Multimodal trajectory prediction in the CARLA simulator

Francesco Marchetti Federico Becattini Lorenzo Seidenari Alberto Del Bimbo
Media Integration and Communication Center (MICC)
University of Florence
Email: name.surname@unifi.it

***Abstract*—Trajectory prediction is a key ingredient to ensure the safety of autonomous vehicles and the moving agents in the surrounding environment. To establish and compare the capabilities of forecasting methods, quantitative metrics such as Final Displacement Error (FDE) and Average Displacement Error (ADE) are commonly adopted. While these metrics provide a fundamental tool to compare different approaches, it is difficult to grasp the actual effectiveness of such methods, especially for approaches that output multimodal predictions. We propose a simulated environment based on CARLA in which to showcase both qualitatively and quantitatively the performance of several trajectory forecasting methods, with a particular interest on MANTRA, a recent state of the art predictor.**

I. Introduction

In autonomous driving, ensuring safety is of primary importance. The system must decide which is the safest manoeuvre for every time step to avoid collisions with other agents, structures and obstacles in the scene. To choose the best manoeuvre, it is important to be able to predict future movements of surrounding vehicles and pedestrians. For this task, we have developed a memory-augmented neural network, MANTRA, recently presented in [1], [2]. MANTRA generates multi-modal trajectory predictions of agents: a writing controller stores non-redundant observed examples and a reading controller reads likely information from memory given the observed past and context. In this demo, we demonstrate MANTRA in CARLA [3], a simulator for autonomous driving research based on Unreal Engine. While our self-driving car drives in urban streets, the model predicts future trajectories of surrounding agents, consistently with their past motion and the context. MANTRA is trained offline with the CARLA-PRECOG dataset and is tested in the simulator. During the simulation, it is possible to switch model, showcasing also baseline methods (Kalman filter, Multi-Layer Perceptron or modified MANTRA versions) to show the difference between different models. Also, any additional model can be successively loaded into the simulation to verify the efficacy of custom trajectory predictors. Common metrics such as Average/Final Displacement Error (ADE/FDE) are shown in real time during the simulation. Similarly custom plots can be added to the interface, monitoring variables such as memory size of MANTRA. At the end of the simulation, a summary about predictions and their FDE and ADE errors, compared to the ground truth of each agent, is saved to be analyzed later. The simulation is able to run 10 fps in a laptop with Intel i7-10510U, Nvidia GeForce GTX1650 and 16GB RAM in a scenario with 15 agents to predict simultaneously: in addition to run the simulator, the computational time includes pre-processing data for the architecture, model inference and rendering predicted trajectories in the scene.

II. CARLA

CARLA (Car Learning to Act) is an open-source simulator for autonomous driving research. CARLA has been developed to support the development of autonomous urban driving systems allowing to face the fundamental tasks in this research field such as trajectory prediction, motion planner, 3D tracking. CARLA is based on Unreal Engine to run the simulation and uses the OpenDRIVE standard to define roads and urban settings. Control over the simulation is granted through an API handled in Python and C++. CARLA simulates a dynamic world and provides a simple interface between the world and an agent that interacts with the world. To support this functionality, CARLA is designed as a server-client system, where the server runs the simulation and renders the scene. The environment is composed of 3D models of static objects such as buildings, vegetation, traffic signs, and infrastructure, as well as dynamic objects such as vehicles and pedestrians.

III. MANTRA

MANTRA, Memory Augmented Neural TRAjectory predictor, is a novel trajectory forecasting model that exploits memory augmented networks to effectively predict multiple trajectories of other agents in an urban context. The model is trained to store training samples in an external memory, which can be explicitly addressed at inference time to guide predictions and read trajectories that are likely to occur in the future. A writing controller is trained to store significant and previously seen non-redundant examples in a persistent memory and a reading controller is trained to generate different trajectories that cover the plausible movements of an agent, given its past trajectory and the context in which it is located.

IV. Simulation

We use two different settings regarding the memory in MANTRA: predetermined memory or online memory. In the former, the memory is filled during an offline training; in the latter, memory is filled online during the simulation using the writing controller. In both cases, predictions are accurate

Fig. 1. A example frame of a simulation.

and plausible. In the online setting, the accuracy improves as memory is populated: at first, samples will often be stored to improve predictions, but, as the simulation continues, less examples will be saved to avoid redundancies.

The simulation is interactive and the user has several commands available to control and display the run in a different way. The user can change the point of view (key C) between camera and top view, pause and restart the simulation (key P) or advance a frame at a time (key N), start and stop a recording and save the video to the disk (key S). It is possible to increase and decrease the number of predictions generated by the MANTRA model (key -/+) from one to a maximum of 10. With the TAB key, it is possible to show and hide the simulation information through an Head-Up Display (HUD). During the simulation, the user can switch the predictor (key M) to utilize other baseline models beyond Mantra (Multi-layer Perceptron, Kalman Filter) or modified MANTRA versions (reading controller based on only past or context) to show the relevance of individual components. Baseline models are able to generate only single predictions losing multimodality. Instead, MANTRA with the memory controller based on only past generates trajectories that go off-road, while with the one based on only context generates less precise predictions. In Figure 1, it is reported a frame of the simulation, where the left side the HUD shows the significant information of the current run: the memory settings, if the simulation is running or paused, if the registration session is active, the name of the selected forecasting model, the memory size of MANTRA (if selected), the number of predictions generated by the model, the maximum and mean Final Displacement Error (FDE) at 4 second between the predictions and the ground-truth for all agents in the scene.

V. Conclusion

In this demonstration we propose a tool to showcase the effectiveness of trajectory prediction models within the CARLA simulator. In the simulation a vehicle drives through and urban environment, predicting the future trajectory of surrounding agents and displaying it on screen, directly in the simulated scenario. Different models can be compared, highlighting the differences between multimodal and non-multimodal approaches, providing insights about the effectiveness of the methods.

References

[1] F. Marchetti, F. Becattini, L. Seidenari, A. Del Bimbo, *Mantra: Memory augmented networks for multiple trajectory prediction*, in Proc of CVPR, 2020.

[2] F. Marchetti, F. Becattini, L. Seidenari, A. Del Bimbo, *Multiple Trajectory Prediction of Moving Agents with Memory Augmented Networks*, transactions on Pattern Analysis and Machine Intelligence, 2020.

[3] A. Dosovitskiy and G. Ros and F. Codevilla and A. Lopez and V. Koltun, *CARLA: An open urban driving simulator*, in Proc of CoRL, 2017.