Vehicle Trajectory Prediction by Transfer Learning of Semi-Supervised Models

Nick Lamm  
Department of Computer Science  
Columbia University  
New York, USA  
nl2680@columbia.edu

Shashank Jaiprakash  
Department of Computer Science  
Columbia University  
New York, USA  
sj3003@columbia.edu

Malavika Srikanth  
Department of Computer Science  
Columbia University  
New York, USA  
ms5908@columbia.edu

Iddo Drori  
Department of Computer Science  
Columbia University  
New York, USA  
idrori@cs.columbia.edu

Abstract—In this work we show that semi-supervised models for vehicle trajectory prediction significantly improve performance over supervised models on state-of-the-art real-world benchmarks. Moving from supervised to semi-supervised models allows scaling-up by using unlabeled data, increasing the number of images in pre-training from Millions to a Billion. We perform ablation studies comparing transfer learning of semi-supervised and supervised models while keeping all other factors equal. Within semi-supervised models we compare contrastive learning with teacher-student methods as well as networks predicting a small number of trajectories with networks predicting probabilities over a large trajectory set. Our results using both low-level and mid-level representations of the driving environment demonstrate the applicability of semi-supervised methods for real-world vehicle trajectory prediction.

I. INTRODUCTION

Predicting the trajectory of a vehicle in a multi-agent environment is a challenging and critical task for developing safe autonomous vehicles. State-of-the-art models rely on a representation of the environment from either direct, low-level input from sensors on the vehicle, or from a mid-level representation of the scene, which is commonly a map annotated with agent positions. Both of these approaches rely on a model to encode either camera data in the low-level case or annotated maps in the mid-level case. We show an example of both types of representations in Figure 1. Mid-level representations as depicted in the top-left are used to predict candidate trajectories as show in the top-right. Low-level representations such as camera data shown in the bottom-left can be used in an end-to-end fashion to predict steering angles as illustrated in the bottom-right. To encode these input representations, rather than training a model from scratch, state-of-the-art models use transfer learning with a model pre-trained on a supervised task [1], [2] such as ImageNet classification. We perform an ablation study comparing transfer learning of supervised and semi-supervised models, while keeping all other factors equal, and show that semi-supervised models perform better than supervised models for both low-level and mid-level representations.

We demonstrate this comparison on state-of-the-art methods for vehicle trajectory prediction. For a low-level representation, we use the winning architecture of the ICCV 2019: Learning-to-Drive Challenge, which uses vehicle camera footage to predict the future speed and steering wheel angle [3]. For a mid-level representation, we use CoverNet [1] and multiple trajectory prediction (MTP) [4], two multi-modal approaches that take an annotated map image as input. In all of these cases, we keep the architecture and computational resources the same, and compare semi-supervised and supervised models to encode the representation. Semi-supervised models have demonstrated state-of-the-art performance on computer vision benchmarks since they are able to learn from unlabeled datasets orders of magnitude larger than available labeled data. Notably, although annotated maps are not representative of the images in the datasets used to pre-train these models, they share common features with the mid-level map representation.

Our results demonstrate three key contributions (described in Section III):

1) Semi-supervised models significantly improve upon supervised models using both low-level and mid-level representations.

2) Contrastive semi-supervised learning (SimCLR [5]) outperforms teacher-student semi-supervised learning (ResNeXt-101 32x4d SSL and SWSL [6]).

3) Using semi-supervised models for predicting probabilities of a large set of trajectories with CoverNet [1] results in significant performance improvement over supervised models across both uni-modal and multi-modal metrics (up to 40.1%); whereas using semi-supervised models for predicting a small set of trajectories with MTP [4] results in significant performance improvement only on uni-modal metrics (up to 17.3%).
In another approach \cite{13}, a multi-modal multi-task method on certain agents or other features of the scene context. multi-head attention \cite{2, 14}, allowing the model to focus in the environment. This approach has been extended using predicting a probability distribution over trajectories for agents works \cite{1}, \cite{2}, \cite{4}, \cite{10}–\cite{13} use a multi-modal approach, plausible trajectories that the vehicle might take. Several associated probabilities, especially when there are multiple networks for trajectory prediction.

A feature vector, which is used in a system of neural low-level approach, the map image is fed through a model to mid-level representation of the entire scene \cite{9}. Similar to the is then rasterized into an image, which serves as a compact map for capturing the scene, losses, and driving goals. Recent semi-supervised models extend and improve upon supervised ResNets by using orders of magnitude more unlabeled input samples \cite{5}, \cite{6}, \cite{20}, as well as output sample statistics \cite{21}, with good results in real-world applications.

**A. Related Work**

Low-level approaches to trajectory prediction use sensor data recorded by the vehicle, such as mounted cameras, as direct input to a model \cite{7}, \cite{8}. These approaches use a model to encode the raw pixels from the camera footage into a feature vector. We evaluate such a low-level representation \cite{8}, which uses front-facing camera images in combination with a vector of semantic map features to predict a vehicle’s future steering wheel angle and speed.

Many approaches instead use a mid-level representation of the environment as input to the model. This commonly involves generating a map of the scene and annotating it with past and current positions of all other agents, using color to designate semantic categories of agents as well as static entities such as road boundaries and crosswalks. The map is then rasterized into an image, which serves as a compact mid-level representation of the entire scene \cite{9}. Similar to the low-level approach, the map image is fed through a model to generate a feature vector, which is used in a system of neural networks for trajectory prediction.

While systems often predict a single trajectory (mode), there is an advantage in predicting multiple modes and their associated probabilities, especially when there are multiple plausible trajectories that the vehicle might take. Several works \cite{1}, \cite{2}, \cite{4}, \cite{10}–\cite{13} use a multi-modal approach, predicting a probability distribution over trajectories for agents in the environment. This approach has been extended using multi-head attention \cite{2}, \cite{14}, allowing the model to focus on certain agents or other features of the scene context. In another approach \cite{13}, a multi-modal multi-task method jointly reasons about the future speed and steering of the vehicle, noting the joint relationship between the two. The Trajectron \cite{15} models multiple agents as dynamic graphs, and performs trajectory prediction for multi-modal, dynamic and variable multi-agent scenarios. SPAGNN \cite{16} addresses the behavior of other human drivers who make complex trade-offs while driving, modeling this relational behavior with graph neural networks.

Incorporating prior knowledge about the geometry and topology of roads into loss functions \cite{17} has been shown to result in more precise trajectory distributions over future outcomes. Rules of the road \cite{18} encodes high-level semantic information such as the entity state, other entities’ states and road networks into a spatial grid allowing deep convolutional networks to learn entity-entity and entity-environment interactions. ChauffeurNet \cite{19} introduces perturbations to trajectories and incorporates a loss for real-world driving mistakes, such as collisions and driving off-road. Our mid-level representation overlays multiple elements onto a single map for capturing the scene, losses, and driving goals.

Recent semi-supervised models extend and improve upon supervised ResNets by using orders of magnitude more unlabeled input samples \cite{5}, \cite{6}, \cite{20}, as well as output sample statistics \cite{21}, with good results in real-world applications.

**II. Methods**

We perform ablation studies comparing transfer learning of semi-supervised and supervised models on trajectory prediction tasks. We examine both low-level representations, which use the vehicle’s front-facing camera images as input, and mid-level representations, which use an annotated map image as input. In both cases, we use different semi-supervised models to encode the input, while keeping all other factors equal, including the system architecture and computational resources.

**A. Input Representation**

- **a) Mid-level representation:** Following state-of-the-art trajectory prediction models \cite{1}, \cite{2}, \cite{4}, \cite{9}, \cite{12}, we generate an annotated map image to represent the driving environment. This includes annotations for drivable areas, crosswalks and walkways using color coding to represent semantic categories. All scenes are oriented such that the agent under consideration is centered and directed towards the top of the image. The positions of all agents in the scene are drawn onto the image, using faded bounding boxes to represent past positions in a historical window. By encoding all this information into a single map, a large amount of information is condensed into a single image. The top row of Figure \cite{1} shows an annotated map of a scene in the nuScenes dataset \cite{22}. In addition to the map, a vector of the target agent’s state at the moment of prediction is also included as input. This includes the agent’s speed (between 0 and 30 m/s), acceleration (between -25 and 25 m/s²) and yaw rate (between $-2\pi$ and $2\pi$ radians/s).

- **b) Low-level representation:** We use front-facing camera images from the Drive360 dataset \cite{23} as a low-level representation of a driving environment. In addition to the image,
we include a vector of semantic map data, which includes datapoints such as the distance to the nearest intersection, the speed limit, and the approximate road curvature. The bottom row of Figure 1 show an example image from a front-facing camera in the Drive360 dataset.

B. Semi-Supervised Models

State-of-the-art models [1], [2] use transfer learning of supervised models, whereas we evaluate the use of semi-supervised models. We perform transfer learning by fine-tuning each semi-supervised model on our training set, lever-aging models already trained on up to a Billion images, orders of magnitude larger than the nuScenes dataset [22] which consists of 1.4 Million images. We provide a summary of the semi-supervised models we use in Table I. Next, we describe each semi-supervised model in detail.

a) Teacher-student self-training: We use ResNeXt-101 32x4d SSL and SWSL [6]. ResNeXt-101 32x4d SSL is trained on a semi-supervised task using a teacher-student method on an unlabeled dataset of 90M images, and fine-tuned on 1.2M images from the ImageNet1k dataset. ResNeXt-101 32x4d SWSL is trained using a teacher-student method on 940M images, leveraging associated hashtags in a semi-weakly supervised approach, and fine-tuned on the ImageNet1k dataset. Both of these models use the ResNeXt-101 32x4d architecture from [24].

b) Contrastive learning: We use SimCLR [5], trained using a contrastive learning method on ImageNet1k. During training, augmented versions of images are passed through a ResNet architecture [25]. The contrastive loss objective serves...
to minimize the distance between different augmentations of the same image, and maximize the distance between representations of other images. We use a ResNet-50 architecture trained with the SimCLR method.

C. Experiments

We experiment with using transfer learning from semi-supervised models in place of supervised models for both low-level and mid-level representations of the input.

a) Mid-level representation: For mid-level representations, we train our models to predict a 6-second trajectory for an agent, using 2 seconds of historical observations of the scene represented as an annotated map image. We use two architectures that have been shown to be successful on this task: Multiple-Trajectory Prediction (MTP) [4] and CoverNet [1], which we describe in more detail below. For each architecture, we substitute different semi-supervised models for the backbone used to encode the map image.

b) Low-level representation: For low-level representations, we test our approach by training models on the Drive360 dataset [23] used in the ICCV 2019: Learning-to-Drive Challenge. The task is to predict the speed and steering wheel angle of a human driver one second in the future after the observation. We experiment with different semi-supervised and supervised models to encode the front-facing camera footage, analogous to our experiments with the input map image of the mid-level representation. We use the architecture of the winning team of the competition [3] which uses a supervised image encoder and has been shown to be an effective end-to-end model.

D. Datasets

a) nuScenes: For our experiments with mid-level representations, we use nuScenes [22], a public large-scale dataset which consists of 1000 driving scenes in Boston and Singapore. Each scene is 20 seconds in length and is sampled at a frequency of 2Hz. We use the official data partitions from the nuScenes dev-kit implementation for all our experiments. We evaluate these scenarios using a dynamic and hybrid version of this trajectory set closest to the ground truth. In our experiments, we hold constant the configuration of the architecture during all experiments, and vary the ResNet component used to encode the images with different semi-supervised and supervised models.

b) Drive360: For our experiments with low-level representations, we use the Drive360 dataset [23]. The dataset includes 55 hours of driving recorded in Switzerland, divided into 27 routes and 682 chapters. We partition the data into disjoint datasets for training (43%), validation (43%), and test (14%). The dataset contains observations at a frequency of 10Hz, including GoPro images positioned around the car, of which we only use the front-facing camera, and map features in the form of a vector with 20 semantic datapoints such as the distance to the nearest intersection, the current speed limit, and the road curvature.

E. Architectures

We perform our experiments with low-level representations on the winning architectures of the ICCV 2019: Learning-to-Drive challenge [3], as shown in Figure 2 trained on the Drive360 dataset. The architecture for mid-level representations is shown in Figure 3. We use two networks that are successful on the nuScenes dataset: (i) Multiple-Trajectory Prediction (MTP) [4] which predicts a small number of trajectories; and (ii) CoverNet [1] which assigns probabilities to a large set of trajectories. In all cases, we hold constant the configuration of the architecture during all experiments, and vary the ResNet component used to encode the images with different semi-supervised and supervised models.

a) Mid-level representation: MTP [1], [4] uses the annotated map image and the target agent’s current state to predict a fixed number of trajectories, as well as their associated probabilities. The map image is passed through the “backbone” vision component, which is the model that we vary in our experiments. This representation and a vector of the agent’s state are passed through a fully-connected neural network used for fusing the different inputs. The output is a set of trajectories \( K \), and a vector of logits corresponding to their probabilities. The loss is calculated as a sum of the classification loss \( L_C \), which is a cross-entropy with the positive sample determined by the element in the trajectory set closest to the ground truth, referred to as the “best matching” mode, and a regression loss \( L_R \) for the best matching mode and the ground truth. In our experiments, we fix the number of output trajectories to 3. This matches one of the configurations evaluated in [1].

CoverNet [1] performs trajectory prediction by computing the probability distribution over a set of candidate trajectories. Similar to MTP, the model uses the annotated map image and a vector representing the target agent’s state as input. However, rather than predicting an entire set of trajectories \( K \) and their associated probabilities, the model only outputs probabilities for a fixed trajectory set \( K \). Although the original paper evaluates these scenarios using a dynamic and hybrid version of this trajectory set, we use the fixed version provided in the nuScenes dev-kit implementation for all our experiments. The loss function is only the classification loss \( L_C \) of the closest trajectory to the ground truth. In our experiments, we use the set of 415 trajectories. We show a visualization of this trajectory set in the top-right of Figure 1.

b) Low-level representation: The architecture for the low-level representation is depicted in Figure 2. The model consists of neural networks (in blue) and intermediate feature vectors (in orange). We replace the ResNet with different semi-supervised and supervised models in our experiments. Images are fed into the ResNet model and the vector of semantic map data is passed through an encoder. These are then fused together using a fusion layer to capture the non-linear interactions between the data sources. An LSTM then combines observations from the current timestep and a recent timestep (400ms in the past). This output is then fused together with data from the initial timestep and passed through regres-
sors to obtain the vehicle speed and steering angle prediction which are shown in green. The overall loss is the sum of the regression losses for the two targets.

III. RESULTS

a) Mid-level representation: We perform experiments showing the performance of transfer learning from semi-supervised models for encoding annotated maps in a 6-second trajectory prediction task. We use two architectures: CoverNet and MTP. For each semi-supervised model, we compare against a supervised model trained on ImageNet with the same architecture and number of layers. We compare CoverNet and MTP models by a standard set of metrics for multi-modal trajectory prediction: minADE$_1$, minADE$_5$, minADE$_{10}$, FDE and HitRate$_{5,2m}$.

The minimum Average Displacement Error (minADE$_k$) is the minimum displacement of the $k$ most likely trajectories from the ground truth, averaged along corresponding points of the ground truth and predicted trajectories. The HitRate$_{k,d}$ is the average number of trajectory sets in which this minimum, maximised along corresponding points of the ground truth and predicted trajectories, is below a threshold $d$. The final displacement error (FDE) is the error between the final predicted point and ground truth trajectory position, for the most likely trajectory. minADE$_5$, minADE$_{10}$ and HitRate$_{5,2m}$ take into consideration multiple modes while the other metrics are uni-modal.

As shown in Table II using semi-supervised models instead of supervised models shows significant improvement on most metrics when all other factors are held equal. Semi-supervised models result in minADE$_1$ improvements ranging from 8.6% to 40.1%, minADE$_5$ improvements ranging up to 16.8% and minADE$_{10}$ improvements as high as 15.5% across CoverNet and MTP. The improvement in FDE from semi-supervised models ranges from 9.1% to 35.3%. The improvement in HitRate$_{5,2m}$ is as high as 33% when SimCLR replaces ResNet-50 in the CoverNet architecture.

In Figure 4 we show the 2-meter HitRate metric as we increase $k$, the number of most probable trajectories included in the metric, for our experiments with CoverNet. It is clear that even over a wide range of $k$, the semi-supervised models outperform the supervised models, with SimCLR performing the strongest. We note that the supervised ResNet-50 model is a popular backbone model used in several implementations of CoverNet [1], [2], and our SimCLR model shows a clear improvement over this on all metrics without increasing the number of layers or inference time. Of all the semi-supervised methods, SimCLR, trained with constrastive learning, outperforms ResNeXt-101 SSL and SWSL, both trained with noisy Student methods.

We notice that while semi-supervised models perform better than supervised models across all metrics on CoverNet, this is not the case for MTP. For MTP, semi-supervised models improve performance significantly on the uni-modal metrics, however they perform only incrementally better or worse than supervised models on the multimodal metrics. This can be attributed to the fact that MTP predicts a small set of modes (3), as opposed to CoverNet which assigns probabilities to a much larger set of modes (415).

Examples of mid-level representations from the nuScenes dataset with their corresponding trajectory predictions using CoverNet are illustrated in Figure 5 and predictions using MTP are shown in Figure 6.

b) Low-level representation: The results of our experiments on low-level representations using the Drive360 dataset are shown in Table III. For the ICCV 2019: Learning-to-Drive winning architecture (L2D), which predicts the speed and steering for a timestep one second in the future, we report mean squared error (MSE) for both targets. SimCLR performs the best on the overall dataset, having the lowest MSE for both

| Model               | Size  | Type          | Label Ratio | Parameters |
|---------------------|-------|---------------|-------------|------------|
| ResNeXt-101 32x4d SWSL | 940M  | Teacher-student | 1:780       | 42M        |
| ResNeXt-101 32x4d SSL  | 90M   | Teacher-student | 1:75        | 42M        |
| SimCLR ResNet-50    | N/A   | Contrastive learning | N/A         | 25.6M      |

TABLE I
Comparison of semi-supervised models used in our experiments. The labeled dataset in all the models consists of 1.2M images. Since SimCLR is trained on augmentations, there is no measure of unlabeled data set size.
TABLE II
RESULTS OF COVERNET AND MTP ON THE nuSCENES DATASET, COMPARING DIFFERENT SEMI-SUPERVISED AND SUPERVISED MODELS TO ENCODE THE ANNOTATED MAP. FOR EACH SEMI-SUPERVISED MODEL, WE MAKE A DIRECT COMPARISON TO A SUPERVISED MODEL WITH THE SAME ARCHITECTURE. SEMI-SUPERVISED MODELS SIGNIFICANTLY OUTPERFORM THEIR SUPERVISED COUNTERPARTS ON MOST METRICS.

| Model                              | minADE1 | minADE5 | minADE10 | FDE  | HitRate10m |
|------------------------------------|---------|---------|----------|------|------------|
| Constant velocity                  | 5.48    | 5.48    | 5.48     | 13.44| 0.05       |
| Physics oracle                     | 3.91    | 3.91    | 3.91     | 9.53 | 0.10       |
| CoverNet on nuScenes               |         |         |          |      |            |
| ResNet-50 supervised               | 9.23    | 3.03    | 2.20     | 18.48| 0.12       |
| SimCLR ResNet-50                   | 5.53    | 2.52    | 1.86     | 11.95| 0.16       |
| ResNeXt-101 32x4d sup.             | 9.28    | 2.95    | 2.10     | 18.75| 0.14       |
| ResNeXt-101 32x4d SSL              | 7.03    | 2.67    | 1.99     | 14.67| 0.14       |
| ResNeXt-101 32x4d SWSL             | 7.43    | 2.65    | 1.99     | 16.64| 0.14       |
| MTP on nuScenes                    |         |         |          |      |            |
| ResNet-50 supervised               | 5.13    | 2.97    | 2.97     | 11.71| 0.14       |
| SimCLR ResNet-50                   | 4.69    | 3.13    | 3.13     | 10.65| 0.11       |
| ResNeXt-101 32x4d sup.             | 6.26    | 2.98    | 2.98     | 13.93| 0.13       |
| ResNeXt-101 32x4d SSL              | 6.02    | 3.06    | 3.06     | 13.50| 0.13       |
| ResNeXt-101 32x4d SWSL             | 5.18    | 2.96    | 2.96     | 11.63| 0.15       |

Fig. 5. Trajectory prediction examples on the nuScenes dataset using CoverNet. The mid-level representation of the annotated map is on the left. The colored lines on the right represent the ground truth trajectory (blue) and those predicted by CoverNet with various backbone models, including the supervised ResNet-50 (orange) and the semi-supervised models we evaluate. The gray lines in the background are the set of 415 trajectories in the fixed trajectory set.

Fig. 6. Trajectory prediction examples on the nuScenes dataset using MTP. The mid-level representation of the annotated map is on the left. The colored lines on the right represent the ground truth trajectory (blue) and those predicted by MTP with various backbone models, including the supervised ResNet-50 (orange) and the semi-supervised models we evaluate. In the third example, we show a dense traffic scenario.
TABLE III
COMPARISON OF SPEED AND STEERING WHEEL ANGLE PREDICTION ON THE DRIVE360 TEST DATASET FOR THE SEMI-SUPERVISED AND SUPERVISED MODELS WE EVALUATE. FOR BOTH SPEED AND STEERING WHEEL ANGLE PREDICTION, THE SEMI-SUPERVISED SIMCLR MODEL IMPROVES UPON THE SUPERVISED MODELS. STEERING ANGLE MSE IS MEASURED IN DEGREES\(^2\) AND THE SPEED MSE IN (KM/H)\(^2\).

| Model                        | Angle MSE | Speed MSE |
|------------------------------|-----------|-----------|
| L2D winner on Drive360      | 1013.46   | 10.40     |
| ResNet-50 supervised        | 1003.56   | 9.53      |
| SimCLR ResNet-50            | 1010.64   | 10.43     |
| ResNet-101 supervised       | 1050.58   | 10.80     |
| ResNeXt-101 32x4d SSL       | 1103.13   | 9.69      |

speed and steering wheel angle, outperforming the supervised models. This reiterates the findings from our experiments on mid-level representations where SimCLR, trained with contrastive learning, outperforms the other models in most cases. We however do not observe improvements when using the semi-supervised ResNext-101 32x4d SSL and ResNext-101 32x4d SWSL models, trained with noisy-student methods, as compared to the supervised ResNet-101 on this task. Examples of low-level inputs from the Drive360 dataset and their corresponding steering wheel angle predictions are shown in Figure 7.

c) Implementation Details: Training is performed on a Google Cloud Platform instance with an NVIDIA Tesla T4 or P100 GPU. For the mid-level representations, we downsample the nuScenes training data by a ratio of 5:1 during training, which reduces training time to 10-20 hours per model. For the low-level representations, we downsample the Drive360 dataset by a ratio of 10:1 during training to reduce the number of training instances, and we additionally downsample the input images from from 1920x1080 to 160x90 pixels. This reduces training to about 5-10 hours per model. For all models, we report results on the complete test split without downsampling. During training, we freeze \(\frac{3}{4}\) of the lowest blocks of the semi-supervised and supervised models, fine-tuning the remaining blocks.

IV. Conclusion
We demonstrate the benefits of using transfer learning of semi-supervised models on real-world driving benchmarks. By performing an ablation study comparing transfer learning of semi-supervised models with supervised models while keeping all other factors equal, we show that using semi-supervised models improves performance for both low-level and mid-level representations. Within semi-supervised models, we compare: (i) contrastive learning with teacher-student methods; and (ii) networks predicting a small number of trajectories with networks predicting the probabilities over a large set of trajectories. Using semi-supervised models in place of supervised models requires no additional computational resources when performing transfer learning or inference, hence our results present a simple recipe for significantly improving trajectory prediction.

REFERENCES
[1] Phan-Minh, T., Grigore, E.C., Boulton, F.A., Beijbom, O., Wolff, E.M.: CoverNet: Multimodal behavior prediction using trajectory sets. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 14074–14083
[2] Messaoud, K., Deo, N., Trivedi, M.M., Nashashibi, F.: Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. arXiv (2020)
[3] Diodato, M., Li, Y., Goyal, M., Drotri, I.: Winning the ICCV 2019 Learning to Drive Challenge. ICCV Autonomous Driving Workshop (2019)
[4] Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: IEEE International Conference on Robotics and Automation (ICRA). (2019) 2090–2096
[5] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of International Conference on Machine Learning. (2020)
[6] Yalniz, I.Z., Jégou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)
[7] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)

[8] Fernando, T., Dennan, S., Sridharan, S., Fookes, C.: Going deeper: Autonomous steering with neural memory networks. In: Proceedings of IEEE International Conference on Computer Vision Workshops. (2017) 214–221

[9] Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F., Lin, T., Singh, N., Schneider, J.: Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In: IEEE Winter Conference on Applications of Computer Vision (WACV), (2020) 2084–2093

[10] Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2018) 1468–1476

[11] Tang, C., Salakhutdinov, R.R.: Multiple futures prediction. In: Proceedings of Advances in Neural Information Processing Systems. (2019) 15398–15408

[12] Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In: 3rd Conference on Robot Learning (CoRL). (2019)

[13] Yang, Z., Zhang, Y., Yu, J., Cai, J., Luo, J.: End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions. In: International Conference on Pattern Recognition. (2018) 2289–2294

[14] Kim, H., Kim, D., Kim, G., Cho, J., Huh, K.: Multi-head attention-based probabilistic vehicle trajectory prediction. arXiv preprint arXiv:2004.03842 (2020)

[15] Ivanovic, B., Pavone, M.: The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of IEEE/CVF International Conference on Computer Vision. (2019) 2375–2384

[16] Casas, S., Gulino, C., Liao, R., Urtasun, R.: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. IEEE International Conference on Robotics and Automation (2020)

[17] Casas, S., Gulino, C., Suo, S., Urtasun, R.: The importance of prior knowledge in precise multimodal prediction. arXiv preprint arXiv:2006.02636 (2020)

[18] Hong, J., Sapp, B., Philbin, J.: Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 8454–8462

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

[20] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 9729–9738

[21] Lamm, N., Srikanth, M., Jaiprakash, S., Drori, I.: Trajectograms: Which semi-supervised trajectory prediction model to use? In: ICML Workshop on AI for Autonomous Driving. (2020)

[22] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multi-modal dataset for autonomous driving. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 11621–11631

[23] Hecker, S., Dai, D., Van Gool, L.: End-to-end learning of driving models with surround-view cameras and route planners. In: Proceedings of European Conference on Computer Vision. (2018) 435–453

[24] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2017) 1492–1500

[25] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2016) 770–778