Trajectograms: Which Semi-Supervised Trajectory Prediction Model to Use?

Nick Lamm [1] **Malavika Srikanth** [1] **Shashank Jaiprakash** [1] **Iddo Drori** [1 2]

Abstract

Supervised models learn a mapping from an input to an output space from example pairs. Semi-supervised models leverage the availability of unlabeled inputs, using data on the input space manifold. We propose a method for extending semi-supervised learning to the output space for vehicle trajectory prediction. We show that a meta-model which adaptively selects a semi-supervised model based on recent driving trajectograms, or trajectory histograms, improves trajectory prediction by applying a suitable semi-supervised model to the given driving scenario. Our meta-model improves trajectory prediction accuracy compared with the best supervised model as well as state of the art semi-supervised input models, demonstrating that semi-supervised data of both input and output spaces are a useful signal for trajectory prediction.

1. Introduction

The traditional supervised approach to vehicle trajectory prediction involves mapping observations from the input space, such as sensor data from the vehicle, to the output space of trajectories, mapping $\mathcal{X} \mapsto \mathcal{Y}$. Semi-supervised approaches improve performance by also using unlabeled data from the input space $\mathcal{X}$, such as unlabeled images available in large quantities, as illustrated in Figure 1. In this work, we extend semi-supervised learning for trajectory prediction to the output space. We show that data from the output space alone, $\mathcal{Y}$, namely vehicle trajectories, which are also available in large quantities, are useful for trajectory prediction.

We demonstrate our method by training a meta-model that takes as input a trajectogram, which is a trajectory histogram, as depicted in Figure 2. The meta-model uses the trajectogram observed from recent driving to predict which out of a set of trained semi-supervised trajectory models to employ in a given driving scenario. The semi-supervised models are fine-tuned on the trajectory prediction task to learn mappings $\mathcal{M}_i : \mathcal{X} \mapsto \mathcal{Y}$. Our meta-model learns a mapping from the output space to the semi-supervised model: $\mathcal{Y} \mapsto \mathcal{M}_i$, which is then used for trajectory prediction.

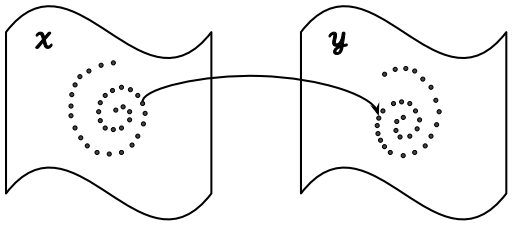

Figure 1. Supervised approaches learn a mapping $\mathcal{X} \mapsto \mathcal{Y}$ between the input space sensor data to output space driving trajectories from example pairs $\{\mathcal{X}_i, \mathcal{Y}_i\}$ illustrated by points. Semi-supervised approaches use unlabeled input data $\{\mathcal{X}_j\}$, in addition to the supervised pairs. Input samples are often abundant and knowing the underlying manifold of the input space improves performance. In this work we demonstrate that using unlabeled output space samples $\{\mathcal{Y}_k\}$, in addition to supervised pairs and unlabeled input space samples, further improves prediction accuracy.

We test our approach by training models on the Drive360 dataset (Hecker et al., 2018) used in the ICCV 2019: Learning-to-Drive Challenge. The dataset includes camera footage, visual maps, and semantic maps, with the task of predicting the speed and steering wheel angle of a human driver one second into the future after the given observations. We use the supervised architecture of the winning team of the competition (Diodato et al., 2019) for comparison. We show that our meta-model improves the steering prediction accuracy by 3.7% and speed by 5.95% compared with supervised models, and by 3% and 0.5% compared with semi-supervised models. Figure 3 shows an example of predictions made by individual models and the meta-model.

1.1. Related Work

End-to-end models have been used to predict steering commands using low-level representations like raw pixels from a front camera (Bojarski et al., 2016). Other systems construct

*Equal contribution [1]Department of Computer Science, Columbia University, New York, USA [2]School of Operations Research and Information Engineering, Cornell University, New York, USA. Correspondence to: Nick Lamm <nl2680@columbia.edu>.

Workshop on AI for Autonomous Driving (AIAD), the 37^{th} International Conference on Machine Learning, Vienna, Austria, 2020.

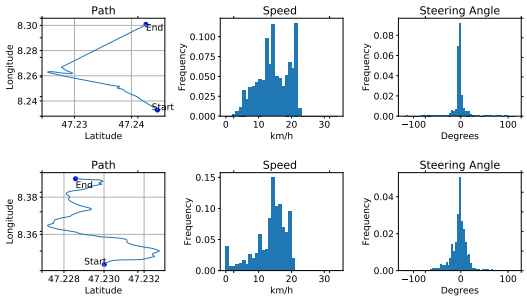

Figure 2. Trajectograms from different 5-minute segments of the same drive. Notice that the statistics, represented by the speed and steering wheel angle histograms, are different for the driving scenarios in the top and bottom rows.

mid-level representations of the environment, such as a map annotated with agent positions (Djuric et al., 2020), rather than predicting a vehicle's trajectory directly from visual input. These representations are then used as input to predict trajectories. We follow a low-level, end-to-end approach, using front-facing camera footage as input to predict speed and steering wheel angle.

Fernando et al. (2017) shows the power of neural memory networks to incorporate long-term dependencies critical to self-driving. In our work, rather than using an LSTM to represent long-term dependencies, we construct a compact representation with trajectograms.

Whereas our model predicts future driving actions, other approaches (Deo & Trivedi, 2018; Cui et al., 2019; Tang & Salakhutdinov, 2019; Chai et al., 2019) predict a probabilistic distribution of output trajectories for agents in the environment. This approach is extended using multi-head attention (Kim et al., 2020; Messaoud et al., 2020) to focus on trajectories of certain agents more than others. CoverNet (Phan-Minh et al., 2020) performs trajectory prediction by building diverse trajectory sets, imposing dynamic constraints on feasible trajectories, and solving a classification problem over these trajectory sets. Representing the behavior of other agents with graphs has been explored by the Trajectron (Ivanovic & Pavone, 2019) and SPAGNN (Casas et al., 2020a). A multi-modal multi-task approach to jointly reason between speed and steering predictions (Yang et al., 2018) is similar to the semi-supervised baseline model that we propose. ChauffeurNet (Bansal et al., 2019) uses imitation learning to learn driving trajectories and introduces trajectory perturbations to improve robustness. Incorporating prior knowledge into loss functions (Casas et al., 2020b) results in more precise trajectory distributions over future outcomes. Rules of the road (Hong et al., 2019) 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.

Figure 3. Steering wheel predictions during three of the recorded drives. In these examples, each of the individual models predict steering angles similar to the ground truth, shown in the top-left wheel in each example. The meta-model, using the trajectogram from recent driving, selects which of the semi-supervised models will have the lowest error. The meta-model's selections are shown in the dotted green line. In these examples, the meta-model correctly selects the best model in each scenario. By choosing the best model at each scenario throughout a drive, the meta-model's trajectories have lower error than any individual model.

Recent semi-supervised models extend and improve upon supervised ResNets by using orders of magnitude more unlabeled input samples (Yalniz et al., 2019; He et al., 2020; Chen et al., 2020; Xie et al., 2020) with good results in real-world applications.

2. Methods

Next, we describe our system of neural networks as illustrated in Figure 4 for predicting steering angle and speed. The input data (in green at the bottom) consists of images captured by the vehicle's front-facing camera and semantic map data at 100 millisecond intervals. The Drive360 dataset includes 55 hours of driving recorded in Switzerland, divided into 27 routes and 682 chapters. We partition the data into three disjoint datasets, for training the semi-supervised models (43%), training the meta-model (43%), and validating the meta-model (14%). The architecture consists of neural networks (in blue) and intermediate feature vectors

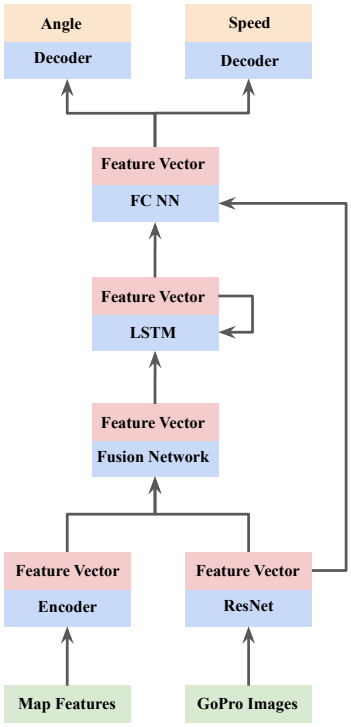

Figure 4. System architecture: Inputs are shown in green, neural networks in blue, intermediate feature vectors in red, and outputs in orange. The ResNet component is one of the supervised or semi-supervised models we evaluate.

(in red). We replace the ResNet with different supervised or semi-supervised models. This allows for a comparison of models while keeping all other factors equal. Images are fed into the supervised or semi-supervised model and the semantic map data is passed through an encoder, a fully-connected network. A fusion layer captures the non-linear interactions between data sources, passing a feature vector to an LSTM, which combines data from the current timestep and a recent timestep (400ms in the past). The LSTM output is then fused together with data from the initial timestep and passed to the regressors. We predict two outputs (in orange at the top), steering angle and the vehicle speed.

2.1. Semi-Supervised Models

We evaluate and compare different semi-supervised models in the place of the ResNet component in the architecture. We perform transfer learning by fine-tuning each semi-supervised model on our training set, allowing us to leverage models already trained on datasets orders of magnitude larger than our own. Next, we describe each semi-supervised base model:

Teacher-student self-training. We use ResNeXt-101 32x4d SSL and SWSL (Yalniz et al., 2019). 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.

Contrastive learning. We use SimCLR (Chen et al., 2020), trained using a contrastive learning method on ImageNet1k with a ResNet-50 architecture.

Noisy student training. We use a noisy student model (Xie et al., 2020), trained using a modification of the teacher-student method in which the student model is noised. The teacher is initially trained on the ImageNet1k dataset, and then used to generate pseudo labels for the student model, which in turn is trained on 300M unlabeled images sourced from the JFT dataset.

2.2. Meta Model

While the semi-supervised base models leverage unlabeled data from the input space $\mathcal{X}$ to improve the mapping from input to output space, $\mathcal{X} \mapsto \mathcal{Y}$, our meta-model learns a mapping from the output space to a semi-supervised model: $\mathcal{Y} \mapsto \mathcal{M}_i$. Our meta-model uses trajectograms from the output space to predict which semi-supervised model $\mathcal{M}_i$ to use in a driving scenario. Trajectograms represent the characteristics of the current segment of road, as illustrated in Figure 2 where we compare the trajectograms of a mostly straight (top row) and a highly curved (bottom row) driving segment. Our meta-model uses these output space samples without corresponding input labels, reversing the roles of input and output applied in a traditional semi-supervised model. In both cases, such unlabeled data may be easy to obtain: (i) semi-supervised input model data: Billions of unlabeled images; (ii) semi-supervised output model data: Unlabeled trajectories, without their corresponding input images, for example, from a bird's eye view. Obviously, there is mutual information between the input space $\mathcal{X}$ and output space $\mathcal{Y}$, since a driving environment strongly determines the possible trajectories that a vehicle may take. A trajectogram therefore not only represents recent driving, but also indirectly represents the recent driving environment. An advantage of using a trajectogram to represent the driving environment is that we efficiently capture information from a long time period, such as a 5 minute window of driving, without fully processing input signals over this period. We efficiently update the histograms incrementally by a moving window. In practice, this also allows us to easily incorporate new base models: we provide a standard driving dataset to all base models, and then only need their predictions on this dataset to train a meta-model on these base models.

Table 1. Comparison of speed and steering wheel angle prediction on a test dataset for each of the individual supervised and semi-supervised models, and our adaptively selected semi-supervised model chosen by a network trained on trajectograms. For both speed and steering wheel angle prediction, our meta-model improves upon the semi-supervised models, which in turn improve upon the supervised models. Steering angle MSE is measured in degrees^2 and the speed MSE in $(\text{km/h})^2$.

MODEL	TYPE	ANGLE	SPEED
RESNET-101	SUPERVISED	1010.64	10.43
RESNET-50	SUPERVISED	1013.46	10.40
RESNET-34	SUPERVISED	1067.36	10.08
SIMCLR RESNET-50	SEMI-SUPERVISED	1003.56	9.53
RESNEXT-101 32X4D SSL	SEMI-SUPERVISED	1050.58	10.80
RESNEXT-101 32X4D SWSL	SEMI-SUPERVISED	1103.13	9.69
NOISY STUDENT EFFICIENTNET B7	SEMI-SUPERVISED	1213.00	13.17
OURS	ADAPTIVE SEMI-SUPERVISED	**973.30**	**9.48**
SEMI-SUPERVISED IMPROVEMENT		0.70%	5.46%
ADAPTIVE SEMI-SUPERVISED IMPROVEMENT		3.02 %	0.52%
OVERALL IMPROVEMENT		**3.69%**	**5.95%**

3. Results

We experiment with different types of models for speed and steering wheel angle prediction based on the Learning-to-Drive dataset and setup. Our meta-model is trained on a classification task with cross-entropy loss over the semi-supervised models, where the target is the model with the lowest error on the task and the input is the trajectograms from recent driving. We train a separate meta-model for speed and for steering wheel angle to select the best performing model on each task. Table 1 shows that our meta-model improves steering angle prediction accuracy by 3.7% over the best supervised model and by 5.95% for speed prediction. Our meta-model improves upon the semi-supervised models for trajectory prediction, which in turn improve upon the supervised models for both tasks.

The meta-model improves on the best semi-supervised model since no individual semi-supervised model is best across all driving scenarios. Although SimCLR has the lowest overall error for an individual model, it is the best choice for steering angle in only 38% of the examples used to train the meta-model, while the overall lowest performing steering angle model, Noisy Student EfficientNet-B7, is the best choice in 10% of the examples. This emphasizes the advantage of adaptively selecting from a set of semi-supervised models based on the driving conditions.

Implementation. Neural network training is performed on a Google cloud instance running an NVIDIA Tesla T4 GPU, and takes between 5-10 hours per model. We freeze $\frac{3}{4}$ of the lowest blocks of the semi-supervised and supervised models, fine-tuning the remaining blocks. We downsample the images from 1920x1080 to 160x90 pixels, and during training we downsample the dataset over time at a ratio of 1:10. For the meta-model, we use a separate feed-forward network for the steering and speed tasks, allowing a different semi-supervised model to be used for each task in a driving scenario. We find that a longer trajectogram window (4 minutes) provides a useful signal for the steering angle task, while a shorter window (2 minutes) is more useful for the speed task. This indicates that understanding the characteristics of a longer driving segment is more indicative of the steering ahead, while speed prediction is concerned with more local observations.

Discussion. Our experiments show that training a meta-model on observed trajectograms improves end-to-end trajectory prediction. Although our available dataset is only 55 hours of driving, we are able to show that data from the output space is a useful signal which improves trajectory prediction. We believe that this is a promising concept that is worth scaling up to more driving data. Our approach of using trajectograms to predict the semi-supervised model with the most accurate trajectory is just one example of how learning from unlabeled data in the output space may improve trajectory prediction. Through further exploration we may find different ways to represent the output space other than trajectograms.

There are a number of practical advantages to applying our method at scale. For instance, it is simple to incorporate a new base model without any knowledge of how that model is trained. Establishing a common driving dataset and distributing each model's predictions on that dataset allows us to train the meta-model to predict which model to use in a driving scenario in an adaptive fashion. In a production setting, we may include additional constraints on the meta-model, such as avoiding trajectories that violate driving restrictions. Our experiments are a step towards an input-output semi-supervised model, and we demonstrate its applicability to vehicle trajectory prediction.

Acknowledgements

We would like to thank Michael Diodato and Yu Li of Columbia University for the baseline supervised model that won the ICCV Learning to Drive challenge.

References

Bansal, M., Krizhevsky, A., and Ogale, A. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science and Systems*, 2019.

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.

Casas, S., Gulino, C., Liao, R., and Urtasun, R. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *International Conference on Robotics and Automation*, 2020a.

Casas, S., Gulino, C., Suo, S., and Urtasun, R. The importance of prior knowledge in precise multimodal prediction. *arXiv preprint arXiv:2006.02636*, 2020b.

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

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning*, 2020.

Cui, H., Radosavljevic, V., Chou, F.-C., Lin, T.-H., Nguyen, T., Huang, T.-K., Schneider, J., and Djuric, N. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *International Conference on Robotics and Automation*, pp. 2090–2096, 2019.

Deo, N. and Trivedi, M. M. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1468–1476, 2018.

Diodato, M., Li, Y., Goyal, M., and Drori, I. Winning the ICCV 2019 Learning to Drive Challenge. *ICCV Autonomous Driving Workshop*, 2019.

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

Fernando, T., Denman, S., Sridharan, S., and Fookes, C. Going deeper: Autonomous steering with neural memory networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 214–221, 2017.

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

Hecker, S., Dai, D., and Van Gool, L. End-to-end learning of driving models with surround-view cameras and route planners. In *Proceedings of the European Conference on Computer Vision*, pp. 435–453, 2018.

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

Ivanovic, B. and Pavone, M. The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2375–2384, 2019.

Kim, H., Kim, D., Kim, G., Cho, J., and Huh, K. Multi-head attention-based probabilistic vehicle trajectory prediction. *arXiv preprint arXiv:2004.03842*, 2020.

Messaoud, K., Deo, N., Trivedi, M. M., and Nashashibi, F. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. 2020.

Phan-Minh, T., Grigore, E. C., Boulton, F. A., Beijbom, O., and Wolff, E. M. CoverNet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14074–14083, 2020.

Tang, C. and Salakhutdinov, R. R. Multiple futures prediction. In *Advances in Neural Information Processing Systems*, pp. 15398–15408, 2019.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M., and Mahajan, D. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

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