Vehicle Trajectory Prediction by Transfer Learning of Semi-Supervised Models

Nick Lamm
Columbia University
nl2680@columbia.edu

Shashank Jaiprakash
Columbia University
sj3003@columbia.edu

Malavika Srikanth
Columbia University
ms5908@columbia.edu

Iddo Drori
Columbia University
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.

1 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 shown 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 rely on transfer learning with a model pre-trained on a supervised task [22, 24] 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 [11]. For a mid-level representation, we use CoverNet [24] and multiple trajectory prediction (MTP) [9], two multi-modal approaches that take an annotated map image as input. In all of these cases,

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

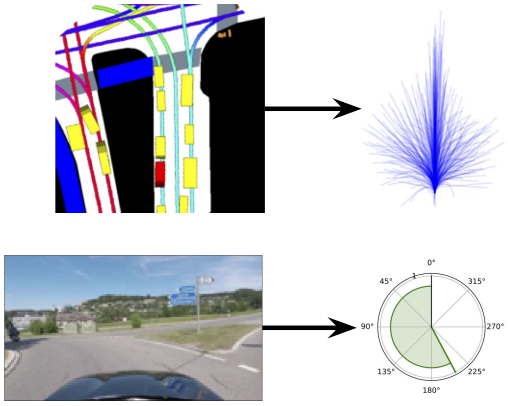


Figure 1: An example of input and output representations for mid-level (top) and low-level representations (bottom). In the top row, the mid-level input representation is an annotated map of the scene (top left), with boxes representing agent positions and colors representing semantic categories. The output (top right) is a probability distribution over a set of candidate trajectories. In the bottom row, a low-level representation uses an image from the vehicle's front-facing camera as input (bottom left), and predicts the future steering wheel angle (bottom right) and speed of the vehicle.

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 [4, 15, 27, 29]. 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 for trajectory prediction (described in Section 3):

1. Semi-supervised models significantly improve upon supervised models using both low-level and mid-level representations as described in Section 3 and shown in Tables 2 and 3.
2. Contrastive semi-supervised learning (SimCLR [8]) outperforms teacher-student semi-supervised learning (ResNeXt-101 32x4d SSL and SWSL [29]).
3. Using semi-supervised models for predicting probabilities of a large set of trajectories with CoverNet [24] 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 [9] results in significant performance improvement only on uni-modal metrics (up to 17.3%).

1.1 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 [2, 13]. 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 [11], 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 [7, 9, 12, 22, 24]. 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. 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 [7, 9, 10, 22, 24, 25, 30] 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 [20, 22], allowing the model to focus on certain agents or other features of the scene context. In another approach [30], 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 [19] models multiple agents as dynamic graphs, and performs trajectory prediction for multi-modal, dynamic and variable multi-agent scenarios. SPAGNN [5] 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 [6] has been shown to result in more precise trajectory distributions over future outcomes. Rules of the road [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 [1] 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 research has shown the advantages of semi-supervised learning, whereby a model learns from a set of unlabeled data in addition to labeled data. The key advantage is that the model can learn the underlying manifold of the input space using the unlabeled data, which are often abundant compared to the number of available labeled samples. In the computer vision domain, semi-supervised models have been shown to improve upon supervised ResNets by incorporating large unlabeled datasets of images [8, 15, 27, 29]. Self-supervised and semi-supervised models have been shown to perform well on transfer learning tasks compared to supervised models [8, 15, 29, 14, 23, 26]. In our work, we extend transfer learning of semi-supervised models to the domain of trajectory prediction. We describe the semi-supervised approaches used in our experiments in Section 2.2.

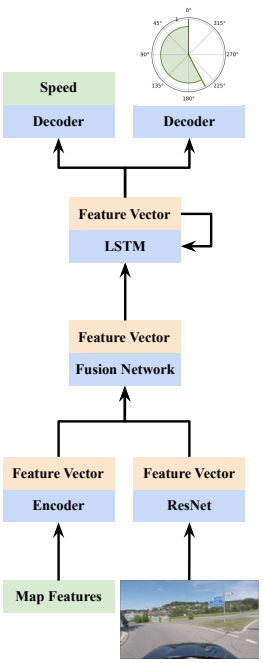
2 Methods

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.

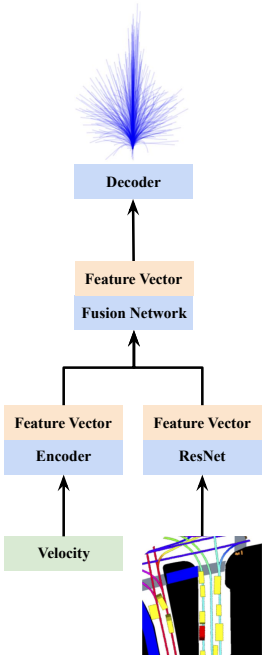
2.1 Input Representation

Mid-level representation. Following state-of-the-art trajectory prediction models [7, 9, 12, 22, 24], 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 1 shows an annotated map of a scene in the nuScenes dataset [3]. 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/s^2) and yaw rate (between -2π and 2π radians/s).

Low-level representation. We use front-facing camera images 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.



(a) Architecture for Low-level input representation.



(b) Architecture for Mid-level input representation.

Figure 2: System architectures for the (a) low-level and (b) mid-level input representations. Inputs are shown at the bottom, neural networks in blue, intermediate feature vectors in orange, and the model output at the top. (a) The network for low-level representations takes as input a vector of semantic map features and an image from the vehicle's front-facing camera. The inputs are encoded, and then fused together with a fully-connected network to capture non-linear interactions. An LSTM combines observations from multiple timesteps, and finally decoder networks predict the speed and the steering wheel angle of the driver. (b) The network of the mid-level representation takes as input the current state of the agent (a vector including velocity, acceleration, and yaw change rate), and an annotated map image of the environment. The map is fed into a ResNet backbone, and this representation is then concatenated with the agent state vector and passed to a fully-connected fusion network. For MTP, the final layer outputs a set of trajectories and an associated probability distribution. For CoverNet we use a fixed setting in which the output is a probability distribution over a set of candidate trajectories.

Table 1: 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 dataset size.

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

2.2 Semi-Supervised Models

State-of-the-art models [22, 24] 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, leveraging models already trained on up to a Billion images, orders of magnitude larger than the nuScenes dataset [3] which consists of 1.4 Million images. In all experiments, we use pretrained weights provided with the original model rather than performing the semi-supervised learning ourselves, which requires significant computational resources. We provide a summary of the semi-supervised models we use in Table 1. Next, we describe each semi-supervised model in detail.

Teacher-student self-training. We use ResNeXt-101 32x4d SSL and SWSL [29]. 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 [28].

Contrastive learning. We use SimCLR [8], trained using a contrastive learning method on ImageNet1k. During training, augmented versions of images are passed through a ResNet architecture [16]. 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.

2.3 Datasets

nuScenes: For our experiments with mid-level representations, we use nuScenes [3], 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 prediction challenge: 32,186 instances in the training set, 8,560 in the validation set, and 9,041 in the test set. Each instance is comprised of a scene at a particular point in time, with a particular agent of interest whose trajectory the model predicts. The dataset includes a high definition map of the scene, bounding boxes and past positions for all agents.

Drive360: For our experiments with low-level representations, we use the Drive360 dataset [17]. 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.

2.4 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.

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 networks that are successful on the nuScenes dataset: (i) Multiple-Trajectory Prediction (MTP) [9] which predicts a small number of trajectories; and (ii) CoverNet [24] which assigns probabilities to a large set of trajectories. The architecture for mid-level representations is shown in Figure 2b. 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.

MTP [9, 24] 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 $\mathcal{K}$, 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 [24].

CoverNet [24] 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 $\mathcal{K}$ and their associated probabilities, the model only outputs probabilities for a fixed trajectory set $\mathcal{K}$. 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.

Low-level representation. For low-level representations, we train models to predict the speed and steering wheel angle of a human driver one second in the future, using front-facing camera footage and a vector of semantic map features. We perform our experiments with the winning architectures of the ICCV 2019: Learning-to-Drive challenge [11], as shown in Figure 2a, trained on the Drive360 dataset [17]. 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. The architecture is depicted in Figure 2a. 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 regressors 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.

3 Results

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 additionally include SimCLR with the wider ResNet-50(4x) [8] architecture, one of the latest and best performing semi-supervised models on ImageNet benchmarks to date, to evaluate how improvements in semi-supervised pre-training contribute to our task.

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 $\text{HitRate}_{5,2m}$. 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 $\text{HitRate}_{k,d}$ [24] 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 $\text{HitRate}_{5,2m}$ take into consideration multiple modes while the other metrics are uni-modal.

As shown in Table 2, 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 5.8% to 33.9%, minADE_5 improvements up to 17.8% and minADE_{10} improvements as high as 15.5% across CoverNet and MTP. The improvement in FDE from semi-supervised models are as high as 28.8%. The improvement in $\text{HitRate}_{5,2m}$ is as high as 33% when SimCLR Resnet-50 replaces supervised ResNet-50 in the CoverNet architecture.

Table 2: 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. Additionally we experiment with SimCLR ResNet-50(4x), one of the latest and top performing semi-supervised models to see how improvements in semi-supervised pre-training contribute to our task. For minADE(mADE) and FDE, lower is better, and for HitRate(HR) higher is better.

Model	**Type**	**mADE$_1$**	**mADE$_5$**	**mADE$_{10}$**	**FDE**	**HR$_{5,2m}$**
Baselines						
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						
ResNet-50	Supervised	9.23	3.03	2.20	18.48	0.12
SimCLR ResNet-50	Semi-Supervised	6.10	**2.49**	1.86	13.16	0.16
SimCLR ResNet-50(4x)	Semi-Supervised	**5.53**	2.52	**1.86**	**11.95**	**0.16**
ResNeXt-101 32x4d	Supervised	9.28	2.95	2.10	18.75	0.14
ResNeXt-101 32x4d SSL	Semi-Supervised	7.03	2.67	1.99	14.67	0.14
ResNeXt-101 32x4d SWSL	Semi-Weakly Super.	7.43	2.65	1.99	16.64	0.14
MTP						
ResNet-50	Supervised	5.13	2.97	2.97	11.71	0.14
SimCLR ResNet-50	Semi-Supervised	4.83	3.04	3.04	11.11	0.14
SimCLR ResNet-50(4x)	Semi-Supervised	**4.69**	3.13	3.13	**10.65**	0.11
ResNeXt-101 32x4d	Supervised	6.26	2.98	2.98	13.93	0.13
ResNeXt-101 32x4d SSL	Semi-Supervised	6.02	3.06	3.06	13.50	0.13
ResNeXt-101 32x4d SWSL	Semi-Weakly Super.	5.18	**2.96**	**2.96**	11.63	**0.15**

It is notable that SimCLR ResNet-50(4x) outperforms all other semi-supervised models under consideration for the CoverNet architecture. SimCLR ResNet-50(4x) is relatively new and known to be one of the best performing semi-supervised models on the ImageNet dataset. This shows that improvements in semi-supervised pre-training can be leveraged to improve results in this domain through transfer learning.

In Figure 3, 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 [22, 24], 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). We note that there is not a single method that is the clear winner across both CoverNet and MTP on all evaluated metrics. This indicates that, in practice, it may be beneficial to select the semi-supervised model for a given task at hand [21].

Low-level representation. The results of our experiments on low-level representations using the Drive360 dataset are shown in Table 3. 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 speed and steering wheel angle, outperforming the supervised models. This reiterates the findings from our experiments on mid-level representations where SimCLR, trained with constrastive 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

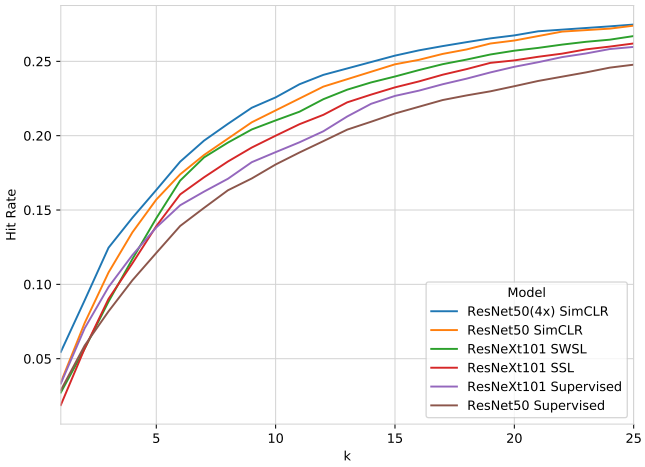


Figure 3: Hit Rate for CoverNet (415 modes, fixed) for each backbone model over the top k predicted trajectories as k is increased. Beginning around k=5, there is a clear separation between the different backbone models, with the semi-supervised models outperforming the supervised models. As k increases, the relative ordering of the models remains for the most part constant. This indicates that increasing the number of candidate trajectories considered in the Hit Rate metric has a consistent effect across all the models.

Table 3: 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 $(\text{km/h})^2$.

Model	**Type**	**Angle MSE**	**Speed MSE**
L2D winner on Drive360			
ResNet-50	Supervised	1013.46	10.40
SimCLR ResNet-50	Semi-Supervised	**1003.56**	**9.53**
ResNet-101	Supervised	1010.64	10.43
ResNeXt-101 32x4d SSL	Semi-Supervised	1050.58	10.80
ResNeXt-101 32x4d SWSL	Semi-Weakly Super.	1103.13	9.69

SWSL models, trained with noisy-student methods, as compared to the supervised ResNet-101 on this task.

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.

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

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] 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. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[4] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2959–2968, 2019.

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

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

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

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

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

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

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

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

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

[14] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020.

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

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

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

[18] J. Hong, B. Sapp, and J. Philbin. 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*, pages 8454–8462, 2019.

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

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

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

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

[23] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

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

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

[26] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

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

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

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

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

Appendix

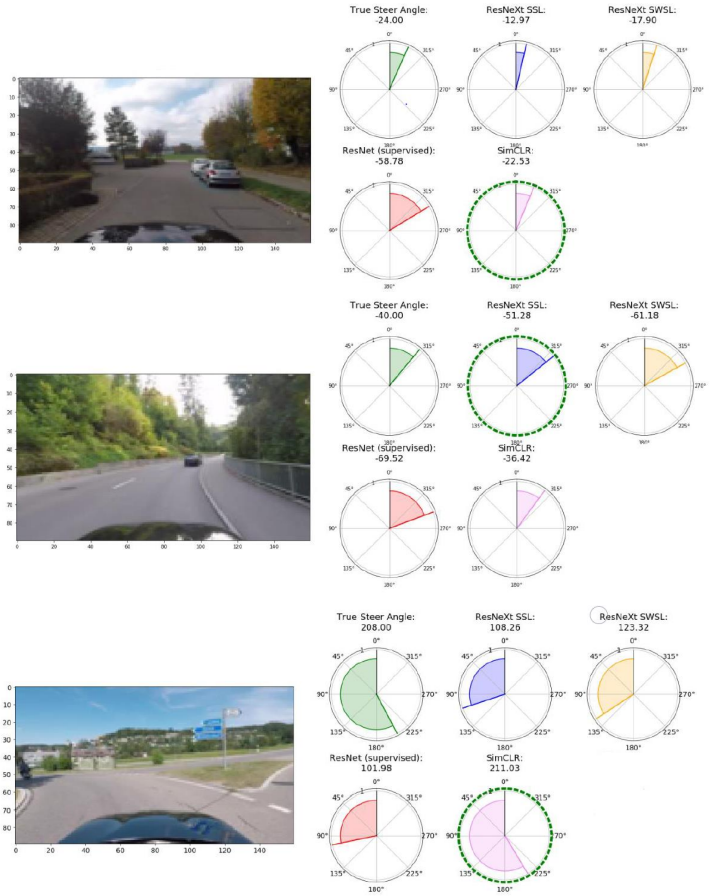


Figure 4: Steering wheel angle predictions from three examples in the Drive360 dataset. Using the low-level representation of the front-facing camera image (left), the model predicts the steering wheel angle 1 second in the future. To the right of each image, we show the ground truth in the top-right position, the prediction from the supervised ResNet-50 model in the bottom-left, and the predictions from the three semi-supervised models in the right positions. We highlight the most accurate model with a dashed green line. In all three examples, one of the semi-supervised models outperforms the supervised model.

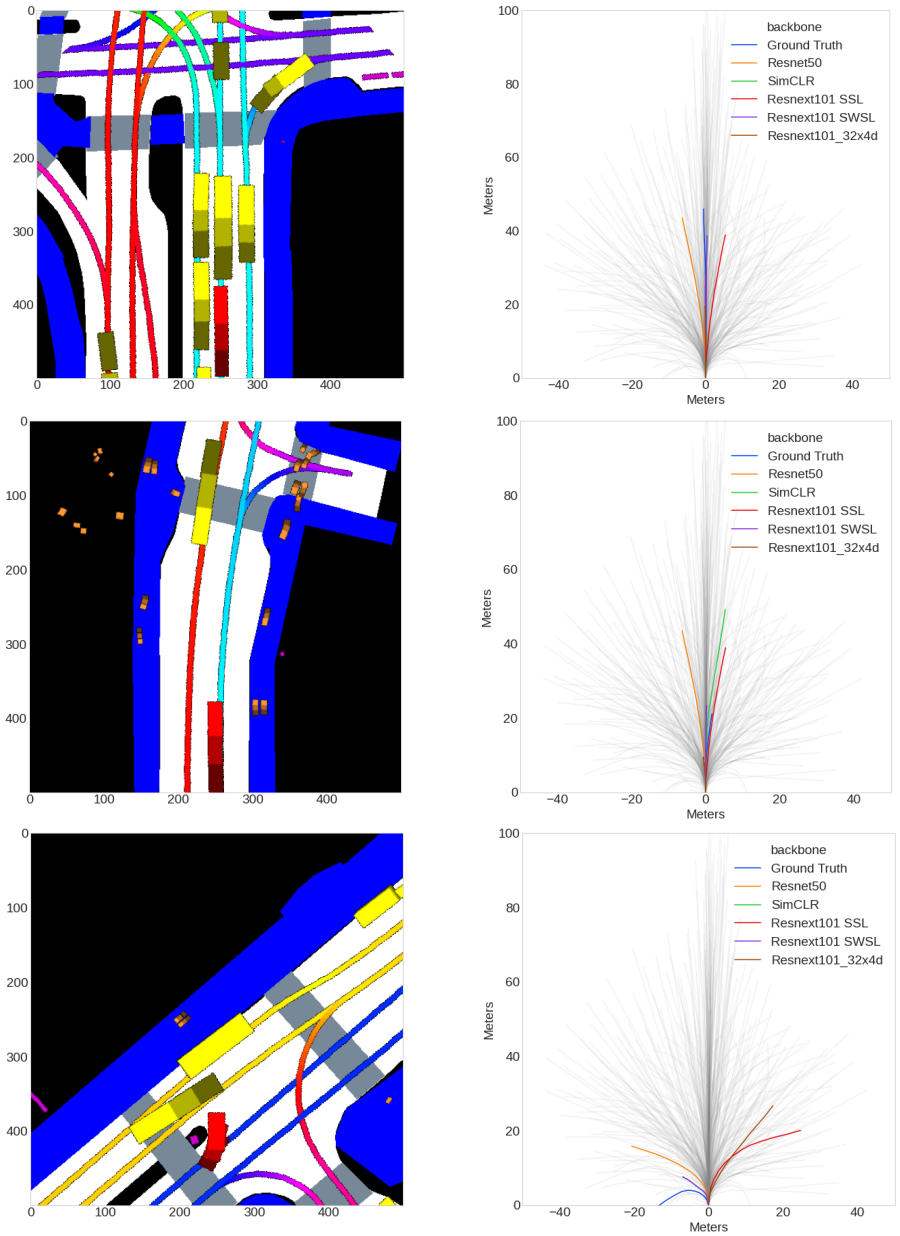


Figure 5: Trajectory prediction examples on the nuScenes dataset using CoverNet. The mid-level representation of the annotated map is on the left. The colored lines on the right represent the ground truth trajectory (blue) and those predicted by CoverNet with various backbone models, including the supervised 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.

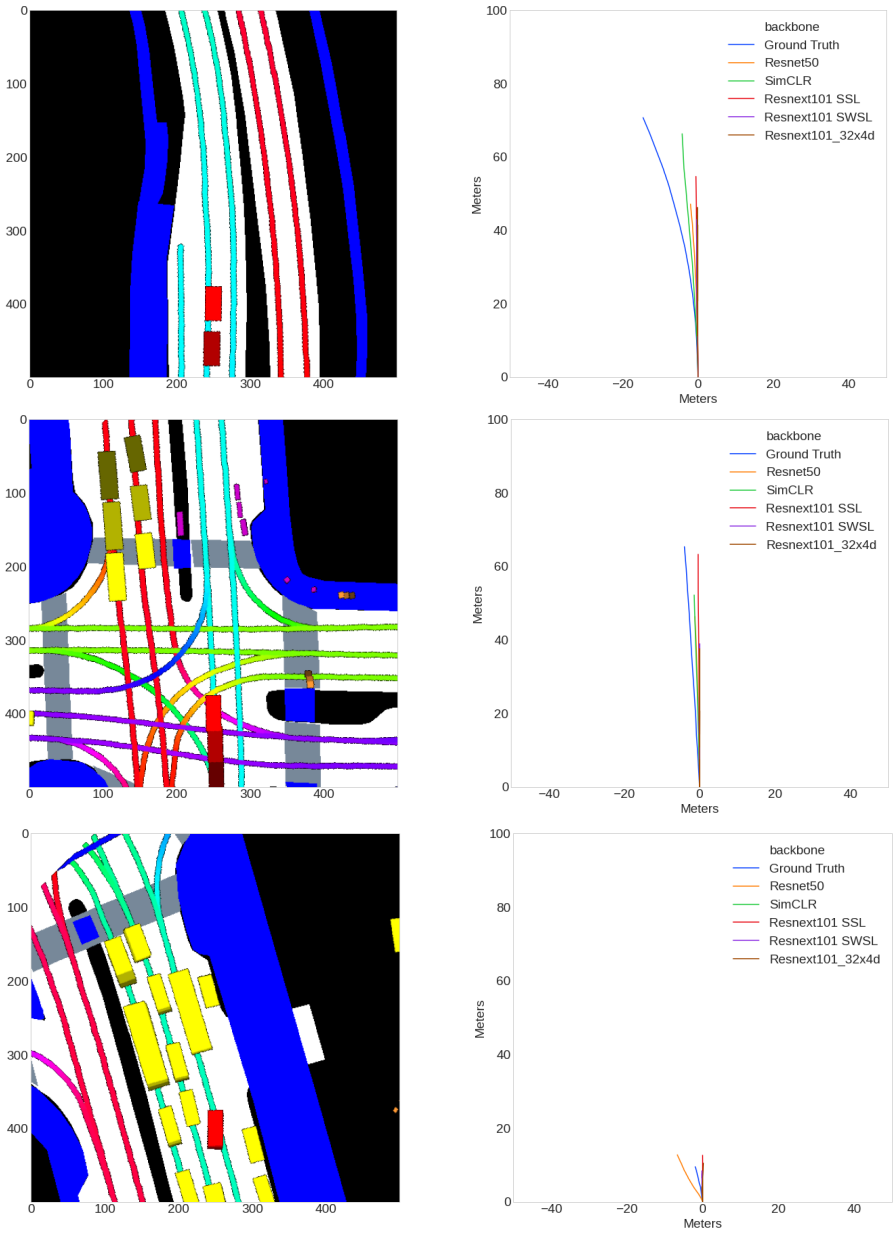


Figure 6: Trajectory prediction examples on the nuScenes dataset using MTP. The mid-level representation of the annotated map is on the left. The colored lines on the right represent the ground truth trajectory (blue) and those predicted by MTP with various backbone models, including the supervised ResNet-50 (orange) and the semi-supervised models we evaluate. In the third example, we show a dense traffic scenario.