One Thousand and One Hours:
Self-driving Motion Prediction Dataset

John Houston **Guido Zuidhof** **Luca Bergamini** **Yawei Ye**
Long Chen **Ashesh Jain** **Sammy Omari** **Vladimir Iglovikov** **Peter Ondruska**
Lyft Level 5
level5data@lyft.com

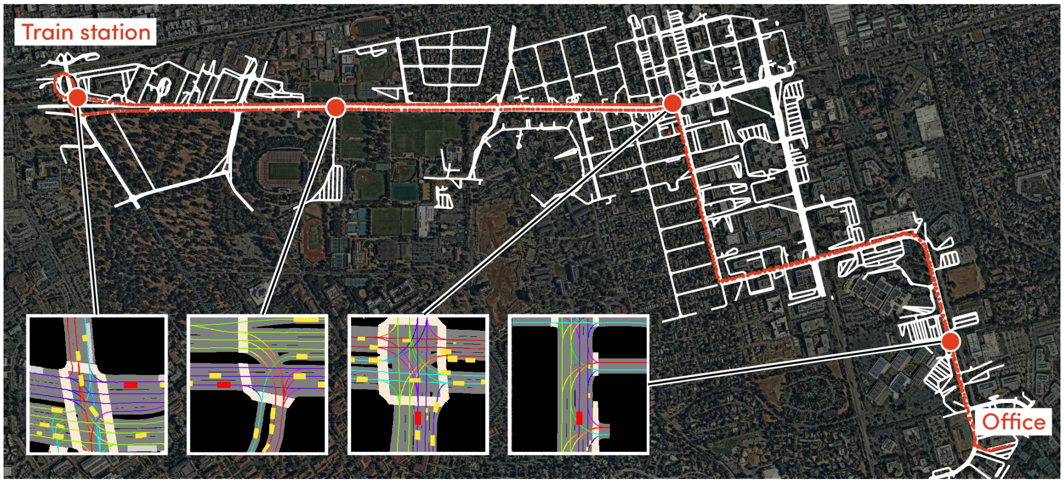

Figure 1: An overview of the released dataset for motion modelling, consisting of 1,118 hours of recorded self-driving perception data on a route spanning 6.8 miles between the train station and the office (red). The examples on the bottom-left show released scenes on top of the high-definition semantic map that capture road geometries and the aerial view of the area.

Abstract: Motivated by the impact of large-scale datasets on ML systems we present the largest self-driving dataset for motion prediction to date, containing over 1,000 hours of data. This was collected by a fleet of 20 autonomous vehicles along a fixed route in Palo Alto, California, over a four-month period. It consists of 170,000 scenes, where each scene is 25 seconds long and captures the perception output of the self-driving system, which encodes the precise positions and motions of nearby vehicles, cyclists, and pedestrians over time. On top of this, the dataset contains a high-definition semantic map with 15,242 labelled elements and a high-definition aerial view over the area. We show that using a dataset of this size dramatically improves performance for key self-driving problems. Combined with the provided software kit, this collection forms the largest and most detailed dataset to date for the development of self-driving machine learning tasks, such as motion forecasting, motion planning and simulation.

Keywords: Dataset, Self-driving, Motion prediction

1 Introduction

The availability of large-scale datasets has been a large contributor to AI progress in the recent decade. In the field of self-driving vehicles (SDVs), several datasets, such as [1, 2, 3], enabled great progress within the development of perception systems [4, 5, 6, 7]. These allow an SDV to process

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

LiDAR and camera sensors for understanding positions of other traffic participants including cars, pedestrians and cyclists around the vehicle.

Perception, however, is only the first step in the modern self-driving pipeline. Much work remains to be done around data-driven motion prediction of traffic participants, trajectory planning and simulation, before SDVs can become a reality. Datasets for developing these methods differ from those used for perception in that they require large amounts of behavioural observations and interactions. These are obtained by combining the output of perception systems with an understanding of the environment - in the form of a semantic map that contains priors over expected behaviour. Broad availability of datasets for these downstream tasks is much more limited though, and mostly available only to large-scale industrial efforts in the form of in-house collected data. This limits progress within the computer vision and robotics communities to advance modern machine learning systems for these important tasks.

In this work, we share the largest and most detailed dataset to date for training motion forecasting and planning solutions. We are motivated by the scenario of a self-driving fleet serving a single, high-demand route - rather than serving a broad area. We consider this to be a more feasible deployment strategy for ridesharing, since SDVs can be allocated to particular routes while human drivers serve the remaining traffic. This focus allows setting better bounds on required system performance and accident likelihood, both key factors for real-world self-driving deployment.

In summary, the released dataset consists of:

- The largest dataset to date for motion prediction, containing 1,000 hours of traffic scenes that capture the motions of traffic participants around 20 self-driving vehicles, driving over 26,000 km along a suburban route.
- The most detailed high-definition (HD) semantic map of the area, counting over 15,000 human annotations including 8,500 lane segments.
- A high-resolution aerial image of the area, spanning 74 km^2 at a resolution of 6 cm per pixel, providing further spatial context about the environment.
- A Python software library L5Kit for accessing and visualising the dataset.
- Baseline machine learning solutions for the motion forecasting and motion planning task that demonstrate the impact of large-scale datasets.

2 Related Work

In this section we review related existing datasets for training SDV systems from the viewpoint of a classical state-of-the-art self-driving stack as summarised in Figure 2. In this stack the raw sensor input is first processed by a perception system to estimate positions of nearby vehicles, pedestrians, cyclists and other traffic participants. Next, the future motion and intent of these actors is estimated (also called motion prediction or forecasting) and used for planning SDV trajectory. In Table 1 we summarise the current leading datasets for training machine learning solutions for different components of this stack, focusing mainly on the perception and prediction components.

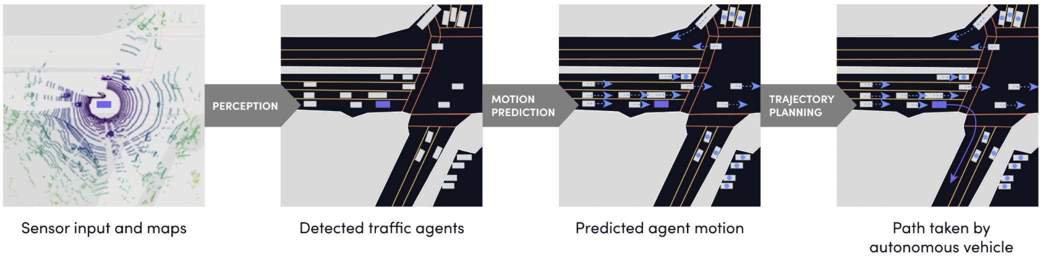

Figure 2: An example of a state-of-the-art self-driving pipeline. First, the raw LiDAR and camera data are processed to detect the positions of nearby objects around the vehicle. Then, their motion is predicted to allow the SDV to plan a safe collision-free trajectory. The released dataset enables the modelling of a motion prediction component.

Name	Size	Scenes	Map	Annotations	Task
KITTI [1]	6h	50	None	3D bounding boxes	Perception
Oxford RobotCar [8]	71h	100	None	-	Perception
Waymo Open Dataset [9]	10h	1000	None	3D bounding boxes	Perception
ApolloScape Scene Parsing [10]	2h	-	None	3D bounding boxes	Perception
Argoverse 3D Tracking v1.1 [2]	1h	113	Lane center lines, lane connectivity	3D bounding boxes	Perception
Lyft Perception Dataset [3]	2.5h	366	Rasterised road geometry	3D bounding boxes	Perception
nuScenes [11]	6h	1000	Rasterised road geometry	3D bounding boxes, trajectories	Perception, Prediction
ApolloScape Trajectory [12]	2h	103	None	Trajectories	Prediction
Argoverse Forecasting v1.1 [2]	320h	324k	Lane center lines, lane connectivity	Trajectories	Prediction
Ours	1,118h	170k	Road geometry, aerial map, crosswalks, traffic lights state, ...	Trajectories	Prediction, Planning

Table 1: A comparison of various self-driving datasets available today. Our dataset surpasses all others in terms of size, as well as level of detail of the semantic map (see Section 3).

Perception datasets The perception task is usually framed as the supervised task of estimating 3D positions of nearby objects around the SDV. Deep learning approaches are now state-of-the-art for most subproblems relevant for autonomous driving, such as 3D object detection and semantic segmentation [7, 6, 5, 4].

Among the datasets for training these systems the KITTI dataset [1] is the most common benchmarking dataset for many computer vision and autonomous driving related tasks. It contains around 6 hours of driving data, recorded from front-facing stereo cameras, LiDAR and GPS/IMU sensors. 3D bounding box annotations are available, including class annotations such as cars, trucks and pedestrians. The Waymo Open Dataset [9] and nuScenes [11] are of similar size and structure, providing 3D bounding box labels based on fused sensory inputs. The Oxford RobotCar dataset [8] also allows application for visual tasks, but focus more lies on localisation and mapping, rather than object detection.

Our dataset's main target is not to train perception systems. Instead, it is a product of an already trained perception system used to process large quantities of new data for motion prediction.

Prediction datasets The prediction task we focus on in this paper builds on top of perception by trying to predict positions of detected objects a few seconds into the future. In order to obtain good results, one needs significantly more detailed information about the environment including, for example, semantic maps that encode possible driving behaviour to reason about future behaviours.

Deep learning solutions leveraging birds-eye-view (BEV) representations of scenes [13, 14, 15, 16, 17, 18] or graph neural networks [19, 20] have established themselves as the leading solutions for this task. Representative large-scale datasets for training these systems are, however, rare. The above mentioned solutions were developed almost exclusively by industrial labs leveraging internal proprietary datasets.

The most relevant existing open dataset is the Argoverse Forecasting dataset [2] providing 300 hours of perception data and a lightweight HD semantic map encoding lane center positions. Our dataset differs in three substantial ways: 1) Instead of focusing on a wide city area we provide 1000 hours of data along a single route. This is motivated by the assumption that, particularly in ride-hailing applications, the first deployments of self-driving fleets are more likely to occur along few high-demand routes. This makes it possible to set bounds for requirements and quantify accident risk more precisely. 2) We are contributing higher-quality scene data by providing the full perception output including bounding boxes and class probabilities instead of pure centroids. In addition, our semantic map is more detailed: it counts more than 15,000 human annotations instead of only lane centers. 3) We further provide a high-resolution aerial image of the area. This is motivated by the fact that much of the information encoded in the semantic map is implicitly accessible in the aerial form. Providing this map can, therefore, unlock the development of semantic-map free solutions.

Planning datasets Planning and executing an SDV trajectory is the final component in the autonomous driving stack. It is also the component having received the least attention from the community and exhibited the least progress when counting ML solutions. One of the reasons is that it is difficult to model and evaluate this task as an ML problem trained from real data. Closed-loop evaluation of a driving policy requires collecting new data that are not present in the dataset.

As a result, a majority of research work instead focuses on open-loop ego-motion forecasting [21, 22]. However, it is known that such methods significantly underperform in closed-loop evaluation due to the distribution shift [23]. Consequentially, most of state-of-the-art solutions used in industry resort to traditional trajectory optimisation systems based on hand-crafted cost functions instead of machine learning.

In our work we follow the recent approach [24] leveraging imitation learning and perturbations. Authors, however, demonstrated this approach only on a proprietary dataset with no available open equivalent. As part of our work we provide an ML planning baseline inspired by [24] and show it can be effectively trained and evaluated using our dataset, yielding the first such open evaluation. We show that, similarly to motion forecasting, the performance of ML planning significantly improves with the amount of training data - showing much promise for ML planning solutions to the future together with the need of datasets to train them.

3 Dataset

Here we outline the details of the released dataset, including the process that was used to construct it. An overview of different dataset statistics can be found in Table 2.

The dataset has three components:

1. 170,000 scenes, each 25 seconds long, capturing the movement of the self-driving vehicle, traffic participants around it and traffic lights state.
2. A high-definition semantic map capturing the road rules, lane geometry and other traffic elements.
3. A high-resolution aerial picture of the area that can be used to further aid in prediction.

3.1 Scenes

The dataset consists of 170,000 scenes, each 25 seconds long, totalling over 1,118 hours of logs. Example scenes are shown in Figure 4. All logs were collected by a fleet of self-driving vehicles driving along a fixed route. The sensors for perception include 7 cameras, 3 LiDARs, and 5 radars

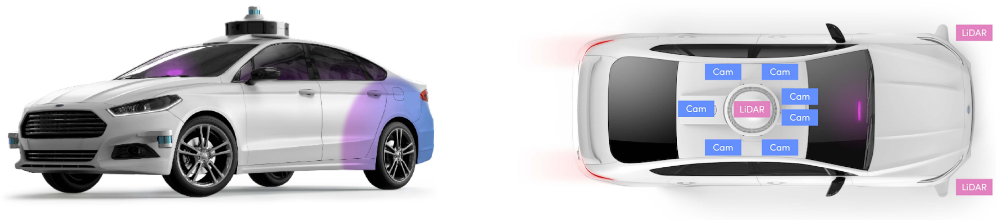

Figure 3: The self-driving vehicle configuration used to collect the data. Raw data from LiDARs and cameras were processed by a perception system to generate the dataset, capturing the poses and motion of nearby vehicles.

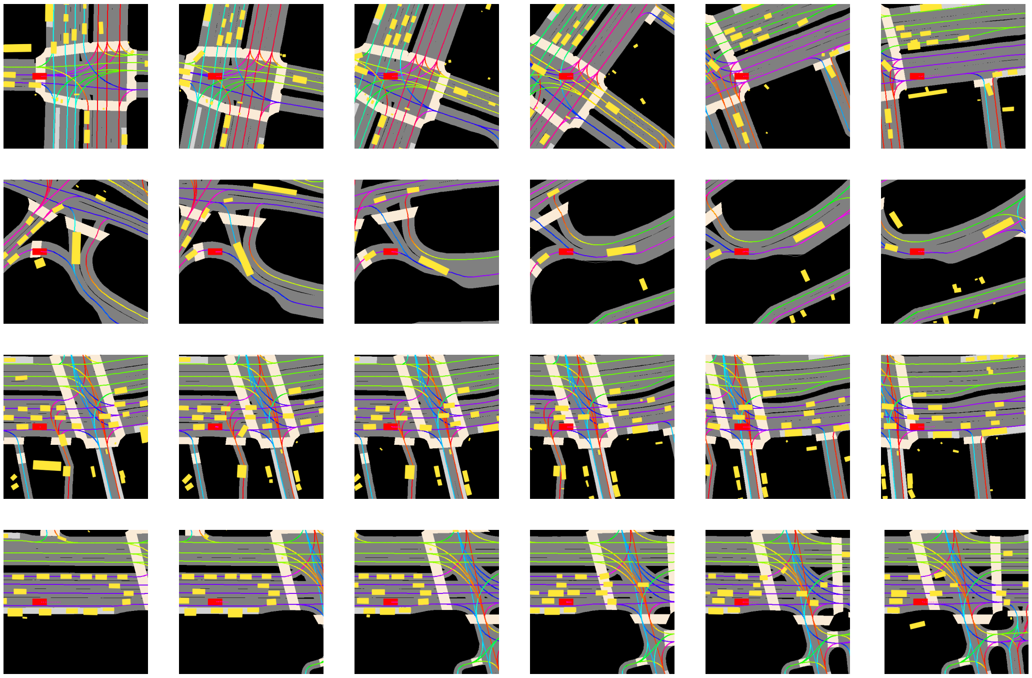

Figure 4: Examples from the scenes in the dataset, projected over a BEV of the rasterised semantic map. The self-driving vehicle is shown in red, other traffic participants in yellow, and lane colours denotes driving direction. The dataset contains 170,000 such sequences, each 25 seconds long with sensor data at 10Hz.

Statistic	**Value**
# self driving vehicles used	20
Total data set size	1,118 hours / 26,344 km / 162k scenes
Training set size	928 hours / 21,849 km / 134k scenes
Validation set size	78 hours / 1,840 km / 11k scenes
Test set size	112 hours / 2,656 km / 16k scenes
Scene length	25 seconds
Total # of traffic participant observations	3,187,838,149
Average # of detections per frame	79
Labels	Car: 92.47% / Pedestrian: 5.91% / Cyclist: 1.62%
Semantic map	15,242 annotations / 8,505 lane segments
Aerial map	74 km^2 at 6 cm per pixel

Table 2: Statistics of the released dataset.

(see Figure 3). The sensors are positioned as follows: one LiDAR is on the roof of the vehicle, and two LiDARs on the front bumper. The roof LiDAR has 64 channels and spins at 10 Hz, while the bumper LiDARs have 40 channels. All seven cameras are mounted on the roof and together have a 360° horizontal field of view. Four radars are also mounted on the roof, and one radar is placed on the forward-facing front bumper.

The dataset was collected between October 2019 and March 2020. It was captured during daytime, between 8 AM and 4 PM. For each scene we detected the visible traffic participants, including vehicles, pedestrians, and cyclists. Each traffic participant is internally represented by a 2.5D cuboid, velocity, acceleration, yaw, yaw rate, and a class label. These traffic participants are detected using our in-house perception system similar to [7, 6, 5, 4] and trained using a dataset similar to [3] along the same route. It fuses data across multiple modalities to produce a 360° view of the world surrounding the SDV. Table 2 outlines some more statistics for the dataset.

Property	Values
Lane boundaries	sequence of (x, y, z) coordinates
Lane connectivity	possible lane transitions
Driving directions	one way, two way
Road class	primary, secondary, tertiary, ...
Road paintings	solid, dashed, colour
Speed limits	mph
Lane restrictions	bus only, bike only, turn only, ...
Crosswalks	position
Traffic lights	position, lane association
Traffic signs	stop, turn, yield, parking, ...
Restrictions	keep clear zones, no parking, ...
Speed bumps	position

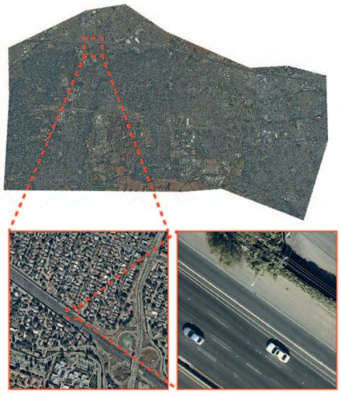

Figure 5: Elements of the provided HD semantic map (left) and overhead aerial map surrounding the route (right). We provide 15,242 human annotations including 8,505 individual lane segments. The aerial map covers 74 km^2 at a resolution of 6 cm per pixel.

We split the dataset into train, validation and test set using a 83–7–10% ratio, where a particular SDV only contributes to a single set. We encode the dataset in the form of n-dimensional compressed zarr arrays. The zarr format[1] was chosen to represent individual scenes. It allows for fast random access to different portions of the dataset while minimising the memory footprint, which allows efficient distributed training on the cloud.

3.2 High-definition semantic map

The HD semantic map that we provide encodes information about the road itself as well as various traffic elements along the route totalling 15,242 labelled elements, including 8,505 lane segments. This map was created by human curators who annotated the underlying localisation map, which in turn was created using a simultaneous localisation and mapping (SLAM) system. Given the use of SLAM, the position of the SDV is always known with centimetre-grade accuracy. Thus, the information in the semantic map can be used both for planning driving behaviour and for anticipating the future movements of other traffic participants.

The semantic map is given in the form of a protocol buffer[2]. We provide precise road geometry through the encoding of the lane segments, their connectivity and other properties (as summarised in Figure 5).

3.3 Aerial map

The aerial map captures the area of Palo Alto surrounding the route at a resolution of 6 cm per pixel. It enables the use of spatial information to aid with motion prediction. Figure 5 shows the map coverage and the level of detail. The covered area of 74 km^2 is provided as 181 GeoTIFF tiles of size 10560×10560 pixels, each spanning approximately 640×640 meters.

4 Development tools

In combination with the dataset, we are releasing a Python toolkit called L5Kit[3]. It provides access to data loading and visualisation functionality and implementation for two baseline tasks of motion forecasting and SDV motion planning.

Multi-threaded data loading and sampling We provide an API that can sample scenes and load the data efficiently. Scenes can be sampled from multiple points of view: for planning of the SDV motion path, we can center the scene around the SDV. For predicting the motions of other traffic participants, we provide functionality to recenter the scene around those traffic participants.

[1] https://zarr.readthedocs.io/

[2] https://developers.google.com/protocol-buffers

[3] https://github.com/lyft/l5kit

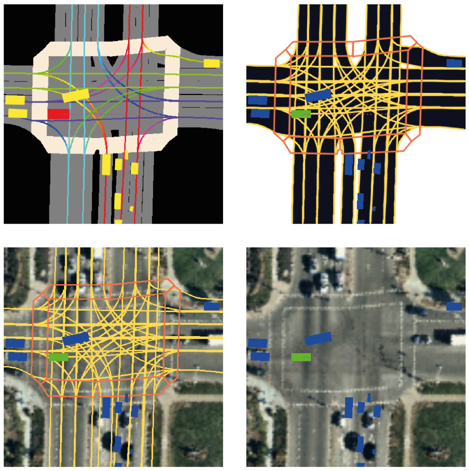

Figure 6: Examples of different BEV scene rasterisations that can be made using the associated software development kit. These can be used, for example, as input to convolutional neural network architectures.

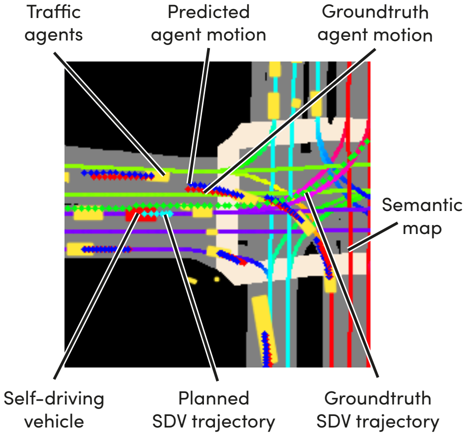

Figure 7: Example output of the agent motion forecasting and SDV motion planning baseline, supplied as part of the software development kit.

Customisable BEV scene rasterisation

We provide several functions to visualise and rasterise a sampled scene. Our visualisation package can draw additional information, such as the future trajectory, onto an RGB image and save files as images, GIFs or full scene videos.

We support several different rasterisation modes for creating a meaningful representation for the underlying image. Figure 6 shows example images generated by the different modes and created from either the semantic map (upper right image) or the aerial map (lower right), or a combination of both (lower left). Such images can then be used as input to a conventional machine learning pipeline akin to [24, 16].

Motion forecasting baseline In motion forecasting, the task is to predict the expected future (x,y)-positions over a $T = 5$-second-horizon for different traffic participants in the scene given their current (and sometimes also historical) positions. We use a ResNet-50 backbone [25] with L_2 loss that was trained on 224×224 pixel BEV rasters centered around different vehicles of interest. We also provide a history of the vehicles' movements over the past few seconds by simply stacking BEV rasters. This allows the network to implicitly compute an agent's current velocity and heading. Figure 7 displays typical predictions after training each vehicle on this architecture for 38,000 iterations with a batch size of 64.

Table 3 summarises the displacement error (the L_2-norm between the predicted point and the true position at horizon T) for a network trained on the semantic map at various prediction horizons for

Configuration		**Displacement error [m]**						
Training size	History length	@0.5s	@1s	@2s	@3s	@4s	@5s	ADE
1%	0 sec	0.44	0.84	1.62	2.41	3.26	4.22	**2.47**
10%	0 sec	0.36	0.69	1.29	1.91	2.57	3.30	**1.95**
100%	0 sec	0.31	0.59	1.10	1.61	2.14	2.74	**1.64**
1%	1 sec	0.16	0.30	0.59	0.97	1.47	2.08	**1.08**
10%	1 sec	0.15	0.27	0.54	0.87	1.30	1.84	**0.96**
100%	1 sec	0.13	0.23	0.44	0.70	1.03	1.46	**0.77**

Table 3: Performance of the motion forecasting baseline in open-loop evaluation. The performance continues increasing with the size of the training set, both for models using history and not using it. We list the displacement error for different prediction horizons.

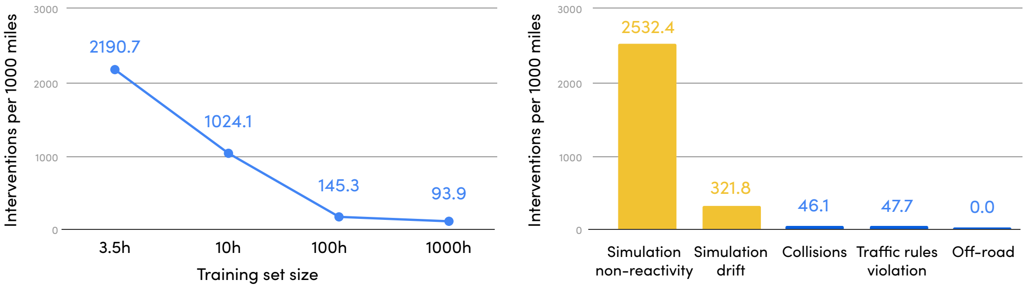

Figure 8: Performance of the ML planning task in a closed-loop evaluation. The SDV is free to take actions and diverge from the recorded behaviour while other traffic participants follow the recorded log. The left plot shows the performance of the SDV increasing dramatically with the amount of training data. Right is the qualitative performance of a model trained on 1000h.

different history lengths , as well as the displacement error averaged over all timesteps (ADE). To evaluate the impact of the size of the dataset, we train models using different sample sizes from the training set. As seen, adding history improves the prediction performance as the network gains knowledge about vehicle speed and acceleration. Moreover, the performance keeps increasing with the size of the training set.

Motion planning baseline As a related experiment, we employ our dataset for the SDV planning task. The task here is to predict and also execute a trajectory for the SDV. This allows the actual SDV pose to be different in future timesteps than recorded in the log. As outlined in [23] using motion forecasting is not a suitable solution for this problem as it suffers from accumulation of errors due to broken i.i.d assumption. Our implementation is based on [24] of augmenting motion forecasting model with synthetic perturbations to mitigate this problem. The network is trained to accept BEV rasters of size 224×224 pixels (centered around the SDV this time) to predict future (x,y)-positions over a 5-second-horizon. We use the same architecture and loss as in the motion forecasting task, and train for a similar number of iterations.

Testing of the system constitutes simulating SDV behaviour in 25 sec. long scenes from the test dataset. In each episode each traffic participant follows logged behaviour. The SDV is, however, controlled by the network and is free to take any action and diverge from its original behaviour. Results are summarised in Figure 8. This simulation is not perfect as it is non-reactive, but still gives valuable insights into SDV's actual performance. We count only errors caused by the SDV's actions (collisions, traffic rules violations, off-road events) and not limitations of blind log-replay simulation (e.g. other cars colliding into the SDV due to non-reactivity, diverging too far from the log position rendering perception ineffective), which dominate the total errors. Similarly to motion forecasting the performance keeps increasing with more data. This suggests that the performance is not saturated and both tasks can benefit from even larger datasets in the future.

5 Conclusion

The dataset introduced in this paper is the largest and most detailed dataset available for training prediction and planning solutions. It is three times larger and significantly more descriptive than the current best alternatives. We show that this difference yields a meaningful increase in performance for both the motion forecasting and motion planning task. This is in-line with the intuition, that datasets are key ingredients in unlocking and making large-scale machine learning systems work well. These datasets, however, are not available for everyone as they often come from proprietary industrial efforts.

We believe that publishing this dataset marks an important next step towards the democratisation within the development of self-driving applications. This, in turn, can result in faster progress towards a fully autonomous future. At the same time, we observe the performance of the motion forecasting and motion plannning tasks still keeps increasing with the size of the training data. This suggests, that even larger datasets counting tens of thousands or even millions of hours can be desirable in the future, together with algorithms that can take advantage of them.

Acknowledgments

This work was done thanks to many members of the Lyft Level 5 team. Specifically, we would like to thank Emil Praun, Christy Robertson, Oliver Scheel, Stefanie Speichert, Liam Kelly, Chih Hu, Usman Muhammad, Lei Zhang, Dmytro Korduban, Jason Zhao, Hugo Grimmett and Luca Del Pero.

References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *Int. Journal of Robotics Research (IJRR)*, 2013.

[2] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. 2019.

[4] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun. Multi-task multi-sensor fusion for 3d object detection. *Int. Conf. on Computer Vision and Pattern Recognition*, 2019.

[6] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000km: The oxford robotcar dataset. *Int. Journal of Robotics Research (IJRR)*, 2017.

[9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. 2019.

[10] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The apolloscape open dataset for autonomous driving and its application. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[11] 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. *arXiv preprint arXiv:1903.11027*, 2019.

[12] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *AAAI Conference on Artificial Intelligence*, 2019.

[13] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Int. Conf. on Computer Vision and Pattern Recognition*, 2018.

[15] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. K. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] H. Cui, V. Radosavljevic, F. Chou, T. Lin, T. Nguyen, T. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. *Int. Conf. on Robotics and Automation (ICRA)*, 2019.

[17] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. 2019.

[18] J. Hong, B. Sapp, and J. Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] S. Casas, C. Gulino, R. Liao, and R. Urtasun. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[20] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] J. Philion, A. Kar, and S. Fidler. Learning to evaluate perception models using planner-centric metrics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14055–14064, 2020.

[23] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[24] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science and Systems (RSS)*, 2019.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.