# End-to-End 3D Multi-Object Tracking and Trajectory Forecasting

Xinshuo Weng*, Ye Yuan*, and Kris Kitani

Robotics Institute, Carnegie Mellon University
{xinshuow, yyuan2, kkitani}@cs.cmu.edu

**Abstract.** 3D multi-object tracking (MOT) and trajectory forecasting are two critical components in modern 3D perception systems. We hypothesize that it is beneficial to unify both tasks under one framework to learn a shared feature representation of agent interaction. To evaluate this hypothesis, we propose a unified solution for 3D MOT and trajectory forecasting which also incorporates two additional novel computational units. First, we employ a feature interaction technique by introducing Graph Neural Networks (GNNs) to capture the way in which multiple agents interact with one another. The GNN is able to model complex hierarchical interactions, improve the discriminative feature learning for MOT association, and provide socially-aware context for trajectory forecasting. Second, we use a diversity sampling function to improve the quality and diversity of our forecasted trajectories. The learned sampling function is trained to efficiently extract a variety of outcomes from a generative trajectory distribution and helps avoid the problem of generating many duplicate trajectory samples. We show that our method achieves state-of-the-art performance on the KITTI dataset. Our project website is at http://www.xinshuoweng.com/projects/GNNTrkForecast.

**Keywords:** multi-object tracking, trajectory forecasting

## 1 Introduction

3D multi-object tracking (MOT) and trajectory forecasting are critical components in modern perception systems. Historically, MOT [8,7,9] and trajectory forecasting [6,2,4] have been studied separately. As a result, modern perception systems often perform 3D MOT and forecasting separately in a cascaded order, where tracking is performed first to obtain trajectories in the past, followed by trajectory forecasting to predict trajectories in the future. However, this cascaded pipeline with separately trained modules can lead to sub-optimal performance, as information is not shared during training. Since tracking and forecasting modules are highly dependent, it would be beneficial to optimize them jointly. For example, a better MOT module can lead to better performance of its downstream forecasting module while a more accurate motion model learned in trajectory forecasting can improve data association for MOT.

## 2 A Joint 3D MOT and Trajectory Forecasting Model

Our goal is to jointly optimize the MOT and forecasting modules and enable the propagation of performance information through the entire system during training. Instead of running two modules separately in a sequential order as shown in

* First two authors contributed equally to this work.

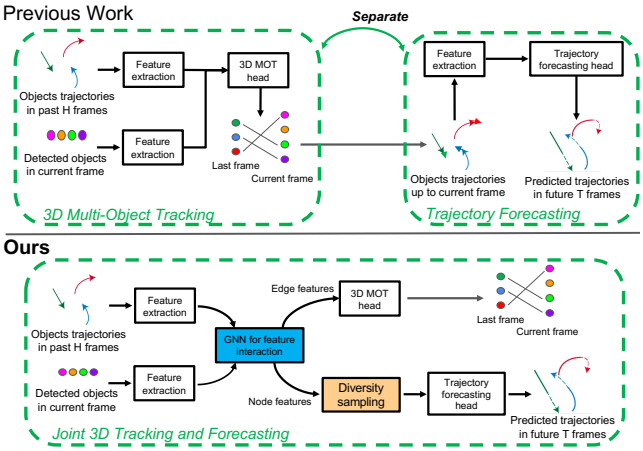


**Fig. 1: (Top)** Previous work: 3D MOT and trajectory forecasting treated separately and connected as a sequential process. **(Bottom)** Our proposed model: Joint process for tracking and forecasting. Two key innovations: (1) a feature interaction using GNNs (blue box) to improve tracking association and trajectory forecasting in the presence of multiple agents; (2) a diversity sampling (orange box) to improve sample efficiency and produce diverse and accurate trajectory samples.

Fig. 1 (top), we propose to perform MOT and forecasting in parallel as shown in Fig. 1 (bottom). As a result, the gradients computed in both heads (one for each task) can be propagated back to learn a better shared feature representation for both tasks. By keeping the MOT and forecasting heads parallel, i.e., forecasting does not explicitly depend on the MOT results, we can prevent association errors made in MOT from directly influencing the forecasting module. The forecasting module can still use the implicit MOT information encoded in the shared features computed by the GNN.

## 3 Social Interaction Modeling with Graph Networks

Modeling interaction for 3D MOT is crucial in the presence of multiple agents but has often been overlooked in prior work. Prior work in 3D MOT extracts the feature of each object *independently*, i.e., each object's feature only depends on the object's own inputs (image crop or location). As a result, there is no interaction between objects. We found that *independent feature extraction leads to inferior discriminative feature learning*, and object dependency is the key to obtaining discriminative features. Intuitively, the features of the same object over two frames should be as similar as possible and the features between two different objects should be as different as possible to avoid confusion during data association. This can only be achieved if object features can be obtained as a dependent or context-aware process, i.e., modeling interactions between objects.

To model interaction in 3D MOT, we employ a feature interaction mechanism as shown in Fig. 1 (Bottom) by introducing Graph Neural Networks (GNNs) to 3D MOT. Specifically, we construct a graph with each node being an object in the scene. Then, at every layer of the GNNs, each node can update its feature by aggregating features from other nodes. This node feature aggregation is useful

**Table 1:** 3D MOT evaluation on the KITTI dataset.

| Methods | **sAMOTA**(%)↑ | AMOTA(%)↑ | AMOTP(%)↑ | MOTA(%)↑ | MOTP(%)↑ | IDS↓ | FRAG↓ |
|---|---|---|---|---|---|---|---|
| FANTrack [1] | 82.97 | 40.03 | 75.01 | 74.30 | 75.24 | 35 | 202 |
| AB3DMOT[8] | 93.28 | 45.43 | **77.41** | 86.24 | **78.43** | **0** | 15 |
| **Ours** | **94.41** | **46.15** | 76.83 | **86.89** | 78.32 | 3 | 8 |

because the resulting object features are no longer isolated and are adapted according to other objects. We observe in our experiments that, after a few GNN layers, the affinity matrix becomes more discriminative than the affinity matrix obtained without interaction. In addition to using GNNs to model interaction for 3D MOT, GNNs can also provide socially-aware context to improve trajectory forecasting [3]. To the best of our knowledge, we are the first to employ GNNs in a unified 3D MOT and trajectory forecasting method.

## 4 Diversity Sampling for Trajectory Forecasting

Since future trajectories of objects should be stochastic and multi-modal due to many unobserved factors (e.g., hidden intentions), prior work in trajectory forecasting often learns the future trajectory distribution with deep generative models. At test time, these methods randomly sample a set of future trajectories from the generative model without considering the correlation between samples. As a result, the samples can be very similar and only cover a limited number of modes, leading to poor sample efficiency. This inefficient sampling strategy is harmful in real-time applications because producing a large number of samples can be computationally expensive and lead to high latency. Moreover, without covering all the modes in the trajectory distribution and considering all possible futures, the perception system cannot plan safely, which is important in safety-critical applications such as autonomous driving.

To improve sample efficiency in trajectory forecasting, we depart from the random sampling in prior work and employ a diversity sampling technique that can generate diverse trajectory samples from a pretrained CVAE model. The idea is to learn a separate sampling network which maps each object's feature to a set of latent codes. The latent codes are then decoded into trajectory samples. In this way, the produced samples are correlated (unlike random sampling where the samples are independent), which allows us to enforce structural constraints such as diversity onto the samples. Specifically, we use determinantal point processes (DPPs) to optimize the diversity of the samples.

## 5 Experiments

**Datasets**. We use standard autonomous driving datasets: KITTI [5]. Also, since there is no existing evaluation procedure that can jointly evaluate 3D MOT and trajectory forecasting, we evaluate two modules separately and compare against prior work on each individual module of our joint method. For KITTI, same as most prior works, we report results on the car subset for comparison.

**Evaluating 3D Multi-Object Tracking.** We use standard CLEAR metrics (including MOTA, MOTP, IDS) and new sAMOTA, AMOTA and AMOTP metrics [8] for evaluation. We summarize the results in Table 1. Our method consistently outperforms baselines in sAMOTA, which is the primary metrics

**Table 2:** Trajectory forecasting evaluation on the KITTI dataset.

| Settings | Metrics | Conv-Social [4] | Social-GAN [6] | TraPHic [2] | Graph-LSTM [3] | **Ours** |
|---|---|---|---|---|---|---|
| KITTI-1.0s | ADE↓ | 0.607 | 0.586 | 0.542 | 0.478 | **0.471** |
| | FDE↓ | 0.948 | 1.167 | 0.839 | 0.800 | **0.763** |
| | ASD↑ | 1.785 | 0.495 | 1.787 | 1.070 | **2.351** |
| | FSD↑ | 1.987 | 0.844 | 1.988 | 1.836 | **4.071** |
| KITTI-3.0s | ADE↓ | 2.362 | 2.340 | 2.279 | 1.994 | **1.319** |
| | FDE↓ | 3.916 | 4.102 | 3.780 | 3.351 | **2.299** |
| | ASD↑ | 2.436 | 1.351 | 2.434 | 2.745 | **5.843** |
| | FSD↑ | 2.973 | 2.066 | 2.973 | 4.582 | **10.123** |

for ranking MOT methods. We hypothesize that this is because our method leveraging GNN obtains more discriminative features to avoid confusion in MOT association while all 3D MOT baselines ignore the interaction between objects. Moreover, joint optimization of the tracking and forecasting modules might help.

**Evaluating Trajectory Forecasting.** We use the standard metrics: Average Displacement Error (ADE) and Final Displacement Error (FDE) for evaluation. Additionally, to evaluate the diversity of the trajectory samples, we use Average Self Distance (ASD) and Final Self Distance (FSD) in [10] for sample diversity evaluation. We summarize the results in Table 2. Our method, which (1) is jointly trained with a 3D MOT head, (2) uses GNNs for feature interaction and (3) uses diversity sampling, outperforms the baselines in both accuracy and diversity metrics. Particularly, our method outperforms baselines for the long-horizon (i.e., 3.0s) experiment. This is because our method has a higher sample efficiency and can cover different modes of the future trajectory distribution.

## References


1. Baser, E., Balasubramanian, V., Bhattacharyya, P., Czarnecki, K.: FANTrack: 3D Multi-Object Tracking with Feature Association Network. IV (2020)
2. Chandra, R., Bhattacharya, U., Bera, A., Manocha, D.: TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions. CVPR (2019)
3. Chandra, R., Guan, T., Panuganti, S., Mittal, T., Bhattacharya, U., Bera, A., Manocha, D.: Forecasting Trajectory and Behavior of Road-Agents Using Spectral Clustering in Graph-LSTMs. arXiv:1912.01118 (2019)
4. Deo, N., Trivedi, M.M.: Convolutional Social Pooling for Vehicle Trajectory Prediction. CVPRW (2018)
5. Geiger, A., Lenz, P., Urtasun, R.: Are We Ready for Autonomous Driving? the KITTI Vision Benchmark Suite. CVPR (2012)
6. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. CVPR (2018)
7. Wang, Y., Weng, X., Kitani, K.: Joint Detection and Multi-Object Tracking with Graph Neural Networks. arXiv:2006.13164 (2020)
8. Weng, X., Wang, J., Held, D., Kitani, K.: 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. IROS (2020)
9. Weng, X., Wang, Y., Man, Y., Kitani, K.: GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with 2D-3D Multi-Feature Learning. CVPR (2020)
10. Yuan, Y., Kitani, K.: Diverse Trajectory Forecasting with Determinantal Point Processes. ICLR (2020)