

DBNet: A Large-Scale Dataset for Driving Behavior Learning

Yiping Chen^{*1}, Jingkang Wang^{*2}, Jonathan Li^{#1,3}, Cewu Lu^{#2},
Zhipeng Luo¹, Han Xue², and Cheng Wang¹

¹Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University

²Shanghai Jiao Tong University ³University of Waterloo

<http://www.dbehavior.net>

Abstract

Learning autonomous-driving policies is one of the most challenging but promising tasks for computer vision. Most researchers believe that future research and applications should combine cameras, video recorders and laser scanners to obtain comprehensive semantic understanding of real traffic. However, current approaches only learn from large-scale videos, due to the lack of benchmarks that consist of precise laser-scanner data. In this paper, we are the first to propose a Driving Behavior Net (DBNet), which provides large-scale high-quality point clouds scanned by a Velodyne laser, videos recorded by a dashboard camera and standard drivers' behaviors. Extensive experiments demonstrate that extra depth information help networks to determine driving policies indeed.

1. Introduction

Driving policy learning is a core problem in autonomous driving research. Computer vision is expected to play an important role in this challenging task, since driving planning and perception together run as a closed loop. Therefore, some computer vision researchers [14, 20, 21, 27] attempt to model it as a perception-action model, which is an end-to-end system that maps from pixels to actuation. It opens a new direction in the autonomous driving field.

However, current research and dataset neglect an important cue, namely, depth information. On the one hand, biological experiments [4, 19] show that monocular people can not drive nicely. For instance, monocular drivers in experiments did worse in parking and lane changing tasks for

* indicates equal contributions.

the corresponding authors are Jonathan Li and Cewu Lu. E-mails: junli@uwaterloo.ca and lucewu@sjtu.edu.cn. Cewu Lu is also member of AI research institute at SJTU.

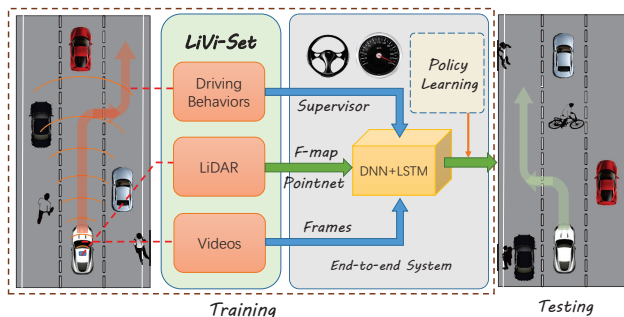


Figure 1. Driving Behavior Net (DBNet): a benchmark for policy learning in autonomous driving.

the lack of stereoscopic depth perception. It verifies depth information is necessary, though drivers have perfect recognition ability. Most of the people believe that depth information should be a necessary cue in real-world auto-driving due to the consideration of safety. On the other hand, many high-quality depth sensors would be cheap and widely affordable. For example, the cost of Velodyne comes to hundreds of dollars, that is, it will be ready to be equipped in most autonomous cars.

In consequence, computer vision researchers should pay more attention to perception-action model with depth. Whereas, we found it still misses out both research roadmaps and datasets. Thus, this paper aims to fundamentally study this problem. We offer a large-scale dataset that includes both driving videos with depth and corresponding driving behaviors. Our dataset is largely different from previous ones for vision-based auto-driving research. On the one hand, the depth data sampled by a LiDAR camera is provided, which misses in [8, 27]. On the other hand, some datasets like *KITTI* [10, 11] provide depth information, however, driving behavior is not included, which makes them fail to be a benchmark for policy learning. In short, the proposed dataset is the first driving policy learning dataset

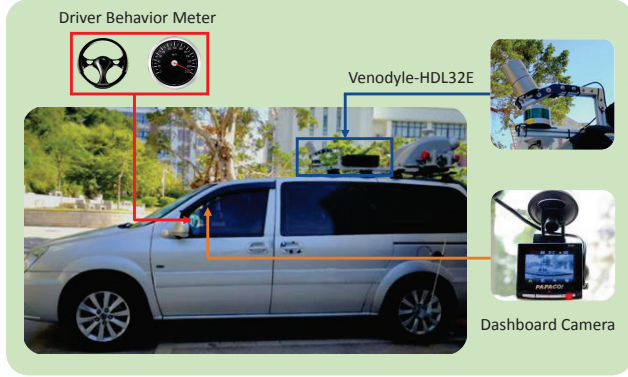


Figure 2. Our data collection platform with multiple sensors.

that includes depth information. Our dataset involves many features: (1) **large-scale**: our dataset consists of more than 10k frames of real street scenes and the amount of data exceeds 1TB in total. (2) **diversity**: we record continuous but varied scenes in real traffic, such as seaside roads, school areas and even mountain roads, which include a number of crossroads, pedestrians and traffic signs. (3) **high-quality**: point clouds, videos and drivers’ behaviors in our dataset are all acquired by high-resolution sensors, which provides distinct recovery of real driving conditions.

Apart from the dataset, this paper attempts to thoroughly study how important depth information is for auto-driving and fully discuss what we can achieve if current techniques are used. First, we produce an analysis that why depth is necessary for autonomous vehicles. Second, we answer the question that how to leverage current techniques, if depth is given. Finally, we draw a conclusion that depth information would benefit learning driving policies and it has a large room to improve techniques in terms of how to use depth. It again verifies that a qualified dataset is crucial for advancing this topic.

In conclusion, the key contributions of our work in this paper are mainly two aspects: First, we propose a dataset which is the first policy learning benchmark composed of driving videos, LiDAR data and corresponding driving behaviors. Second, we conduct complete analysis on how important depth information is, how to leverage depth information and what we can achieve by utilizing current techniques.

2. Related Work

The ultimate goal in autonomous vehicle navigation is to learn driving policies. In this section, we investigate driving policy learning methods and existing driving datasets.

2.1. Driving Policy Learning

Because of the complexity of real street scenes, deep learning techniques such as neural network are expected the

most promising methods to solve this problem. Pomerleau *et al.* [21] was the pioneer to use neural networks for lane following and obstacles avoiding. There are now two main-stream ways for this promising task.

End-to-end learning: This line of works employed end-to-end systems mapping pixels directly to policies. [14, 20, 21] demonstrated that autonomous vehicles are capable to learn driving policies nicely in simple scenarios, such as highway. NVIDIA [14] group did excellent attempts to map directly from images by utilizing multi-layer convolution neural network and successfully self-drive in real roads. Recently, [27] broadened video scenes and illustrated that it is feasible for vehicles to drive in multiple complex situations.

Learning affordable rules: Rather than directly obtain driving policies, these works learned some affordable information in advance which is helpful for decision making. [6] proposed to learn some pre-defined low-level measures such as depth information. Whereas, more works [1, 3, 29] used neural networks to solve relevant helpful problems such as semantic segmentation based on monocular images. [2, 5, 7, 18] attempt to perform 3D object detection or segmentation leveraging LiDAR information.

2.2. Existing Driving Datasets

Large-scale datasets have contributed greatly to the development of machine learning and computer vision. As for the autonomous driving area, research relies much on some benchmarks [8, 11, 17, 24, 27]. These datasets have different features and hierarchies. We conducted a comprehensive survey on existing driving datasets in the view of policy learning challenge.

Comma.ai [24] proposed their novel architecture for policy learning with their dataset published, which contains around 7.25-hour highway driving data divided into 11 videos. The released video frames are 160×320 pixels in the middle of the captured screen. Besides, the vehicle is equipped with several sensors that were measured with different frequencies and interpolated to 100Hz. Example data coming from sensors are the car speed, steering angle, GPS, gyroscope, IMU, etc. However, this dataset only concentrates on highway driving scenarios, which is not suitable for generic driving policy learning. In addition, it only consists of 2D vision information, that is, only images are used for making decisions.

KITTI [10, 11] established a benchmark which comprises 389 stereo and optical flow image pairs, stereo visual odometry sequences of 39.2 km length, and over 200k 3D object annotations captured in cluttered scenarios. It provides instance-level annotations for humans and vehicles in real scenes, which is intended for object detection and segmentation tasks. However, KITTI is only composed of less

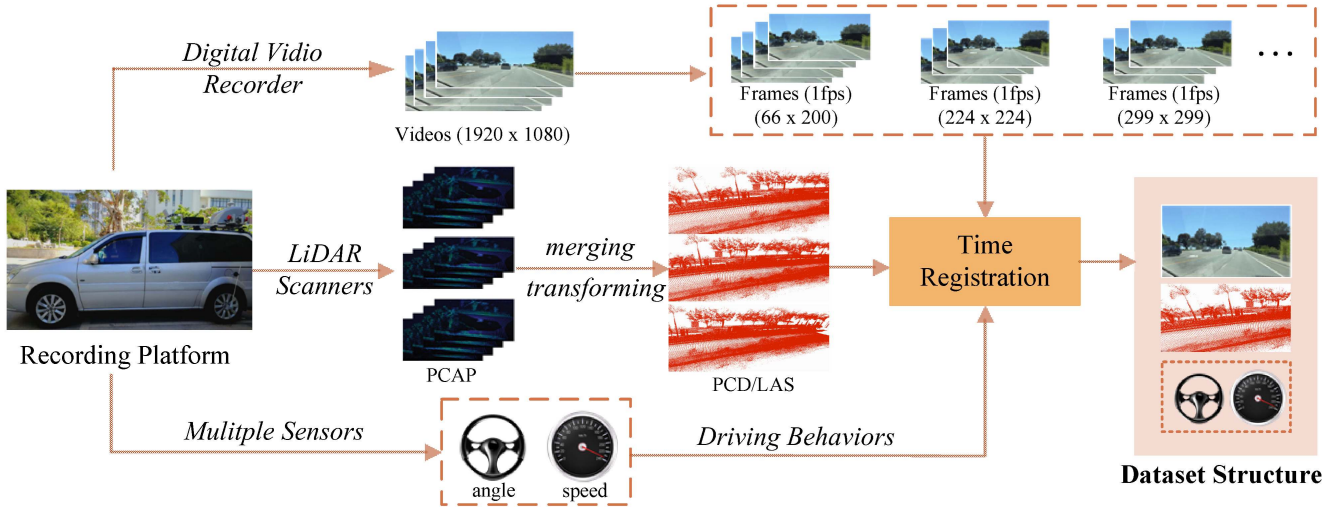


Figure 3. **The pipeline of data preprocessing when constructing dataset.** Multiple perception are equipped for acquiring high-resolution data. Videos, point clouds and driving behaviors are preprocessed jointly in figure. Finally, we register the corresponding time for three types of data and obtain our benchmark.

busy suburban traffic scenes. In other words, KITTI exhibits significantly fewer flat ground structures, fewer humans, and more suburb scenes, which results in lack of diversity. Moreover, the vehicles do not fix multiple sensors, so there is no standard drivers’ behaviors. On the whole, this benchmark is not designed for learning driving policies, but for other affordable tasks.

Cityscapes Cityscapes [8] is a large-scale, diverse set of stereo video sequences recorded in streets from 50 different cities. It mainly provides images and a small number of videos. In particular, 5000 of these images have high-quality pixel-level annotations and 20000 additional images have coarse annotations to enable methods that leverage large volumes of weakly-labeled data. The data set is designed to capture the high variability of outdoor street scenes and was acquired from a moving vehicle during several months, covering spring, summer, and fall in 50 cities, primarily in Germany but also in neighboring countries. Although this benchmark did well in the diversity of scenarios, the shortage of 3D perception such as LIDAR and driving status data makes it not so appropriate to learn driving policies.

Oxford The data was collected by the Oxford RobotCar platform [24], an autonomous Nissan LEAF. It includes over 1000km of recorded driving with almost 20 million images collected from 6 mounted cameras, along with LIDAR, GPS and INS ground truth. In addition, it was collected in all weather conditions, including heavy rain, night, direct sunlight and snow. Road and building works over the period of a year significantly changed sections of the route from the beginning to the end of data collection. Same as KITTI and Cityscapes, it omits drivers’ behaviors, which is

of great significance for the decision prediction.

BDDV Berkley DeepDrive Video dataset [27] (unpublished completely) is a benchmark that is intended for driving predictions, which provides more than 10k-hour dash-cam videos in different periods of multiple cities with varied weather conditions. From the paper, it is at least two orders larger than other public datasets for vision-based autonomous driving. It also contains labels including steering angles and vehicle speeds like Comma.ai. Due to focus on end-to-end generic driving model training, it neglects specific car annotations. Unfortunately, it only concerns 2D vision. In another word, it misses 3D stereoscopic depth perception information such as point clouds or meshes, which is an essential cue for future vehicle driving.

3. Dataset

Our dataset is intended for driving policy learning and largely different from previous ones for its novel hierarchy and excellent properties. In this section, we firstly introduce our collection platform system in Section 3.1. Then the pipeline of preprocessing LiDAR data is given in Section 3.2. Finally, in Section 3.3, we compare our dataset with existing benchmark and display features of *DBNet*.

3.1. Platform and Data Collection

As is shown in Figure 2, the dataset was acquired by our collection system in a multi-functional road information acquisition vehicle. The vehicle we used is a Buick GL8 loaded with multiple perception scanners and sensors. We collected three types of signals, namely, point clouds, videos and driving behaviors.

Datasets	Video/Image	LiDAR	Behaviors
KITTI	✓	✓	×
Cityscape	✓	×	×
Oxford	✓	✓	×
Comma.ai	✓	×	✓
BDDV	✓	×	✓
DBNet (ours)	✓	✓	✓

Table 1. **Comparison with existing public driving datasets.** Our dataset is first to combine 2D and 3D vision with labeled drivers’ behaviors.

Point Cloud We equipped the vehicle with a pair of Velodyne scanners, including one HDL-32E and one VLP-16 laser scanner. In our experiment, we mainly exploited HDL-32E to collect point cloud data. The HDL-32E is always used for high-precision and wide-range point clouds acquisition, of which frequency is 10 Hz with 32 laser beams and depth range is from 1m to 70m with a resolution of 2cm. The range of scanning angle is from +10.67 to −30.67 degrees view in vertical and 360 degrees view in horizontal. The density is about 700,000 points per second. Velodyne laser scanners are installed on the top front of our vehicle.

Video A color dashboard camera with real-time update system is placed on the top right of the front glass, which captures the video frame with 30 frames per second, of which the resolution is up to 1920×1080 . Moreover, 128G memory space makes it possible to record 20-hour continuous 1080P videos maximally.

Driving Behavior A recording software is wirelessly connected to vehicle controller to get velocity from sensors equipped. Its resolution is up to $0.1km/h$. The driver steering angle meter acquires the orientation data, whose resolution is 1 degree. When the steering wheel has a left (right) rotation with regard to standard center angle, angle meter records a negative (positive) value.

Using our platform, we totally obtained seven sets with different test scenarios. Every set contains three types of data including point clouds, videos and driver behaviors. The amount of all point clouds is around 1TB and traffic videos are about 15GB. In addition, collected data includes a variety of traffic conditions such as boulevard, primary road, mountain road, school area, narrow road and even tourist special route.

3.2. Data Preprocessing

In this section, we only introduce the preprocessing of point clouds. The processing of videos and driving polices is given in Section 4.4.

On the whole, there are three major aspects of point cloud processing (middle pipeline in Figure 3).

Frames Fusion Every 200 frames of raw point clouds, one of which captures small part of real scenes, were fused into one scene. One scene is corresponding to one video frame and one pair of driving behaviors. After that, each test scenario (set) owns around 600 scenes. For the point clouds obtained by our fusion algorithm are stored in PCD format, we employed a standard software to transform data into LAS format, which is an industry standard for LiDAR data.

Synchronization LiDAR scanners and speed sensors with video system are synchronized in advance to obtain valid data for driving policy learning. It is worthy to mention that synchronization is essential step before data collecting and we try the best to keep it precise (The bias is lower than 0.1 second).

Addressing Errors We triple-checked acquired data comprehensively and insured that videos, point clouds and driving behaviors are synchronous. Some unexpected errors were corrected after we manually re-calibrated to produce high-quality data in those time sections.

3.3. Data Structure

Our *DBNet* is a benchmark comprised of real driving videos, point clouds and standard driving behaviors. The data structure of dataset is illustrated in Figure 3. Compared with existing benchmark datasets, *DBNet* benchmark has combined 2D and 3D vision and moves the first attempt to leverage depth information (point clouds) to make the driving predictions. More details are shown in Table 1.

In consequence, the dataset is largely different from previous benchmarks for vision-based autonomous driving. To the best of our knowledge, it is the first benchmark for autonomous driving policy prediction combined with 2D and 3D information.

3.4. Features and Statistics

Our dataset has a list of excellent features illustrated in Figure 4, which are beneficial to policy learning. We have performed an in-depth analysis on properties of our dataset.

Scale We have used two kinds of LiDAR scanners to collect point clouds. They produced more than 1TB point clouds covering more than 100km distance, which is **two times** larger than previous KITTI. To the best of our knowledge, it is the largest public LiDAR data with corresponding vehicle status (speed and angle).

Diversity Our dataset contains a variety of traffic conditions, including local route, boulevard, primary road, mountain road, school areas, which contains a number of cross-roads, urban overpasses, ramp ways and hairpin bends. So our benchmark covers light, normal and heavy traffic situations. In addition, it also meets scenes with different numbers of pedestrians. For instance, there are many pedestrians in school areas but few in highway. More specifically, our

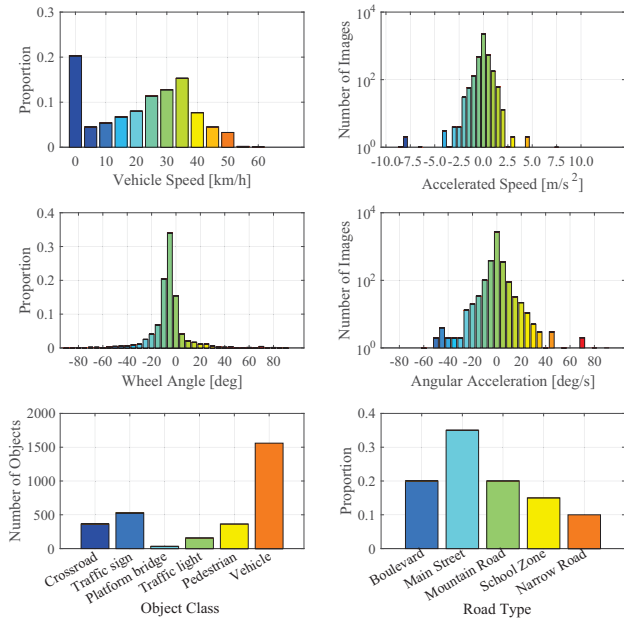


Figure 4. **Driving Behavior Distribution, Object Occurrence and Scenarios Statistics of our Dataset.** This figure demonstrates (from left to right and top to bottom): the vehicle and accelerated speed distribution, the wheel angle and angular acceleration distribution, different types of objects occurring and different traffic conditions (road types) in our sequence.

dataset contains more than 1500 cars, 500 road signs, 160 traffic lights and 363 crossroads and 32 footbridges. The diversity of real road scenes meets the real requirement for autonomous driving practice and makes our models more generic to operate in real scenarios.

Quality We use the Velodyne HDL-32E scanner to acquire 3D point clouds. HDL-32E can produce accurate depth information in mobile platforms. The depth range is 70 meters and it can achieve 2cm resolution. Additionally, the density of points is approximately 34,000 points per second so that abundant information is included in our point clouds data. We can clearly see buildings, trees, road lines traffic lights and even pedestrians. As for digital videos, vehicle’s dashboard camera produces 1920×1080 resolution videos with minor distortion while the vehicle moving at high speed.

4. Experimental Evaluation

This section answers how to leverage depth information and what we can achieve if current state-of-the-art techniques are used. Section 4.1 and 4.2 define prediction tasks in our experiment and evaluation metrics. Then representative approaches tested in our dataset are displayed in Section 4.3 and more details of the training process are supplemented in Section 4.4. Finally, we give experimental results

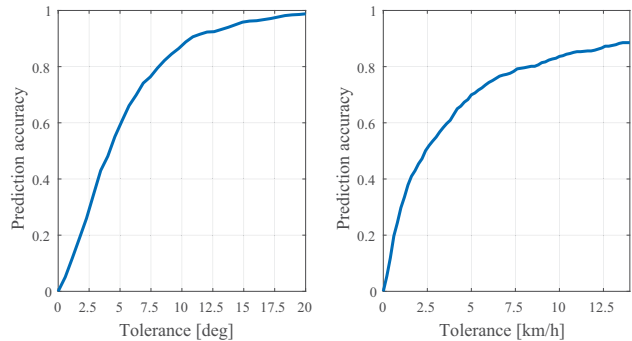


Figure 5. Prediction accuracy variation trends for trained model.

and discussion of our methods in Section 4.5 and 4.6.

4.1. Tasks

Driving behavior prediction tasks can be classified into two categories, discrete and continuous prediction.

Discrete action prediction It is to predict current probability distribution over all possible actions. The limitation of discrete prediction is that autonomous vehicle can only make decisions among limited predefined actions. For example, [27] defines four actions: straight, stop, left turn and right turn and policy decision becomes classification task. Obviously, the discrete task is not suitable for real driving, since it is too coarse to guide the vehicle driving.

Continuous prediction It is to predict current states of vehicles such as wheel angle and vehicle speed, which is a regression task. If driving policies on all real-world states can be predicted correctly, vehicles are expected to be driven successfully by trained model.

Therefore, we model driving process as a continuous prediction task. Our task is to train a model that receives multiple perception information including video frames and point clouds, thus predict correct steering angles and vehicle speeds.

4.2. Evaluation Metric

To evaluate the performance of driving behavior prediction, we investigated previous evaluation metrics. In [27], Xu *et al.* proposed a driving perplexity metric which is inspired by representative Markov model in linguistics. The action perplexity is defined as the exponent of the sum of entropy in sequential prediction events. Perplexity metric is a positive number smaller than one and the smaller score indicates the more accurate prediction.

Nevertheless, many researchers do not consider it as an effective metric. It is because that they do not give it real-world meaning and they believe perplexity value is more suitable for working as loss function in training process. For example, people do not know whether their models are effective enough or not, when the perplexity is 0.1 (seemingly

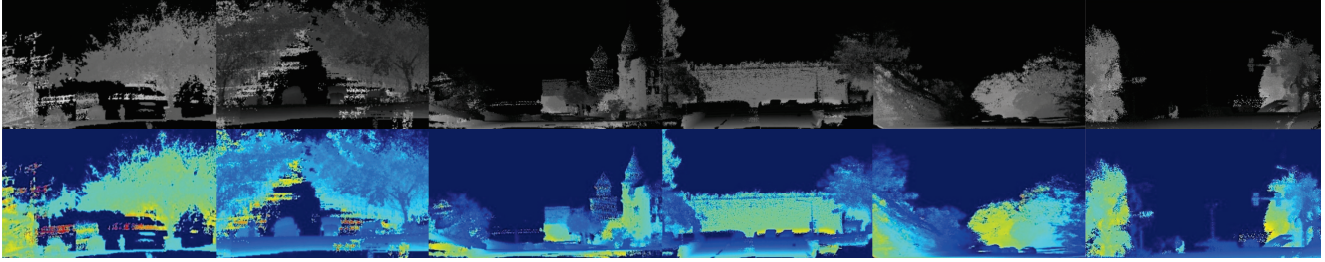


Figure 6. **Examples of gray and jet feature maps.** First row of this figure is gray feature maps and second row is the corresponding colored feature maps using jet color map. Depth information and spatial information (pedestrians, vehicles, trees, traffic lights, bridges, buildings and so on) can be obtained implicitly from maps.

low).

Accuracy metric is more intuitive in comparison to perplexity. More importantly, accuracy metric has been widely adopted [14, 20, 24] and applied to realistic scenarios [14]. If vehicles can always be very close to ground truth behavior, they will self-drive smoothly and safely.

Threshold In accuracy computing, we need to count how many predictions are correct. Therefore, a tolerance threshold is required. When the bias between prediction and ground truth is smaller than tolerance threshold, we count this prediction as a correct case. In fact, human drivers also have minor biases in driving, but it can be tolerated.

4.3. Representative Approaches

To demonstrate the effectiveness of depth information, we explore how well prediction models can achieve if current techniques are utilized. As before, we should introduce some learning tools and depth representation as prior knowledge. In the end, two current mainstream frameworks are presented.

4.3.1 Learning Tools

DNN. DNN has been built as a powerful class of models for extracting image features. In this paper, we adopt Resnet [12] and Inception-v4 [25], which are all the state-of-the-art approaches for extracting image features. These two models are pretrained on ImageNet [23] and fine-tuned in our experiments. Besides, we also use NVIDIA architecture [14] which is much smaller than networks mentioned above but has been tested well in real practice such as highway lane following.

LSTM. Driving policy prediction based on one frame (or small frame batch) only may miss information in the temporal domain. Therefore, we make use of long short-term memory (LSTM) [13] recurrent neural network to capture temporal events. LSTM is a well-improved recurrent neural network by introducing memory gates. It avoids gradient vanishing and is capable of learning long-term dependencies. Actually, LSTM is widely used in state-of-the-art

frameworks for predicting driving behaviors. In [9], LSTM-based framework is proposed for video classification. The championship [16] in *Udacity Self-Driving Challenge 2* also adopts this architecture.

4.3.2 Depth Representation

To leverage point clouds information effectively, we should seek a powerful depth representation. We have attempted different lines of techniques including point clouds reshaping, point clouds mapping and PointNet.

Point Clouds Mapping (PCM). We adopt the algorithm proposed in [28] to preserve geometrical information from raw point clouds. As shown in Figure 7, we firstly divide mobile LiDAR points into $h \times w$ grids on XOZ plane, where h and w are 600 and 1080 in our paper respectively. Each grid is represented by a single value to form a $h \times w$ feature map. The feature values of different grids are calculated using the algorithm in [28]. Intuitive idea behind it is to get the nearest points of Y coordinate in each grid. In short, feature map nicely extracts geometry information in point clouds. Figure 6 demonstrates some samples of feature maps and their jet color visualization in our dataset and Figure 7 depicts pipeline of this process.

PointNet. A novel PointNet architecture is put forward in [22] and opens a new direction for directly utilizing disordered point clouds. It directly takes disordered points as the input of neural networks and finally output the representation features. Currently, this distinct architecture outperforms other shape representation methods and achieves high accuracy.

4.3.3 Two Mainstream Frameworks

As is illustrated in Figure 8, inspired by plentiful previous works [14, 9, 16, 24], we decide to adopt two representative mainstream frameworks for policy prediction tasks, namely “DNN-only” and “DNN-LSTM”.

DNN-only. [14, 24] adopt this line of framework in their

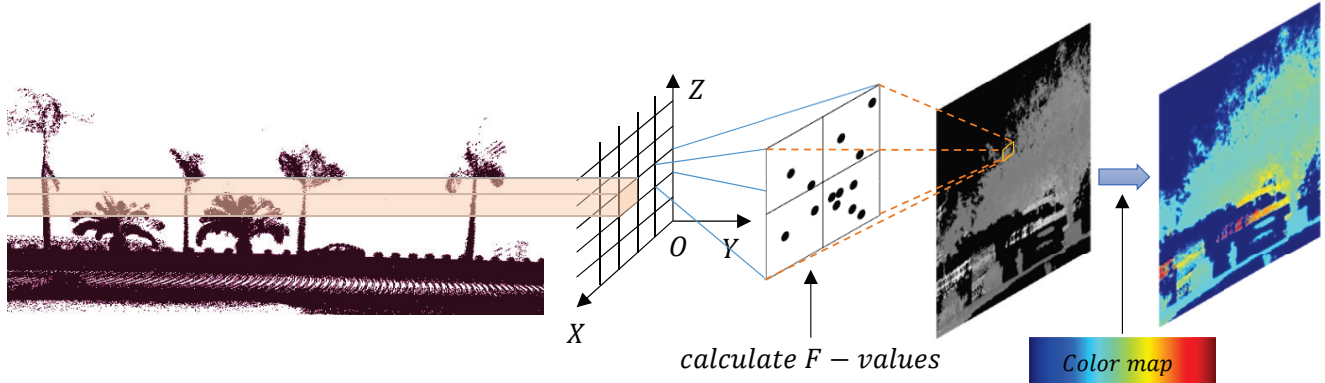


Figure 7. **The pipeline of extracting feature maps from raw point clouds.** Firstly, split XOZ plain into small grids, one of which is corresponding to specific one pixel in feature maps. Secondly, group raw points by projecting points into grids in XOZ . Then calculate feature values (F -values in figure) in each grid. Finally, generate feature maps by visualizing feature matrix and rendering with jet color map.

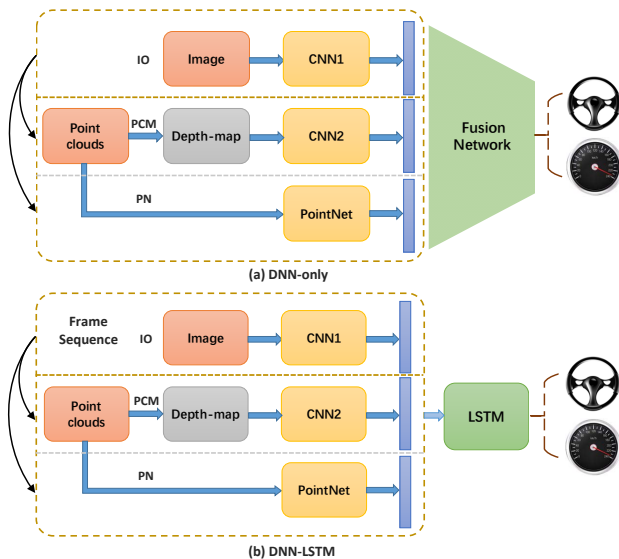


Figure 8. Network architecture of our prediction models.

driving prediction. The “DNN-only” framework is an end-to-end system that receives one (or a mini-batch) frame input and predicts driving behavior (seeing Figure 8 (a)). We employ three representative DNNs (NVIDIA, Resnet152 and Inception-v4) to extract features of RGB frames and 2D depth maps by PCM. The feature of point cloud is also extracted through PointNet. Thus, we concatenate two features (IO + PCM or IO + PointNet) as the input vector of one 1024 layer. This hidden layer is fully connected to fusion network, which outputs final driving behavior prediction.

DNN-LSTM. [9, 16] utilize this line of framework in their research. Different from “DNN-only”, we replace fusion network with stacked LSTM nets in “DNN-LSTM” framework. (seeing Figure 8 (b))

More specifically, two features of input data are extracted and concatenated like “DNN-only” framework. Then the concatenated vectors are sent into stacked LSTMs to get predictions.

4.4. Details of Training

The training samples are time-registered data including videos, point clouds, feature maps and driving behaviors. The captured videos are down-sampled to 1 fps. Frames are reshaped to different sizes which are suitable for three DNNs (NVIDIA: 66×200 , Resnet: 224×224 and Inception: 299×299). Besides, point clouds are down-sampled to 16384 points (16384×3) while adopting PointNet. Original point clouds which contain millions of points in each scene are used to generate feature maps directly to maintain enough information.

Our loss objective is a root-mean-square deviation (**RMSD**) to represent the sample standard deviation of the differences between predicted values and ground truth values. Vehicle speed and steering angle prediction models are trained individually. We attempt to train them jointly, but the performance is slightly worse than the cases where they are trained individually. We use a 80-20 training-testing split in our experiment and Adam optimizer [15] to minimize the loss function.

4.5. Results

Table 2 shows the accuracy of two aforementioned mainstream frameworks. Each setting is measured with the accuracy of predictions on wheel angles and vehicle speeds. Furthermore, we adopt three network structures to extract features of video frames and depth maps. The tolerance thresholds of vehicle speed and wheel angle are 5km/h and 6° , respectively. More results under different tolerance thresholds are available in the supplementary file. In Figure 5,

DNN Architecture	Metric	prediction accuracy of steering angle and vehicle speed					
		DNN only			DNN-LSTM		
		IO	PM	PN	IO	PM	PN
NVIDIA	angle	63.0%	67.1%	71.1%	77.9%	83.5%	81.6%
	speed	70.1%	69.2%	66.1%	70.9%	73.8%	76.8%
Resnet-152	angle	65.3%	70.8%	68.6%	78.4%	84.2%	82.7%
	speed	71.4%	72.6%	69.4%	71.9%	74.3%	78.3%
Inception-v4	angle	70.5%	71.1%	73.2%	78.3%	83.7%	84.8%
	speed	68.5%	70.3%	69.3%	70.3%	76.4%	77.3%

Table 2. **Performance of different combinations of basic networks with and without depth information.** IO represents feeding images only into networks. PM denotes plain images plus feature maps (PCM). PN denotes plain networks combined with PointNet architecture. The accuracies are measured within 6° or 5 km/h biases.

we display the trends of accuracy in IO model that adopted NVIDIA architecture with tolerance threshold increasing.

Overall, “DNN-LSTM” outperforms “DNN-only” setting, which means that feeding videos-frames in the sequence to networks helps autonomous vehicles to make decisions. It is because that independent image neglects important long-term event information.

More importantly, it is fascinating that utilizing depth information improves the accuracy of prediction greatly in comparison to the use of video frames only (IO in Table 2). It again verifies the importance of depth information for driving behavior prediction and also shows the great potentials to improve driving prediction task by designing advanced depth representations and effective ways of extracting point features.

4.6. Discussion

Firstly, regarding autonomous driving prediction process as a temporally sequential model keeps more essential information and gets better results. The system that holds memory in sequence is suitable for deciding future trends.

Secondly, depth information contributes to more reliable results and it helps vehicles learn driving policies more effectively. In consequence, we believe that future autonomous vehicles are likely to equip with 3D-scanners in order to gain comprehensive perception like the human.

Thirdly, although we use powerful DNNs such as Resnet to extract features, the improvement is still minor, which means we may meet an upper-board for 2D vision.

Finally, the large gap among different ways of using depth information tells us current depth representation is still an open problem that is not fully resolved. Even though our paper has attempted various depth representation and seems to produce good results, we still believe that there are huge potentials for depth utilization.

5. Conclusion and Future Work

In this paper, we have proposed a *DBNet* benchmark dataset, which is among the first attempts to utilize point clouds to help driving policy learning. We have performed an in-depth analysis of how important depth information is, how to leverage depth information and what we can achieve by leveraging current representative techniques. From preliminary experiment results, we found that the utilization of depth information had resulted in considerable promotion in prediction performance. However, it still has a large room to improve the usage of point cloud information. We believe our benchmark dataset will open one door to study policy learning by providing extra but significant point clouds.

Our paper has attempted varied ways to take the advantages of point clouds in the benchmark. Even though these methods has helped networks to learn driving policies, they are far from optimal solutions for insufficient utilization of point clouds. How to make the best of these information remains to be further studied.

Moreover, although the supervised end to end segmentation may improved performance greatly, it may be too expensive to annotate the training data. Unlike plane 2D images or videos, point clouds contains rich depth information and geometrical features. In consequence, it is feasible to segment point clouds in unsupervised ways [2, 26]. We believe that affordable weakly-supervised or unsupervised coarse segmentation will help generate quantities of annotated data and learn driving policies.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China under Grants 61601392 and 61772332.

References

- [1] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez. *Road Scene Segmentation from a Single Image*, pages 376–389. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [2] P. Babahajiani, L. Fan, J.-K. Kämäräinen, and M. Gabbouj. Urban 3d segmentation and modelling from street view images and lidar point clouds. *Machine Vision and Applications*, May 2017.
- [3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.
- [4] A. Bauer, K. Dietz, G. Kolling, W. Hart, and U. Schiefer. The relevance of stereopsis for motorists: a pilot study. *Graefe’s Archive for Clinical and Experimental Ophthalmology*, 239(6):400–406, Jul 2001.
- [5] A. Brics, B. Nagy, and C. Benedek. Instant object detection in lidar point clouds. *IEEE Geoscience and Remote Sensing Letters*, 14(7):992–996, July 2017.
- [6] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 2722–2730, Washington, DC, USA, 2015. IEEE Computer Society.
- [7] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, June 2016.
- [8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016.
- [9] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):677–691, 2017.
- [10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.
- [13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [14] J. Kim and C. Park. End-to-end ego lane estimation based on sequential transfer learning for self-driving cars. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Honolulu, HI, USA, July 21-26, 2017*, pages 1194–1202. IEEE Computer Society, 2017.
- [15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [16] Kormanda. Udacity self driving car challenge 2. <https://github.com/udacity/self-driving-car/>. Accessed: 2017-11-10.
- [17] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.
- [18] A. Martinovi, J. Knopp, H. Riemenschneider, and L. V. Gool. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4456–4465, June 2015.
- [19] A. McKnight, D. Shinar, and B. Hilburn. The visual and driving performance of monocular and binocular heavy-duty truck drivers. *Accident Analysis and Prevention*, 23(4):225 – 237, 1991.
- [20] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun. Off-road obstacle avoidance through end-to-end learning. In Y. Weiss, P. B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 739–746. MIT Press, 2006.
- [21] D. A. Pomerleau. Advances in neural information processing systems 1. chapter ALVINN: An Autonomous Land Vehicle in a Neural Network, pages 305–313. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.
- [22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017.
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [24] E. Santana and G. Hotz. Learning a driving simulator. *CoRR*, abs/1608.01230, 2016.
- [25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In S. P. Singh and S. Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4278–4284. AAAI Press, 2017.
- [26] J. Xie, M. Kiefel, M. Sun, and A. Geiger. Semantic instance annotation of street scenes by 3d to 2d label transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3688–3697. IEEE Computer Society, 2016.
- [27] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3530–3538. IEEE Computer Society, 2017.

- [28] B. Yang, Z. Wei, Q. Li, and J. Li. Automated extraction of street-scene objects from mobile lidar point clouds. *International Journal of Remote Sensing*, 33(18):5839–5861, 2012.
- [29] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 669–677. IEEE Computer Society, 2016.