therefore, we apply two augmentations during training to reduce the gap between training and test detections.

**Positional Perturbation.** To simulate positional error, we add a perturbation \((\delta_x, \delta_y)\) on the 2D position \((x, y)\) of detections by randomly sampling from a normal distribution:

\[
\begin{align*}
x &= x + \delta_x \quad \delta_x \sim N(0, \sigma_x^2) \\
y &= y + \delta_y \quad \delta_y \sim N(0, \sigma_y^2)
\end{align*}
\]

**Detection Dropout.** We employ detection dropout to simulate missing detections. At each frame, a fraction \(d\) of all detections are removed, where the fraction is randomly sampled from a uniform distribution \(d \sim U(d_{\text{min}}, d_{\text{max}})\).
Table II: 3D MOT results on the KITTI val set [42] for the Car class with the AB3DMOT [2] evaluation. † indicates methods using 3D detections from PointRCNN [47].

| Method                  | sAMOTA↑ | AMOTA↑ | AMOTP↑ |
|-------------------------|---------|--------|--------|
| AB3DMOT† [2]            | 91.8    | 44.3   | 77.4   |
| GNN3DMOT† [2]           | 93.7    | 45.3   | 78.1   |
| EagerMOT [3]            | 94.9    | 48.8   | 80.4   |
| InterTrack (Ours)       | 95.2    | 48.8   | 80.3   |
| Improvement             | +0.30   | +0.00  | -0.10  |

Table III: Object Feature Learning Ablation on the nuScenes [9] validation set. We compare tracking performance when using a FFN, a GNN, and a transformer for feature learning. We include Association Accuracy (AssA) [48] to validate association improvements.

| Exp | Model | AMOTA↑ | AMOTP↓ | AssA ↑ |
|-----|-------|--------|--------|--------|
| 1   | FFN   | 63.4   | 0.749  | 48.2   |
| 2   | GNN   | 66.3   | 0.688  | 51.1   |
| 3   | Trans | 72.1   | 0.566  | 56.2   |

Table IV: 3D Tracking Ablation on the nuScenes [9] validation set. Rej indicates track overlap rejection. A† indicates the 1st stage affinity estimation method, where Heuri indicates the Eager-MOT [3] heuristic and Learn indicates our learned approach. Pred indicates the prediction model, where KF indicates the Kalman filter and Vel indicates the CenterPoint [1] velocity-based motion model.

| Exp | Rej | A† | Pred | AMOTA↑ | AMOTP↓ |
|-----|-----|----|------|--------|--------|
| 1   |     |    | Heuri | 63.8   | 0.635  |
| 2   | ✓   |    | Heuri | 65.8   | 0.628  |
| 3   | ✓   |    | Learn | 71.8   | 0.573  |
| 4   | ✓   |    | Learn | 72.1   | 0.566  |

We also directly compare the discrimination ability of the feature learning models in Figure 3. We compute feature similarity for all object pairs, and plot separate distributions for objects pairs with same and different identities. We assign ground truth object identities to objects using the 3D IoU matching strategy described in Section III-C. Feature similarity is evaluated using the cosine similarity between track and detection feature vectors, \( \overline{T}_t \) and \( \overline{D}_t \), produced by the Interaction Transformer (see Section III-A). Affinity is directly taken from the affinity matrices, \( A_t \), produced by the Affinity Head (see Section III-A).

We measure discrimination as the amount of separation between same and different object distributions, as increased separation reduces ambiguity when matching object pairs. We measure separation through the difference of distribution means and compute the Jensen–Shannon divergence (JSD).

As in Table III, we observe improvements with increased interaction modeling in Figure 3. Sparse interaction modeling with the GNN shows an improvement over independent feature extraction with the FFN, with increased mean difference (+0.24) and JSD difference (+0.22). Adding complete interaction modeling with the Transformer leads to an increased mean difference (+0.21) and JSD difference (+0.21), validating the use of the Transformer for discriminative feature estimation. Based on Table III and Figure 3, we note that discrimination increases lead to increases in AMOTA, indicating the importance of object discrimination for tracking performance.

Track Rejection. Exp. 1 in Table IV shows the tracking performance of our baseline. We add a track rejection based on overlap (see Section III-B) in Exp. 2, leading to tracking improvements (+2.00%, -0.007m) on AMOTA and AMOTP. We attribute the performance increase to the removal of duplicated trajectories estimated by our tracker.

Affinity Matrix Estimation. Exp. 3 in Table IV replace the heuristic affinity matrix estimation method used in EagerMOT [3] with the learned affinity \( A_t \) estimation method outlined in Section III-A. We note improvements (+6.00%, -0.055m), which we attribute to the increased discrimination of our feature interaction mechanism.

We provide ablation studies on our network to validate our design choices. The results are shown in Tables III and IV.

Object Feature Learning. Table III shows the tracking performance when different feature learning models are used for affinity estimation. Specifically, we swap out the Interaction Transformer (Section III-A) with a feed-forward network (FFN) and a graph neural network (GNN).

Exp. 1 uses a FFN to learn object features independently, applying 8 Linear-BatchNorm-ReLU blocks. Exp. 2 uses a GNN to add sparse interaction modeling. We follow the model architecture of GNN3DMOT [7], and only construct edges between objects that are (1) in different frames and (2) within 5 meters in 3D space. Adding the GNN leads to an improvement (+2.90%, -0.061m) between same and different object distributions, as increased separation reduces ambiguity when matching object pairs. We measure separation through the difference of distribution means and compute the Jensen–Shannon divergence (JSD).

As in Table III, we observe improvements with increased interaction modeling in Figure 3. Sparse interaction modeling with the GNN shows an improvement over independent feature extraction with the FFN, with increased mean difference (+0.24) and JSD difference (+0.22). Adding complete interaction modeling with the Transformer leads to an increased mean difference (+0.21) and JSD difference (+0.21), validating the use of the Transformer for discriminative feature estimation. Based on Table III and Figure 3, we note that discrimination increases lead to increases in AMOTA, indicating the importance of object discrimination for tracking performance.
**Track Prediction/Update.** Exp. 4 in Table IV replace the Kalman filter with the estimated velocity prediction and assignment update strategy outlined in Section III-B. We observe a performance improvement (+0.30%, -0.007m) over the Kalman filtering approach, which we credit to the low accuracy of velocity propagation when filtering, especially when operating at the low 2 Hz update rate of the nuScenes dataset. When ingesting detections without velocity estimates or when operating at a higher update rate, InterTrack may benefit from a Kalman filtering approach.

**V. CONCLUSION**

We have presented InterTrack, a novel 3D multi-object tracking method that extracts discriminative track and detection features for data association via end-to-end learning. The Transformer is used as a feature interaction mechanism, shown to be effective for increasing 3D MOT performance due to its integration of complete interaction modeling. We outline updates to the EagerMOT [3] tracking pipeline leading to improved performance when paired with the Interaction Transformer. Our contributions lead to a 1\textsuperscript{st} place ranking on the nuScenes 3D MOT benchmark [9] among methods using CenterPoint [1] detections.

**REFERENCES**

[1] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3d object detection and tracking,” CVPR, 2021.

[2] X. Weng and K. Kitani, “A baseline for 3d multi-object tracking,” IROS, 2020.

[3] A. Kim, A. Osep, and L. Leal-Taixé, “Eagermot: 3d multi-object tracking via sensor fusion,” ICRA, 2021.

[4] H. Chiu, A. Priolitti, J. Li, and J. Bohg, “Probabilistic 3d multi-object tracking for autonomous driving,” arXiv preprint, 2020.

[5] X. Weng, Y. Yuan, and K. Kitani, “PTP: Parallelized Tracking and Prediction with Graph Neural Networks and Diversity Sampling,” ICRA RA-L, 2021.

[6] H. Chiu, J. Li, R. Ambrus, and J. Bohg, “Probabilistic 3d multi-modal, multi-object tracking for autonomous driving,” ICRA, 2021.

[7] X. Weng, Y. Wang, Y. Man, and K. Kitani, “GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with 2D-3D Multi-Feature Learning,” CVPR, 2020.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” NeurIPS, 2017.

[9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” CVPR, 2020.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” NAACL, 2019.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” arXiv, 2019.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” NeurIPS, 2020.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly \textit{et al.}, “An image is worth 16x16 words: Transformers for image recognition at scale,” ICLR, 2021.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” ECCV, 2020.

[15] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “Loft: Detector-free local feature matching with transformers,” CVPR, 2021.
[16] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, “End-to-end contextual perception and prediction with interaction transformer,” *IJROS*, 2020.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[18] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” *ECCV*, 2010.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” *ICCV*, 2011.

[20] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, “Universal correspondence network,” *NeurIPS*, 2016.

[21] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” *ICCV*, 2017.

[22] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperGlue: Learning feature matching with graph neural networks,” *CVPR*, 2020.

[23] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaberi, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[24] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” *ICIP*, 2016.

[25] P. Chu and H. Ling, “Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking,” *ICCV*, 2019.

[26] C. Luo, X. Yang, and A. L. Yuille, “Exploring simple 3d multi-object tracking for autonomous driving,” *ICCV*, 2021.

[27] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” *arXiv preprint*, 2021.

[28] Z. Lu, V. Rathod, R. Votel, and J. Huang, “Retinatrack: Online single stage joint detection and tracking,” *CVPR*, 2020.

[29] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, “Transtrack: Multiple-object tracking with transformer,” *arXiv preprint*, 2020.

[30] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei, “Motr: End-to-end multiple-object tracking with transformer,” *arXiv preprint*, 2021.

[31] Y. Zeng, C. Ma, M. Zhu, Z. Fan, and X. Yang, “Cross-modal 3d object detection and tracking for auto-driving,” *IROS*, 2021.

[32] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” *CVPR*, 2008.

[33] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, “Mono-camera 3d multi-object tracking using deep learning detections and PMBM filtering,” *IV*, 2018.

[34] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, “Globally-optimal greedy algorithms for tracking a variable number of objects,” *CVPR*, 2011.

[35] H. Hu, Q. Cai, D. Wang, J. Lin, M. Sun, P. Krähenbühl, T. Darrell, and F. Yu, “Joint monocular 3d vehicle detection and tracking,” *ICCV*, 2019.

[36] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, 1955.

[37] P. C. Mahalanobis, “On the generalized distance in statistics,” *Proceedings of the National Institute of Sciences (Calcutta)*, 1936.

[38] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, “TransFusion: Robust Lidar-Camera Fusion for 3d Object Detection with Transformers,” *CVPR*, 2022.

[39] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, “Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation,” *arXiv*, 2022.

[40] Y. Yan, Y. Mao, and B. Li, “SECOND: Sparsely embedded convolutional detection,” *Sensors*, 2018.

[41] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *PAMI*, 2018.

[42] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” *CVPR*, 2012.

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*. Curran Associates, Inc., 2019. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[44] O. D. Team, “Openpcdet: An open-source toolbox for 3d object detection from point clouds,” https://github.com/open-mmlab/OpenPCDet, 2020.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *ICLR*, 2015.

[46] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay,” *arXiv preprint*, 2018.

[47] S. Shi, X. Wang, and H. Li, “PointRCNN: 3D object proposal generation and detection from point cloud,” *CVPR*, 2019.

[48] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, “Hota: A higher order metric for evaluating multi-object tracking,” *IJCV*, 2021.

[49] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, “Class-balanced grouping and sampling for point cloud 3d object detection,” *arXiv preprint*, 2019.