DeepAutoTrack (DAT): Vehicle Trajectory Prediction for Autonomous Driving

We would first like to thank the reviewers for the supportive comments and the constructive criticism. Your valuable feedbacks help us further improve the analysis of the proposed method and the overall presentation of the paper. Due to the significant improvement made to the new version of the paper, we would kindly ask reviewers refer the new version of the paper.

Modifications: In what follows, we give an overview of the main changes made to the manuscript.

(1) We have implemented SocialLSTM [2] as requested by R2 and preliminary results on 2D space exhibit that the inclusion of “social interaction” further improves the prediction accuracy.

(2) We have restructured our methodology in section 3 and further explain the intuition behind the chosen architectures.

(3) We have rewritten our contributions in Section 1 and make our contributions more clearly.

We would like to address the issues that are common to reviewers respectfully as follows:

Prior works on sequence prediction: We found that the first version of the draft is indeed weak in presenting prior works on sequence prediction. We have restructured our related works. Specifically, we have supplemented works on activity forecasting/trajectory prediction works in the “Activity forecasting” part.

SYNTHIA and release of the dataset: Currently, experiments on the SYNTHIA dataset [27, 37] focus on single frame, static images for semantic labeling or scene representation. To our knowledge, our paper is the first attempt to utilize *continuous video streams* for the prediction problem. We would like to release publicly-available dataset for prediction challenge based on SYNTHIA to spur future research upon paper publication.

Instance association via tracking: We would like to clarify the overloaded term “tracking by detection” used in our paper. Most of the modern trackers focus on the problem of “class-agnostic” generic object tracking: in order to adapt to temporal changes, a continuous learning strategy is applied, where the model is updated rigorously in every frame. This excessive update strategy cause both lower frame-rates and degradation of robustness due to over-fitting to the recent frames. In the context of autonomous driving, however, we know in advance the classes of object of interests for tracking (*e.g.*, cars, pedestrians, cyclists). Therefore, with this extra source of information, we propose to use the neural nets object detection result as a more robust template to update the tacker’s model to overcome the problem of model drifting. Hence the tracker is served solely as *instance associator* for the traffic participant.

R1: Explain the intuition of three LSTMs.

R2:

The “average displacement error” is essentially the same as the “center error” in Tab.3 of our paper and the subtle difference is that the former is in unit space and the latter is in pixel space. Similarly “Final displacement error” is the last frame’s “center error” is unit space. “average non-linear displacement error”

Line 124 - The authors discuss a ”lack of proper metrics for evaluating the temporal prediction result.” How are the current metrics poorly suited to the task? What would be better?

In the SocialLSTM papers Related Work section, there are about 20 sources that focus on Activity forecasting

The details about the fusion of location and semantic information into the LSTM hidden state seemed missing (was a bit vague). A figure of the architecture of the SEG-LSTM would go a long way in clarifying this (or are the authors using Xu et al.’s ”End-to-end Learning of Driving Models from Large-scale Video Datasets” FCN-LSTM from Trevor Darrell’s group?).

How is this more than just an engineering pipeline, combining tracking (distilling bounding boxes to coordinates), and then adding an LSTM? SocialLSTM already did the second part in principle.

Why are scene semantics privileged (which you also call auxiliary) information? If the pipeline relies on fusing the location and semantic stream, if privileged information is missing at test time, then wont the pipeline break? (privileged information is usually present at train time, but absent at test time).

Can multiple objects be tracked at once, or just one? The video in the supplementary material only showed one object being tracked. What was the average number of objects in each SYNTHIA scene frame?

The focus of the paper seems a bit unclear: apparently, the goal is to predict cars future odometry given previous egomotion visual input. But the paper is about prediction of other vehicles trajectories, not about predicting ones own car odometry (which is different).

I felt like the detection/tracking theme slightly bogs down the main focus of prediction in the paper. I think refocusing it on prediction would strengthen the argument.

R3 There are too many contributions. For example, I would combine 1 and 3 and maybe also 2 and 4. In general, it is better to have a few strong contributions than many weak ones. Section 3 is quite short and it seems that the tracking methodology would fit well here. In general a lot of methodology detail is actually in section 4 (implementation details). Maybe consider reorganiz-

ing a little bit.

We thank the reviewer for the very meticulous review. We have refined our contributions in the introduction section and restructured the methodology section. We have also corrected the grammatical misuses accordingly.

In a real car accurate depth information might not be available. Also, tracking and semantic segmentation results in the real world will probably be noisier than for synthetic data. Experiments on real data would make the results much stronger. Do you think the results you obtained with the Synthia data will transfer to real-world data? Not exactly a request, but maybe something to consider for future work: There are photo-realistic simulators available now (CARLA, Sim4CV, etc.) which allow simulation of autonomous driving with real physics and photo-realism similar to Synthia. It would be interesting to implement this system and evaluate real-world performance.

We have indeed noticed the recently released photo-realistic simulators CARLA [2] and Sim4CV [3]. And we found that CARLA is especially suited as a further extension to verify our proposed system due to the availability of continuous video streams and ground truth segmentation. In terms of transferring to real-world data, from the semantic segmentation results of Synthia, we have observed that model trained on synthetic dataset produced good segmentations by itself on real datasets, and dramatically boosted accuracy in combination with real data. Hence, we believe the hybrid of synthetic data and real-world data would be the most data efficient and promising way for the model generalization. Model verification on public real-world datasets, e.g., Comma.ai [4] Udacity [1] dataset would be our forthcoming imminent works.

References

[1] Udacity. public driving dataset. https://www.udacity.com/self-driving-car, 2017. 2

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 2017. 2

[3] M. Mueller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem. UE4Sim: A Photo-Realistic Simulator for Computer Vision Applications. *ArXiv e-prints*, Aug. 2017. 2

[4] E. Santana and G. Hotz. Learning a driving simulator. *arXiv preprint arXiv:1608.01230*, 2016. 2