Integrating Supervised and Reinforcement Learning for Heterogeneous Traffic Simulation

Yasin M. Yousif[1 [0000−0002−5282−7259]] and Jörg P. Müller[1 [0000−0001−7533−3852]]

Technical University of Clausthal, Institute of Computer Science, Germany
https://www.ifi.tu-clausthal.de/

Abstract. Traffic simulation is needed for planning safe routes of self-driving cars and in analyzing traffic situations of a given area. Commonly supervised learning methods of vehicle, bicycle, and pedestrian traffic models have several limitations such as drifting errors and weak generalization to novel scenarios. Reinforcement learning can address these issues but it is much slower to converge due to the large state and action spaces involved in real-world traffic.
To overcome this challenge, a hybrid methodology that combines supervised learning for short-term agents kinematics and reinforcement learning for long-term trajectory planning is developed in this work, and then tested on two distinct heterogeneous traffic datasets: 1) InD dataset of intersections traffic and 2) UniD dataset of shared space traffic. The results showcase the effectiveness of this method as it outperforms the supervised learning baseline models in terms of lower average displacement errors, increased success rate, and higher survival time for simulated agents. Additionally, the generalization of this approach was demonstrated by testing it on both regular intersection traffic and shared space traffic.
This approach combines the benefits of supervised learning's ability to learn complicated systems like vehicle kinematics and reinforcement learning's potential for long-term planning in real-world traffic situations. [1]

Keywords: Heterogeneous Traffic Simulation · Reinforcement Learning · Multi-modal Output

1 Introduction

The challenge of planning and controlling the movement of autonomous vehicles in traffic requires modeling of other agents' movements within the scene. For instance, ChauffeurNet [1], a renowned autonomous driving model, has a dedicated component to predict neighboring agents' behavior. This is crucial for ensuring safe and efficient navigation of the car.

The objective at hand entails predicting the surrounding agents' movements over a fixed time horizon based on their observed past paths and the global map of the scene. Achieving this goal can be accomplished through simulating traffic, where future collective movements are predicted in a multi-agent manner.

[1] code and video demonstrations: https://github.com/engyasin/SLRL

However, the problem becomes increasingly difficult due to the intricate interactions among the possible movements of various agents at each time step. This complexity is further increased when dealing with heterogeneous agents of different types such as pedestrians and vehicles sharing space or navigating intersections.

Previous research has tackled this challenge using a variety of supervised learning neural network architectures [13,14,4,19,18] that leveraged massive public traffic datasets for training. Alternatively, rule-based models, which are dependent on assumed equations of movement, have also been employed, such as Social Force Model [7] or Cellular Automata [12]. These methods yielded realistic traffic predictions; however, their accuracy was generally lower than that of data-driven approaches.

In these state-of-the-art supervised learning models, the problem setting was simplified to single agent prediction, which resulted in unrealistic outcomes, such as off-road or collision predictions [17,5]. Supervised learning is also known for its sensitivity to biases present in its training dataset and limited exposure to extreme cases, which can lead to drifting error [1]. Consequently, this may hinder the model's ability to recover from errors over time.

Reinforcement learning offers a powerful solution to overcoming drifting errors in dynamic environments by training within a simulated model which allows free exploration of all possible states, resulting in increased robustness. Examples where reinforcement learning has been successfully used in traffic simulations include [17,5]. However, the key to successful training in reinforcement learning lies in designing an accurate reward model. This is usually done manually as seen in [17,2], where rules for non-collision and driving within road boundaries are appropriately included to ensure realistic results.

Another requirement arrises, if we exclude collision and driving outside road for vehicles, or outside sidewalk for pedestrians, where the exact kinematic model of movement relating to realistic velocity and acceleration is still in need to be enforced. Setting explicit rules for these kinematic models, is complicated and tricky process. Supervised learning can be leveraged at this point, to learn these relations from a traffic dataset. This motivated the proposal of the method here: a hybrid training strategy of supervised learning (SL) and reinforcement learning (RL) for heterogeneous traffic simulation.

Specifically, short-term steps are learned from the data with a supervised model, leading to a multi-modal output. This multi-modality of the output is based on clustering the position of the final point on each trajectory, referred to here as the destination. This is clarified in the left of Fig 1, where the terminal points with lower intensity are the destinations. Then the frozen trained step-wise models are used by a higher level selector model, which is trained within a simulation environment to maximize a manually designed reward for a longer time horizon. The selector model job is to select one of the multi-modes of the frozen model. This has the advantage of greatly reducing its action space to a discrete choice of fixed size. After that, the selected sub-model will predict the exact location. This procedure is shown in the right of Fig 1.

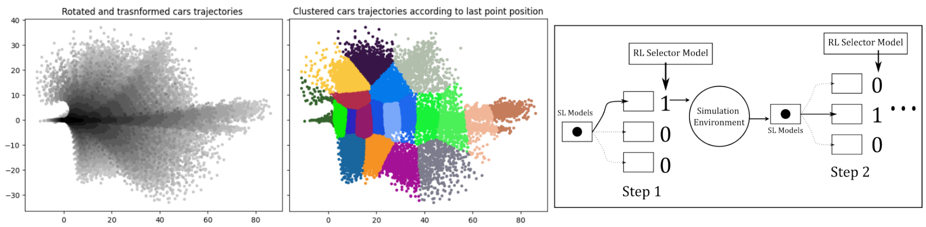

Fig. 1. Car trajectories in InD dataset, rotated and transferred to make it symmetrical on the left. The clustered trajectories with K-Means according to position after 4.8 seconds in the middle. The integration of the two models (RL and SL) to generate a possible long-term trajectory on the right.

Based on that, the learning of the different models in the simulation will be interactive, as well as realistic due to the learning from the datasets. This methodology is found to be efficient and practical in comparison with other learning methodologies that address the same issue of combining supervised learning with reinforcement learning, like reward augmented imitation learning in [2] or [11]. These methods are hard to tune for realistic heterogeneous traffic data, due to the complexity of the reward structure, as it contains varying ranges of the learned vs the manual rewards, making the learning unstable [15].

To test the accuracy of this method, two datasets are used: InD [3] for intersection traffic, and UniD [6] for shared space traffic. In both dataset, criteria like average and final displacement errors (ADE,FDE) are calculated for the test splits. Additionally, performance measures, like survival time and average speed are also measured.

The main contributions of this work are summarized as follows:

1. Introducing a new training methodology for a realistic simulation of a heterogeneous multi-agent traffic. This method avoids the problems that would arrise from using each of reinforcement learning or supervised learning by itself.
2. Introducing a simple simulation framework, which is needed for interactive training of reinforcement learning. It also has the ability to produce random road networks of intersections, which helps the generalization.
3. Testing on two distinct traffic regulations, namely, regulated traffic in intersections using InD dataset, and unregulated traffic in shared spaces using UniD dataset. Then evaluating criteria like success rate, average speed and displacement errors and comparing directly with the SL baseline, and indirectly with the previous multi-modal SL methods like [13], as well as a state-of-the-art traffic planning method done in a different simulation setting [17].

In the following sections, the problem formulation is introduced in section 2, then in section 3 the methodology is presented, and lastly the experimental results and discussion followed by the conclusion are in sections 4 and 5 respectively.

2 Problem Formulation

As the higher level planning model is based on reinforcement learning here, the task is formed as Markov Decision Process (MDP), which is defined by the tuple: $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \rho_0)$, where $\mathcal{S}$ is the set of all possible states, known as state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the set of transition probabilities among states, ρ_0 is the distribution of the initial state s_0, and $r_i = f_r(s, a_0, a_1, \cdots, a_N) : \mathcal{S} \times \mathcal{A}^N \to \mathbb{R}$ is the reward signal where N is the number of agents and i is the ego agent index. Lastly $\gamma \in (0, 1)$ is the discount factor used to optimize the policy. Furthermore, a stochastic policy is denoted here as $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and the expert policy π_E, is the target policy generating the trajectories in the datasets.

The state vector in our experiments contains the speed, acceleration, and the image of the road around the agent. The action vector is a discrete categorical output of the policy π_r, to choose one output from a frozen multi-modal supervised learning π_s policy, as shown in the right of Fig. 1, the latter a_{cont} is the continuous displacement on x and y axes for the next timestep, which will allow the simulation to advance. This can be written as:

$$a_{cont} = \pi_s(s_i, \pi_r(s_i)) \tag{1}$$

The objective of a reinforcement learning policy π_r is to maximize the return, which is the discounted rewards sum over T timesteps of an episode, expressed as:

$$\pi_{ri}^{opt} = \arg\max_{\pi_{ri}} \mathbb{E}_{\pi_{ri}}[\sum_{t=1}^{T} \gamma^t r_i(s_i^t, \{a_i^t\}_{i=1}^N)] \tag{2}$$

It is worth noting that the reward here is a function of the ego state and all of the other agents actions, which represents the multi-agent nature of the problem, where the optimal policy π_{ri}^{opt} for agent i is an equilibrium among all the interacting policies $\{\pi_{r0}^{opt} \cdots \pi_{rN}^{opt}\}$.

3 Methodology

The following subsections start with a description of state and action spaces. Then, it is followed by a discussion about the clustering of the dataset into multiple modalities. The design of the reward for the RL model is clarified next. Afterwards, the training procedure for this RL model is presented, and lastly the evaluation criteria and inference process in the full model are explained.

3.1 Defining State and Action Spaces

As shown in Fig 2. the state consists of the following values for each agent: speed, acceleration, type, heading, number of vulnerable and non-vulnerable road users within each of 5, 10 and 20 meters of the ego agent. This represents the input for the supervised learning model. The reinforcement learning model has an additional access to an image of the ego agent moving along the x axis and located at fixed location extracted from the simulation. This form of the image is calculated by rotating according to the agent heading and transferring it relative to its position, as in Fig 2.

In this image, the road is in white color, the sidewalk is yellow, and each of the pedestrians, cyclists, cars are represented as green, blue and red line segments respectively.

The action vector of the reinforcement learning model is one-hot encoded vector indicating which mode should be used, out of multiple modes generated by clustering the training datasets.

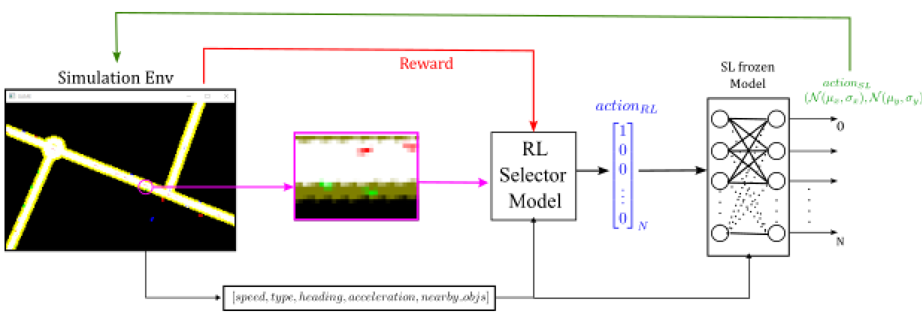

Fig. 2. The workflow of the full model. The supervised learning model is independently trained on K modes, and then frozen to be used by the reinforcement learning model. The latter is trained according to manually set reward function. All agents advance together in the simulation leading to the generation of the next states.

3.2 Clustering of the Training Dataset

First the trajectories of each traffic agent type is preprocessed, by centering each sample in fixed position, and rotating it so the agents direction of movement align with the x axis. For example, Fig 1 on the left shows the full processed cars trajectories in InD dataset.

After that, to represent the multi-modality nature of the prediction, the trajectories are clustered according to the location of its destination points using KMeans algorithm, as shown in the middle of Fig 1. This destination-based clustering of trajectories is also used in previous works [13,4].

The number of modes here is 20, which is commonly used in other works [13,4]. This is done for every agent type, leading to 60 splits of the full dataset of pedestrians, cyclists, cars. This results in 20 supervised learning models per agent type, where each of them predicts one step into the future for one specific goal sub-area and one agent type. The destination points are just future locations of the agent after 12 timesteps, where timestep is 400 miliseconds. These supervised models only depend on the vectorized data from the datasets, as there is no simulation to get image input. The next step is to train a selector model within the simulation; however, a representative reward should be designed to ensure realistic generation of trajectories.

3.3 Reward Design

The design of a suitable reward for the RL model is a crucial step for successful training. In this work, we care about the following objectives, which are represented in the reward:

- Non-collision: Agents shouldn't collide, or get very close to each other in a dangerous way: `r1 = -1 * agents_closer_than(1)`
- On-road movement: Cars should only drive within roads and not on sidewalks. Pedestrians and cyclists can move on sidewalks; however they are slightly discouraged from moving on roads: `r2 = -1 * is_outside_road()`
- Minimum speed: all agents are encouraged to move faster than a minimum speed to avoid the full stopping solution, which is a valid solution because it avoids any collisions: `r3 = 1 * (speed>0.5)`

3.4 Training with Reinforcement Learning

Using the defined action and state spaces, the online reinforcement learning algorithm of Proximal Policy Optimization (PPO) [16] is used for training with the defined reward signal. PPO learns actively while stepping in the environment, episode by episode, for all agents in the timestep. This will allow all the possible modes to interact with each other in the simulation during training, including models of different road users types, leading to better compatibility among all modalities and better exploration of all the possible states. After each step, the simulation environment will be redrawn with opencv library [9], as shown in Fig 3 of the two simulation environments. The density of traffic will vary from episode to another, by changing the size of the road network, which will allow training over different setting of traffic, i.e. dense vs sparse.

Lastly, the environment is infinitely looping on itself, where each agent is teleported to the other side of the road whenever it gets out of the scene.

3.5 Evaluation of full model

After training the full model, the evolution is done using two sets of criteria:

- Direct criteria for the plausibility of the movement, like collision rate, average speed, and average survival time of the agents.

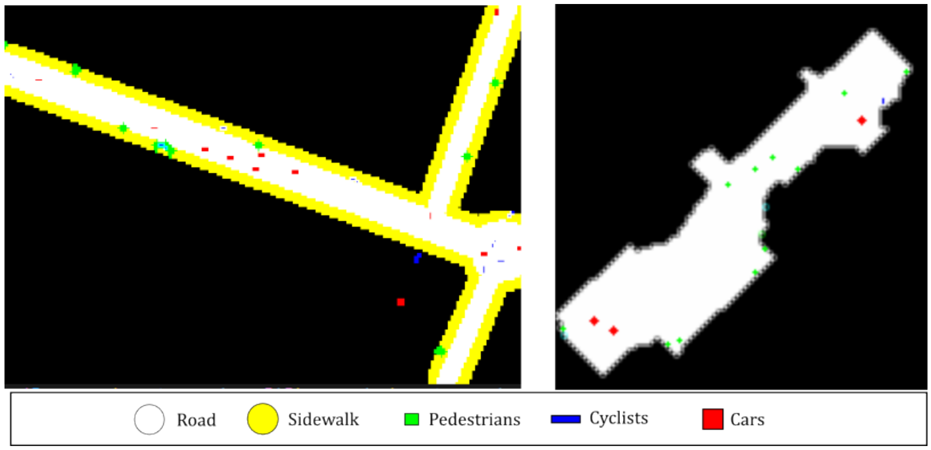

Fig. 3. The two simulation environments: intersection simulation on the left, shared space simulation on the right. Red agents are cars, green are pedestrians, and blue are bikes

- Traffic prediction criteria of final and average displacement error (ADE), (FDE).

These measures will enable the comparison with previous works, as well as, with the supervised learning baseline.

The testing of traffic prediction accuracy is done using a manually drawn environment similar to the real one, by initiating the agents states from the real data in each step. This, however, will slow down the testing process considerably, which prevents us from testing the multi-modal case (testing different modalities of output for each time step). However, the model is still multi-modal because we can sample infinitely from the output categorical distribution.

It is worth noting that the evaluation is aiming to test the similarity between the simulated and real trajectories, as this is what matters for traffic prediction. Research related to autonomous driving may care only about safe planning of the trajectories [1]; therefore using other sets of criteria.

4 Experimental Evaluation and Discussion

The testing was done on two heterogeneous traffic datasets:

- InD [3]: an intersection traffic datasets, containing four different scenes, where three scenes are used for training in our model and the fourth for testing, as in [13]. The intersections contain different arrangements of the sidewalks and traffic islands.
- UniD [6]: This is a more recent dataset of one scene, representing a university campus gate. This scene doesn't contain sidewalks, and is regarded as shared

space. The cars have relatively slower speed there. The dataset is split to 80% training part and 20% testing part.

In both of these datasets, only three types of agents are learned, namely, cars, cyclists and pedestrians, which represent the greatest part of the road users types in these datasets.

4.1 Testing the full model

After clustering each dataset into 20 different modes of each of the three agent's types depending on the destination points. A supervised learning model is trained for each mode to output the next displacement over x and y. The input of the models are the vectorized input of speed, acceleration and the counts of nearby agents with multiple ranges. The task of choosing which mode is best suited for each state is left for a reinforcement learning model; however, a baseline supervised model is trained to predict the distribution over modes for the next step as well. These models are trained for 35 epochs, with Adam optimizer [10] for both RL and SL models. The PPO implementation for training the RL model here is based on the CleanRL implementation [8].

First, the average displacement error (ADE) is found for every step of 16 steps, as shown in Fig. 4 for InD dataset and Fig. 5 for UniD dataset, for both of the SL+RL and SL baseline and all of the three agents' types. The trajectories are simulated collectively for all the agents and then compared with the ground truth in the test split of both of the datasets.

It is noted that the state of the art prediction for InD dataset in the time of writing, has an FDE of around 0.54 meters for pedestrians [4] as a multi-modal output with 20 modes. Therefore, the uni-modal error value here of 2.5 meters for pedestrians seems very sensible.

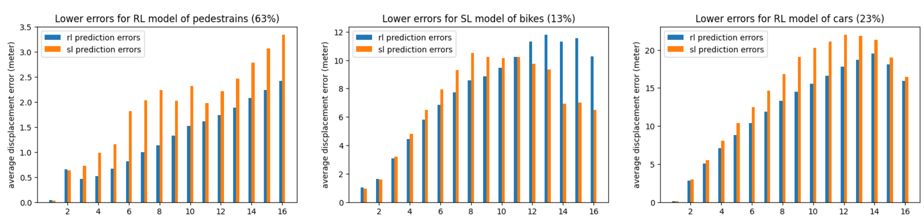

Fig. 4. Average displacement errors for all three traffic types in InD dataset over steps order. The supervised-reinforcement learning model has lower errors with increased steps than the SL baseline due to its interactive training, except in the bike case, which represents only 13% of the traffic

Second, criteria like survival time, which is the average timesteps passed over the agent before it collides, success rate, which is the percentage of cars that

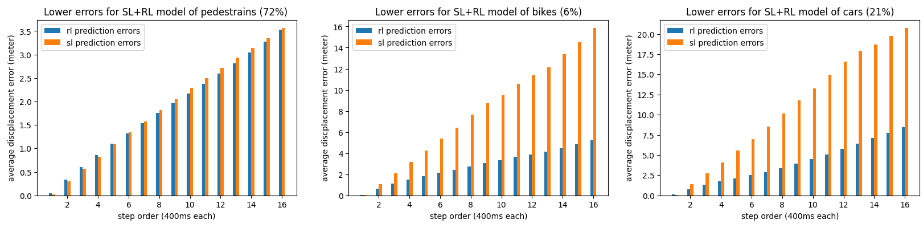

Fig. 5. Average displacement errors for all three traffic types in UniD dataset over steps order. The supervised-reinforcement learning model has lower errors with increased steps than the SL baseline due to its interactive training.

drive without any collision through the episode, and lastly average speed are calculated and compared with the SL baseline in addition to the ground truth values using the test splits, and a recent work (iPLAN)[17] for heterogeneous traffic simulation. These criteria represent a critical features of the traffic, where a collision is a rare occurrence in real usual traffic and the average speed can be seen as a measure for the realism of the simulation. The collision here is defined when two agents get within 0.5 meters from each other.

It's important to note that the comparison with the results in [17] is only to appreciate our current results, as the evaluation setting and simulation frameworks are different there. The results are shown in Tabel 1 for InD dataset, and Table 2 for UniD dataset.

Table 1. InD dataset results of success rate, survival time and average speed for 64 episodes with 128 steps for each.

Method	Average Speed (m/s)			Success rate (%)	Survival time (timestep)		
Types	Peds	Bikes	Cars	Cars	Peds	Bikes	Cars
Supervised Model	0.5	3.7	5.9	28 ± 25	107±26	78±40	59±31
Our Model	0.16	1.12	2.4	43 ± 30	107±35	89±39	79±32
Ground Truth	1.33	3.8	2.9	99 ± 5	25501 ± 372		
iPLAN*	-	-	21	67.8 ± 6	76± 3		

* Not directly comparable as the simulation framework is different

4.2 Discussion

In Fig 4 and Fig 5, it is clear that the SL+RL model has lower ADE than the SL model in general. This difference in errors is increasing with the timestep order, which is due to the drifting issue of SL models.

One exception is for the case of bikes in InD dataset, where the difference in errors is increasing till the ninth step, then it decreases again. By looking at

Table 2. UniD dataset results of success rate, survival time and average speed for 64 episodes with 128 steps for each

Method	Average Speed (m/s)			Success rate (%)	Survival time (timestep)		
Types	Peds	Bikes	Cars	Cars	Peds	Bikes	Cars
Supervised Model	0.17	6.9	14	22 ± 16	101±12	17±21	19±19
Our Model	0.38	0.77	1.6	33 ± 28	92±12	60±43	73±29
Ground Truth	1.27	2.96	3.1	100 ± 0	22497 ± 5371		
iPLAN*	-	-	21	67.8 ± 6	76± 3		

* Not directly comparable as the simulation framework is different

Tabel 1, it is also noted that the average speed in the SL model is closer to the ground truth than the same quantity for SL+RL model. This can be explained by the smaller percentage of bikes in the dataset (only 13%), which is reflected also in the simulation, leading to weaker learning of the bikes case. Another factor is the relative ambiguous changing rules for bikes traffic.

In Fig 5 for UniD dataset, the drifting error is more clear than the InD case, which can be explained by the simpler setting of the shared space, where road features like sidewalks play minor role, and the movement has more freedom.

In Table 1, the SL+RL model has better results for survival time and success rate, however, the average speed for SL model is closer to the ground truth. Additionally, both models perform better than the method in [17] regarding survival time.

In Table 2 of UniD, SL+RL model has average speed closer to the ground truth, than the SL model. The big average speed of cars for the SL model is caused by fitting some fast samples in the datasets, for the car case.

Essentially we can say that the SL+RL model has lower drifting errors than the SL method. It also provided realistic displacement errors, average speed and survival times, when compared with current state-of-the-art planning [17] and traffic prediction [13] methods in the literature.

4.3 Limitations

A notable limitation of our approach is the direct clustering step relying on K-Means. This process can be refined to mitigate issues such as the inherent statistical bias towards forward movements, which hinders agents from adopting more equitable turn-taking strategies.

Another area for improvement lies in incorporating learned rewards alongside manually designed rewards during the reinforcement learning phase. This could help us achieve a better balance between creating realistic and efficient traffic simulations.

5 Conclusion and Future Works

In this work, a new methodology for simulating heterogeneous traffic trajectories is proposed. This methodology is based on hybrid supervised-reinforcement learning approach. Using reinforcement learning allowed the enforcement of physicals constrains on the agents with manually designed rule-based reward, while still confirming to the agent kinematics in the dataset through the definition of different modes on it and learning each mode with a supervised learning model. The approach was tested on two datasets with different traffic patterns, and it proved its applicability and superiority compared with a pure supervised learning baseline and some effective previous works in autonomous driving [17] and trajectories prediction [13].

As a future work, further exploration into the hyperparameters optimization is warranted to improve the performance even more. Potential areas of investigation include refining the reward calibration, the number of modes, and incorporating additional complex simulation elements such as pedestrian crossings or bike lanes.

Acknowledgment

This work was supported by the German Academic Exchange Service (DAAD) under the Graduate School Scholarship Programme (GSSP).

References

1. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018)
2. Bhattacharyya, R., Wulfe, B., Phillips, D.J., Kuefler, A., Morton, J., Senanayake, R., Kochenderfer, M.J.: Modeling human driving behavior through generative adversarial imitation learning. IEEE Transactions on Intelligent Transportation Systems **24**(3), 2874–2887 (2022)
3. Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L.: The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 1929–1934. IEEE (2020)
4. Chiara, L.F., Coscia, P., Das, S., Calderara, S., Cucchiara, R., Ballan, L.: Goal-driven self-attentive recurrent networks for trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2518–2527 (2022)
5. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020)
6. leveLXData by fka GmbH: uniD: The university drone dataset. `https://levelxdata.com/unid-dataset/` (2021)
7. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)

8. Huang, S., Dossa, R.F.J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., AraÃšjo, J.G.: Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. Journal of Machine Learning Research **23**(274), 1–18 (2022)
9. Itseez: Open source computer vision library. `https://github.com/itseez/opencv` (2015)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Li, Y., Song, J., Ermon, S.: Infogail: Interpretable imitation learning from visual demonstrations. Advances in Neural Information Processing Systems **30** (2017)
12. Maerivoet, S., De Moor, B.: Cellular automata models of road traffic. Physics reports **419**(1), 1–64 (2005)
13. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15233–15242 (2021)
14. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 759–776. Springer (2020)
15. Osa, T., Tangkaratt, V., Sugiyama, M.: Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information. Neural Networks **152**, 90–104 (2022)
16. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
17. Wu, X., Chandra, R., Guan, T., Bedi, A., Manocha, D.: Intent-aware planning in heterogeneous traffic via distributed multi-agent reinforcement learning. In: Conference on Robot Learning. pp. 446–477. PMLR (2023)
18. Yousif, Y.M., Müller, J.P.: Generating explanatory saliency maps for mixed traffic flow using a behaviour cloning model. In: International Workshop on Multi-Agent Systems and Agent-Based Simulation. pp. 107–120. Springer (2022)
19. Yue, J., Manocha, D., Wang, H.: Human trajectory prediction via neural social physics. In: European Conference on Computer Vision. pp. 376–394. Springer (2022)