Overview of Challenges in Trajectory Forecasting and 3D Perception for Autonomous Driving

Abstract

This document provides a summary of the challenges faced in the domain of Autonomous Driving. The dataset incorporated into the study includes 150 minutes of labeled Trajectory and 3D Perception data, comprising approximately 80,000 lidar point clouds and 1000 kilometers of trajectories in urban traffic conditions. The competition is divided into two main segments: (1) Forecasting Trajectories and (2) 3D Lidar Object Recognition. Over 200 teams provided their results on the leaderboard, and more than 1,000 individuals took part in the workshop.

1 Introduction

The focus of this paper is to investigate multi-frame perception, prediction, and planning as applied to autonomous driving. It serves as a platform to bring together academic and industry experts to discuss the uses of computer vision in the context of self-driving vehicles.

2 Dataset

The Apolloscape Dataset is utilized as a research tool designed to advance autonomous driving in various dimensions, including perception, navigation, prediction, and simulation. This dataset is comprised of labeled street view images and simulation resources that can accommodate user-defined strategies. The dataset includes tasks such as Trajectory Prediction, 3D Lidar Object Detection, 3D Lidar Object Tracking, lane marking segmentation, online self-positioning, 3D car instance comprehension, Stereo, and Inpainting Dataset. A dedicated online assessment platform and user toolkit are provided for each task.

For data collection related to Trajectory Prediction and 3D Perception, a data-gathering vehicle was utilized to amass traffic information, including camera-captured images and LiDAR-generated point clouds. Our vehicle operates in urban settings during peak traffic times. The dataset features camera imagery, 3D point cloud data, and paths of traffic agents within the LiDAR's operational area. This newly created dataset, which includes 150 minutes of sequential information, is extensive and concentrates on urban roadways, with a particular emphasis on 3D perception, prediction, planning, and simulation activities involving a variety of traffic agents.

3 Challenge

This part elaborates on the specifics of the challenges, the metrics for evaluation, and the outcomes achieved.

3.1 Trajectory Prediction Challenge

Trajectory information is documented at a rate of 2 frames per second. Each entry in the data file includes the frame identifier, object identifier, object category, object's position in the global

.

coordinate system along the x, y, and z axes, the object's dimensions in terms of length, width, and height, and the object's orientation. Measurements for position and bounding box dimensions are provided in meters. There are five distinct categories for object types: small vehicles are designated as 1, large vehicles as 2, pedestrians as 3, motorcyclists and bicyclists as 4, traffic cones as 5, and others as 6.

3.1.1 Evaluation Metric

For the assessment, the categories of small and large vehicles are merged into a single category termed 'vehicle'. The challenge requires using the initial three seconds of data from each sequence as input to forecast the trajectories of objects for the subsequent three seconds. The objects assessed are those present in the final frame of the first three seconds. Subsequently, the discrepancies between the anticipated locations and the actual locations of these objects are calculated.

The following metrics are used to evaluate the effectiveness of the algorithms:

1. Average Displacement Error (ADE): This metric represents the average Euclidean distance between all predicted positions and their corresponding actual positions throughout the forecasting period.

2. Final Displacement Error (FDE): This metric calculates the average Euclidean distance between the ultimately predicted positions and the actual final positions. Given the varying scales of trajectories for vehicles, pedestrians, and bicyclists, a weighted sum of ADE (WSADE) and a weighted sum of FDE (WSFDE) are employed as metrics.

$WSADE = Dv \cdot ADEv + Dp \cdot ADEp + Db \cdot ADEb$ (1)

$WSFDE = Dv \cdot FDEv + Dp \cdot FDEp + Db \cdot FDEb$ (2)

Here, Dv, Dp, and Db are associated with the inverse of the average speeds of vehicles, pedestrians, and bicyclists in the dataset, with values set at 0.20, 0.58, and 0.22, respectively.

3.2 3D Detection Challenge

The dataset for 3D Lidar object detection features LiDAR-scanned point clouds accompanied by detailed annotations. It was gathered in Beijing, China, under diverse conditions of lighting and traffic density. Specifically, the dataset encompasses intricate traffic patterns that include a mix of vehicles, cyclists, and pedestrians.

3.2.1 Data Structure

Each annotated file for 3D Lidar object detection represents a one-minute sequence captured at two frames per second. An entry within each file includes the frame number, object ID, object classification, positions along the x, y, and z axes, object dimensions (length, width, height), and orientation. Object classifications are consistent with those in the trajectory data. In this evaluation, the first two categories—small and large vehicles—are considered as a single 'vehicle' class. Positional data is relative, with units in meters, and the heading angle denotes the object's steering direction.

3.2.2 Evaluation Metric

The evaluation metric is analogous to the one defined in prior work. The aim of the 3D object detection task is to develop detectors for 'vehicle', 'pedestrian', and 'bicyclist' categories. These detectors should estimate the 3D bounding box (dimensions and position) and provide a detection score or confidence. It is important to note that not all objects within the point clouds are labeled. The performance of 3D object detection is assessed using the mean Average Precision (mAP), based on Intersection over Union (IoU). The evaluation standard aligns with the 2D object detection benchmark, utilizing 3D bounding box overlap. The ultimate metric is the average mAP across vehicles, pedestrians, and bicyclists, with IoU thresholds set at 0.7 for cars, and 0.5 for both pedestrians and cyclists.

4 Methods and Teams

4.1 Trajectory prediction

One team utilized an encoder-decoder framework based on LSTM for predicting trajectories on city streets. To enhance prediction accuracy, they implemented four sequence-to-sequence sub-models to capture the distinct movement characteristics of various traffic participants. They produced a future trajectory for each agent through a three-step process: encoding, perturbation, and decoding. Initially, an encoder was employed to embed the past trajectory. Subsequently, they introduced a 16-dimensional random noise to the encoder's output to accommodate the multimodal distribution of the data. Finally, they generated the predicted trajectory via a decoder that mirrored the encoder's structure.

In addition, they attempted to capture the collective influence among road agents using an interaction technique. Improving upon the original methodology, they conducted an interaction operation at each moment during the encoding and decoding phases. The interaction module embedded the positions of all agents and generated a comprehensive 128-dimensional spatiotemporal representation using an LSTM unit. The derived feature was then relayed to the encoders or decoders for the primary prediction task. Each encoder or decoder, linked to a particular individual, produced the private interaction within a confined area through an attention operation, utilizing the aforementioned global feature and the agent's position. Their experimental findings indicated that the interaction module enhanced prediction accuracy on the dataset.

4.2 3D Detection

One team introduced an innovative approach termed sparse-to-dense 3D object detector (STD). STD is characterized as a two-stage, point-based detection system. The initial phase involves a bottom-up network for generating proposals, where spherical anchors are seeded on each point to encompass objects at various orientations. This spherical anchor design reduces computational load and shortens inference time by eliminating the need to account for differently oriented objects during anchor creation. Subsequently, points within these spherical anchors are collected to form proposals for additional refinement. In the second phase, a PointsPool layer is introduced to transform the features of proposals from point-based representations to compact grid formats. These dense features are then processed through a prediction head, which includes two extra fully-connected layers, to derive the final detection outcomes. A 3D intersection-over-union (IoU) branch is also incorporated into the prediction head to estimate the 3D IoU between the final predictions and the ground-truth bounding boxes, thereby enhancing localization precision.

During the training process, four distinct data augmentation techniques were employed to mitigate overfitting. Initially, similar to previous methods, ground-truth bounding boxes with their corresponding interior points were randomly added from different scenes to the existing point cloud, simulating objects in varied settings. Subsequently, each bounding box was randomly rotated based on a uniform distribution and subjected to random translation. Additionally, every point cloud was randomly flipped along the x-axis with a 50% probability. Lastly, random rotation and scaling were applied to each point cloud using uniformly distributed random variables. In the testing phase, predictions were first obtained on both the original and the x-axis flipped point clouds, and these results were then merged using Soft-NMS to produce the final predictions.

Another team's strategy is based on the PointPillars framework. The network configuration largely mirrors that of the original work, with adjustments made to accommodate multiple anchors for each class. The substantial variation in the size of objects within each class suggested that a single anchor might be inadequate. The k-means algorithm was utilized to create five anchors for each class. Another modification involved deactivating the direction classification in the loss function, as the evaluation metric relies on IOU, which is not affected by direction. Detailed settings for each class are presented in Table 1.

To enhance training data, global translation and scaling of the point cloud, along with rotation and translation for each ground truth, were implemented. Global rotation of the point cloud was omitted as it was found to produce less favorable outcomes. The specific parameters for these adjustments are detailed in Table 2.

Table 1: Detailed settings for each class. MNP indicates the maximum number of points, and MNV represents the maximum number of voxels.

Class	Number of anchors	Voxel size	MNP	MNV
Car	5	[0.28,0.28,32]	50	20000
Bicyclist	5	[0.14,0.14,32]	20	80000
Pedestrian	5	[0.10,0.10,32]	15	80000

Table 2: Augmentation parameters for training data.

Global Rotation	Global Translation	Global Scaling	Ground Truth Rotation	Ground Truth Translation
	[0.2,0.2,0.2]	[0.95,1.1]	[-/20, /20]	[0.25,0.25,0.25]

Test Time Augmentation was employed to enhance performance. For every point cloud, four iterations were generated: the original, and versions flipped along the x-axis, y-axis, and both axes. Each iteration was processed by the network to obtain bounding box predictions, which were subsequently unflipped. Due to the flipping operation, anchors across iterations have a one-to-one correspondence. For each anchor, the corresponding predicted boxes were combined by averaging the location, size, and class probability. Redundant boxes were then eliminated using Non-Maximum Suppression (NMS).

Another Team introduced enhancements to the PointPillars method. Their approach incorporated residual learning and channel attention mechanisms into the baseline architecture. The network is composed of the original Pillar Feature Network, an extended 2D CNN backbone, and a detection head for foreground/background classification and regression. The deeper backbone significantly improves detection accuracy compared to the original PointPillars. A separate network was trained for each class in the Apollo training dataset to perform binary classification, resulting in four distinct networks. Final predictions were compiled by aggregating all foreground predictions from these networks.

For dataset preprocessing, methods from the KITTI dataset were adapted, including positive example sampling, global rotation, individual object rotation, and random scaling for each object. However, unlike the KITTI approach, global rotation was excluded, and the ranges for scaling and rotation were reduced. Additionally, more foreground point clouds were sampled to augment positive examples. Table 3 details the specific settings for each class.

Table 3: Detailed settings for each class. MSN indicates the maximum sampling number.

Class	Pointcloud Range (m)	Pillar Size (m)	Anchor Size (m)	MSN
Vehicles	x: -70.8 to 70, y: -67.2 to 67.2, z: -3 to 1	x: 0.16, y: 0.16, z: 3	x: 1.6, y: 3.9, z: 1.56	15
Pedestrian	x: -70.8 to 70, y: -67.2 to 67.2, z: -2.5 to 0.5	x: 0.2, y: 0.2, z: 3	x: 0.6, y: 1.76, z: 1.73	15
Motor&bicyclist	x: -70.8 to 70, y: -67.2 to 67.2, z: -2.5 to 0.5	x: 0.2, y: 0.2, z: 3	x: 0.6, y: 0.8, z: 1.73	15

5 Conclusion and Future Work

This paper provides a review of the challenges encountered in the domain of Autonomous Driving, with a focus on the analysis of 3D Detection and Trajectory prediction. It is anticipated that this paper will offer contemporary insights into these research areas.

Future endeavors will aim to refine the open-source tools and dataset for autonomous driving. Moreover, additional workshops and challenges are planned to foster the exchange of concepts and to collectively propel the field of autonomous driving research forward.