gap between privileged RL agents from sensorimotor agents

Privileged to Predicted: Towards Sensorimotor Reinforcement Learning for Urban Driving

Ege Onat Özsüer, Barış Akgün, and Fatma Güney

***Abstract*— Reinforcement Learning (RL) has the potential to surpass human performance in driving without needing any expert supervision. Despite its promise, the state-of-the-art in sensorimotor self-driving is dominated by imitation learning methods due to the inherent shortcomings of RL algorithms. Nonetheless, RL agents are able to discover highly successful policies when provided with privileged ground truth representations of the environment. In this work, we investigate what separates privileged RL agents from sensorimotor agents for urban driving in order to bridge the gap between the two. We propose vision-based deep learning models to approximate the privileged representations from sensor data. In particular, we identify aspects of state representation that are crucial for the success of the RL agent such as desired route generation and stop zone prediction, and propose solutions to gradually develop less privileged RL agents. We also observe that bird's-eye-view models trained on offline datasets do not generalize to online RL training due to distribution mismatch. Through rigorous evaluation on the CARLA simulation environment, we shed light on the significance of the state representations in RL for autonomous driving and point to unresolved challenges for future research.**

I. Introduction

The effort involved in engineering autonomous driving (AD) systems is immense, prone to failure, and has not yielded a full AD agent yet. As a result, the popularity of learning-based methods is on the rise. Learning from experts, also known as Behavior Cloning (BC), has shown success in simulated environments with careful design components. However, these are limited by the need for expert quality supervision and suffer from the distribution shift problem making real-world deployment problematic. Reinforcement Learning (RL) offers a promising alternative by utilizing existing data and/or environment interaction to correct errors, improve sub-par behaviors, and reinforce good ones, and potentially surpass human expert performance. However, RL has fallen short of its promises in self-driving, consistently trailing BC approaches in benchmark tests like the CARLA AD Challenge. In this paper, we investigate the reasons for this disparity and propose potential solutions, with the aim of unlocking the potential of RL for self-driving.

There are RL agents that achieve impressive driving performance in simulation but with the significant caveat of using *privileged* information. This includes any ground truth information relevant to driving. State representations obtained from such information significantly simplify the learning task. Chen et al. [1], in their seminal work, use these "expert" BC agents to supervise a "student" agent which uses only sensorimotor input. This paradigm, has led to the development of the RL agent ROACH [2]. ROACH attains impressive driving performance on CARLA, surpassing the performance of its contemporary BC agents, even those that benefit from privileged information (see Table I). This underscores the potential of RL, albeit with privileged information.

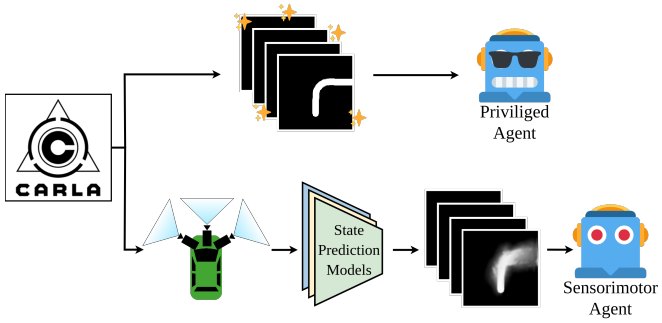

Fig. 1. **Privileged vs. Sensorimotor Agents.** State representations can make or break RL agents which is all too real for autonomous driving. Our aim is to investigate the BEV representations of a successful privileged agent, propose ways to reduce privileged information with learned vision models and discuss potential ways toward sensorimotor RL agents.

Given the success of these privileged RL agents, a natural follow-up question arises: Can we construct high-quality state representations without relying on such privileged information? In this paper, we delve into this question by analyzing the factors contributing to the success of privileged RL agents, using ROACH as a case study. ROACH adopts a bird's eye view (BEV) state representation, where critical elements like roads, lane lines, the desired route, stop-zones, other vehicles, and pedestrians are encoded as binary images in separate channels. Using BEV prediction methods, such as LSS [3] or SimpleBEV [4], to eliminate the dependence on privileged information fail due to poor BEV prediction performance. The primary culprits of this failure are class imbalance between BEV entities and distribution discrepancies between BEV training data and RL training data. More involved BEV predictors, such as BEVFormer [5], prohibit RL training due to space and computational requirements.

In this paper, we perform a fine-grained analysis of ROACH's privileged state representation, dissecting its individual components to understand their contributions. Our goal is to discover methods to mitigate the reliance on privileged information, ultimately paving the way for a fully unprivileged RL AD agent. An interesting finding of our analysis is the significance of the desired route component. The desired route component has not found application

KUIS AI Center and Dept. of Computer Engineering, Koç University
{eozsuer16, baakgun, fguney}@ku.edu.tr

TABLE I

COMPARISON OF THE ROACH EXPERT WITH EXAMPLE IMITATION LEARNING AND RL METHODS ON THE LONGEST-6 BENCHMARK OF THE CARLA SIMULATOR.

Method	TA	Prv.	DS	RC	IS
LbC [1]	BC	✓	24.08±2.83	73.36±1.08	0.31±0.06
NEAT [6]		-	24.08±3.30	59.94±0.50	0.49±0.02
WoR [7]	RL	-	17.36±2.95	43.46±2.99	0.54±0.06
ROACH [2]		✓	**60.14±2.40**	**85.83±0.60**	**0.69±0.03**

TA - training approach, Prv. - whether or not privileged information is used, DS - driving score, RC - road completion rate, IS - infraction score

beyond ROACH, and its prediction remains unexplored in the realm of learning-based AD. We introduce a middle-ground approach to predict the desired route from the BEV input. Furthermore, we demonstrate that a smaller BEV predictor, focusing only on the roads and the lane lines, can be trained to sufficient performance. Lastly, we integrate an unprivileged traffic light predictor, completely replacing the privileged stop-zone input. Our overall idea is depicted in Fig. 1. Our evaluations, conducted on the CARLA simulator, highlight that harnassing purpose-built predictors is a viable path forward for constructing a fully unprivileged state representation.

II. RELATED WORK

BC methods made significant strides in AD since the inception of the CARLA simulator [8] and Chen et al. [9] present a detailed overview of the field. However, BC methods suffer from the distribution shift problem, which causes the learned policy to make mistakes as it diverges from the states present in the expert demonstrations due to compounding errors. In order to circumvent this, Chen et al. [1] use ground truth BEV semantic segmentation maps as input to train an expert agent, which can provide supervision to a sensorimotor agent as it is deployed in the simulation. The availability of expert supervision for on-policy data allows the usage of data aggregation techniques like DAGGER [10] to mitigate the distribution shift. Another approach, called Learning from All Vehicles (LAV) [11], achieves higher performance by using every agent in the scene as a source of supervision.

An important aspect of autonomous driving is input representation. AD agents utilize multiple sensor inputs such as RGB images, LIDAR point clouds, GNSS coordinates, and IMU readings. Which inputs to use and how to combine them are important engineering decisions. One possibility for processing model inputs is using intermediate representations. Intermediate representations are inputs for an autonomous driving system, usually generated through processing the sensor inputs with a deep learning model or by accessing simulator variables. The expert model of Learning by Cheating (LbC) [1] is an example where a BEV semantic segmentation map is used as an intermediate representation. Behl et al. [12] investigate intermediate representations for autonomous driving, focusing on semantic segmentation and analyzing the task-relevant object classes, showing that a good intermediate representation can play a crucial role in driving performance.

Bird's eye view (BEV) as an intermediate representation is closely related to our work. Among the state-of-the-art models, Lift-Splat-Shoot (LSS) [3] uses a depth prediction module and projects encoded features from the camera space to the BEV space based on this depth estimation. The success of this method depends on successfully predicting the depth values of the RGB image. More recent BEVFormer [5] instead utilizes deformable attention to learn a mapping between image features and the BEV grid. BEVFormer achieves great performance in both semantic segmentation and object detection tasks. However, their heavy transformer-based architecture combined with the learned projection method results in a very large model that is difficult to use with RL. Finally, SimpleBEV [4] presents a more efficient approach with a similar performance by using a parameter-free bilinear interpolation between RGB and BEV space. We utilize SimpleBEV for RL training, due to its favorable trade-off between performance and computational efficiency.

The first deep RL method on CARLA, proposed as a baseline in the CARLA challenge [8], uses the A3C[13] algorithm with discrete actions [14] but falls short of BC methods. Liang et al. [15] present one of the first RL approaches that achieved impressive performance by using DDPG [16] with continuous actions where the policy network is initialized from an imitation learning agent. Toromanoff et al. [17] first train a network to predict affordances related to the environment along with semantic segmentation maps. They then freeze this network and use its bottleneck features to train an RL agent using the Rainbow-IQN algorithm [18]. Chen et al. [7] follow a model-based approach by factorizing the driving state and the world model. The world is assumed to be independent of the agent's actions which allows training in pre-recorded driving logs. Despite this limiting assumption, which almost never holds, it outperforms model-free RL methods. However, all of these are outperformed by BC methods.

Privileged agents outperforming sensorimotor agents is expected but surprisingly the gap is larger in the case of RL, leading to the conclusion that BC agents cannot utilize the same privileged data as effectively. We examine this by comparing the performance of four contemporary methods[1] on CARLA. Despite being the best-performing sensorimotor RL method at the time of writing, the World On Rails (WoR) [7] is the least performant. On the other hand, the vision-based NEAT [6] performs on par with the privileged Learning by Cheating (LbC) [1], both BC methods. Finally, the privileged RL method ROACH [2] outperforms the rest. This implies that at least one major problem with sensorimotor RL agents lies in their noisy, high-dimensional state representations. As such, we investigate the privileged state space components of ROACH and how they could be replaced with sensor-based approaches to improve sensorimotor RL agent performance.

[1]We perform the comparison between contemporary methods to highlight the impact of privileged information rather than recent developments in architecture, training procedures, etc.

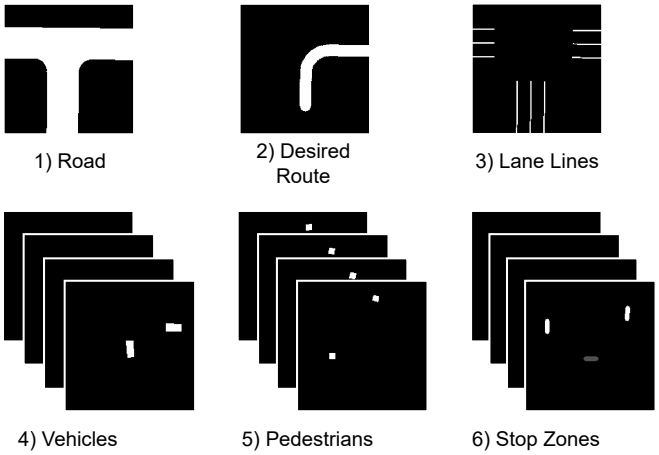

Fig. 2. **State Representation of ROACH.** ROACH uses a state representation where driving information regarding road topology, desired route, objects, and stop zones are stored in separate binary channels.

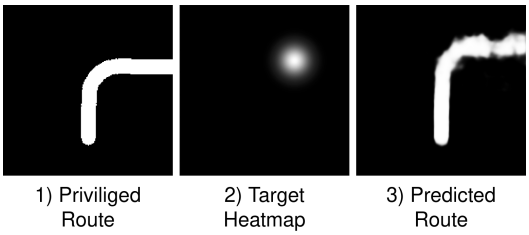

Fig. 3. **Potential Representations of Desired Route Information.** ROACH uses privileged information to create the desired route. We consider an unprivileged alternative target heatmap representation and also propose a model to predict the desired rote from road topology and target waypoints.

III. Toward Unprivileged RL for Self-Driving

The privileged information is critical to the success of the RL agent proposed in ROACH [2]. Our goal is to understand the reasons behind the success of the privileged RL agent and approximate its performance with an unprivileged sensorimotor agent. In particular, we separately focus on various components of the state representation. While some channels can be predicted from RGB only, others like the desired route require additional information such as GNSS and IMU readings. Predicting the location of stop zones without accessing the simulator's internal representation presents additional challenges. This chapter presents our method to address challenges associated with replacing privileged information in ROACH's BEV representation toward developing an unprivileged RL agent for driving.

A. State Space of ROACH

The state representation of ROACH consists of two components: the BEV representation and the measurement vector. The BEV representation consists of binary BEV segmentation maps of road topology, desired route, objects, and stop zones as shown in Fig. 2. The measurement vector is a vector of scalar measurements including the steering, throttle, brake, vehicle gear, and lateral and horizontal speed values observed in the last time step. These measurements are relatively easy to obtain and are not considered privileged.

The BEV masks cover a square area with sides of 38.4 meters, where the ego-vehicle is aligned to be horizontally centered, and 8 meters up vertically from the bottom of the map area. The BEV masks are rendered in 192×192 resolution and contain 15 channels. There is one channel each for roads, the "desired route", and lane lines. The desired route is the fine-grained path that the agent should follow to reach the next target waypoint. There are 4 temporally stacked channels each for vehicles, pedestrians, and "stop zones". The stop zones are rectangular regions where a vehicle should not move due to a traffic light or stop sign that controls the corresponding region. When a traffic light for a stop zone turns red, the corresponding area is rendered white in BEV. The information in the BEV masks is calculated by accessing the simulation internals and, therefore considered privileged.

B. BEV Perception

For BEV segmentation, we take SimpleBEV [4] which is designed and optimized for the real-world NuScenes [19] dataset, and adapt it to CARLA. The dominant paradigm in BEV segmentation, which is also followed by SimpleBEV, is to train a separate model for each class to segment. While training and using separate models for each class improves performance, it is infeasible to perform multiple forward passes for a single state representation in the RL loop. Therefore, we modify the SimpleBEV architecture to simultaneously predict multiple binary segmentation masks corresponding to each class. To that end, we simply increase the output dimension of the final 1D convolution layer of the segmentation head to match the number of classes.

Instead of multiple classes competing with a cross-entropy loss, we use a binary cross-entropy loss for training. This is necessary as a cell on the BEV grid does not necessarily belong to a single class, instead, multiple channels can be active together, for example, a vehicle positioned on the road. We also apply positive weighting when predicting classes that cover a very small portion of the grid, and, therefore dominated by negative samples such as pedestrians or lane lines. We found this weighting to be critical in our experiments.

While the desired route and traffic light can be predicted as additional channels in the output of SimpleBEV in a straightforward manner, we found it to be infeasible in our experiments. We suspect that there are two reasons for this. First, the desired route prediction needs to consider the relative positions of future waypoint coordinates. Second, the stop zones are spatially distant from the traffic lights, both in the image and the BEV spaces. This is especially true in US-style farside traffic lights. Given the limited receptive field sizes, it is challenging to relate a small red light on the RGB image to a distant rectangular area on the BEV grid. Therefore, we propose specialized solutions for these two.

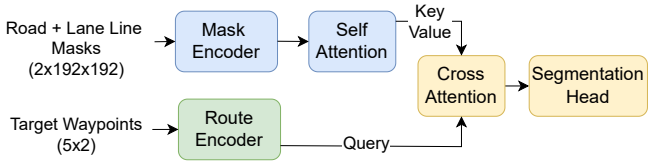

Fig. 4. **Desired Route Generation.** We propose a model to predict the desired route in BEV given road topology in the form of road and line masks and five target waypoints to follow.

TABLE II

BEV PERCEPTION RESULTS. WE SHOW THE RESULTS OF THE MODIFIED SIMPLEBEV [4] TRAINED TO SEGMENT ALL 4 CLASSES AND ONLY 2 STATIC CLASSES. WE REPORT THE IOU VALUES FOR EACH CLASS ON THE VALIDATION SET.

Model	Road	Lane Lines	Vehicle	Pedestrian
All	0.788	0.132	0.403	0.013
Static	0.924	0.756	-	-

C. Desired Route Generation

The desired route is the shortest path towards the future GNSS coordinates of target waypoints, calculated with access to the internal road topology representation of CARLA. If access to the simulator internals is out of the question, as in the case of sensorimotor agents, generating this path becomes a novel problem. To the best of our knowledge, the critical importance of the desired route component has not been studied before. In this work, we show its importance for driving and propose an approach to generate the desired route from BEV.

To predict the desired route, we propose a model that takes information regarding the surrounding road topology, and the target waypoints as input. For the first input, we assume that we have access to the road and lane line masks of the surrounding area in BEV. For a completely unprivileged agent, the road topology also needs to be predicted from raw RGB images but we leave it as future work. For the second input, each agent on CARLA is provided with target waypoints to follow in the form of a list of GNSS coordinates where each coordinate is a vector containing longitude, latitude, and altitude. Following the convention, we first convert these GNSS coordinates into relative coordinates with respect to the ego vehicle's frame of reference using the GNSS and IMU sensors on the vehicle. After converting coordinates to relative target vectors we use the next five waypoints as input to the model.

We encode these two sources of information with separate encoders and fuse information from encoded features with a cross-attention layer as shown in Fig. 4. We encode the road and lane line masks using convolutional layers, tokenize the resulting feature map, and apply self-attention, following vision transformers [20]. We encode the target waypoints using a simple MLP and use the encoded waypoints as the query in the cross-attention layer to generate a BEV representation of the desired route conditioned on the target waypoints to follow. Finally, we reshape the tokenized BEV representation to create a two-dimensional feature map and process it with a segmentation head to predict the mask corresponding to the desired route mask. We train the model using binary cross-entropy loss.

D. Stop Zone Prediction

The final component we replace in the privileged BEV representation is the stop zones due to traffic lights. While object detection methods can be used to detect the presence and status of traffic lights in the scene, what matters for driving is the road region that is affected by the traffic light, so-called the *stop zone*, rather than the position of the traffic light on the image. Moreover, not every detected traffic light affects the ego-vehicle, as there might be traffic lights signaling other vehicles that are visible from the viewpoint of the ego-vehicle.

We propose to predict whether the ego vehicle is currently inside an active stop zone or not. Rather than learning to segment the stop-zones in BEV, we predict a binary variable to inform the agent when it must stop for a red light. To that end, we use a simple off-the-shelf EfficientNet-B0 [21] with a binary classification head. The model takes a single RGB input from a frontal camera. For training, we use a binary cross-entropy loss with a positive weighting to address the scarcity of red light examples in the collected data. During rollouts in the simulator, we apply a threshold of 0.4 to the model's predictions. We observe that this simple approach can effectively predict whether the vehicle is currently affected by a red light for both the European and US-style traffic lights.

IV. EVALUATION

A. Experimental Setup

Camera Setup: Following state-of-the-art BEV methods on CARLA such as Transfuser [22] and NEAT [6], we use 3 cameras and restrict the BEV prediction to the area ahead of the vehicle. The cameras are positioned on the front of the vehicle, one of them facing the front and two of them facing 60 degrees to the left and right. Each camera has a field of view of 100 degrees and generates images of resolution 320×160. Our predicted BEV maps cover the same area as the ROACH [2].

Data Collection: We collect 20 hours of driving data at 10 Hertz using the CARLA autopilot in the ROACH RL environment. We collect data from towns 1 through 6 of the CARLA simulator, reserving 10% of the data for validation. We apply triangular perturbations to the expert agent's actions in order to reduce the effects of distribution shift, following the conventional approach to augment the expert trajectories first proposed by Codevilla et al. [23]. The weather, time of day, and lighting conditions are randomly selected and change dynamically during episodes. We use the same dataset for BEV perception, desired route generation, and stop zone prediction.

Training Details: We train BEV perception for 50,000 steps using the 1-cycle learning rate scheduler proposed by

TABLE III

DRIVING PERFORMANCE WITH BEV PERCEPTION WE COMPARE THE PERFORMANCE OF THE RL AGENT WHILE USING ALL PRIVILEGED INFORMATION (FIRST ROW) TO THE LESS PRIVILEGED VERSION (SECOND ROW) BY REPLACING THE STATIC PART WITH PREDICTIONS.

Predicted	DS	RC	IS
None (Expert)	0.778 ± 0.192	0.927 ± 0.159	0.785 ± 0.089
Static	0.729 ± 0.196	0.844 ± 0.228	0.813 ± 0.011

Smith et al. [24]. We apply a positive weighting factor of 15 for pedestrians, 8 for lane lines, and 5 for vehicles. We train the desired route generation model for 40 epochs, using binary cross-entropy loss with the Adam [25] optimizer and a learning rate of 0.001. We train the stop zone prediction model similarly using binary cross-entropy loss and Adam, but for 10 epochs with a learning rate of 0.0001.

Details of RL Agents: For all our experiments where an RL agent is trained, we use the same environment and the same training scheme as the RL expert of ROACH. We leave the deep learning architecture of the RL agent untouched except for the initial layers to accommodate different input sizes. As RL agents can have considerable variations in performance depending on the random seed, we report the mean and standard deviation over 3 runs.

B. Experimental Results

1) BEV Perception Results: With modified SimpleBEV, our initial goal was to segment the road, lane line, vehicle, and pedestrian classes. We report the results for each of these 4 classes in terms of intersection-over-union (IoU) in the first row (All) of Table II. These results show that while the model can learn to segment the frequent classes such as roads and vehicles, it can not segment less frequent classes such as pedestrians and lane lines. This reveals a shortcoming of state-of-the-art BEV perception models in the face of severe data imbalance that cannot be fixed with a simple positive weighting strategy. Considering the vast literature on object detection in computer vision, we conjecture that specialized architectures can be deployed for object detection. Omitting the vehicles and pedestrians, we train SimpleBEV to segment only the static parts of the scene, i.e. roads and lane lines. As can be seen in the second row (Static) of Table II, this significantly improves the performance of both classes, especially the lane lines.

2) Driving with Predicted BEV: We test the performance of a privileged RL agent by replacing the ground truth road and lane line segmentations with predicted ones from BEV perception. For everything else, we use ground truth information including objects, the desired route, and stop zones. We report the mean and the standard deviation over three runs in Table III. Without requiring any fine-tuning, the results are surprisingly close to the privileged agent.

Discussion: We inspected the driving behavior along with predicted BEV visualizations qualitatively and acquired two critical insights.

TABLE IV

IMPORTANCE OF DESIRED ROUTE. REMOVING THE ROAD AND LANE LINE INFORMATION FROM THE INPUT (W/O STATIC) ALLOWS THE AGENT TO FOCUS ON THE DESIRED ROUTE AND IMPROVES ITS PERFORMANCE. REPLACING THE DESIRED ROUTE INFORMATION WITH HEATMAP REPRESENTATION (W/ TARGET HEATMAP) CAUSES THE AGENT TO FAIL.

Model	DS	RC	IS
Expert	0.778 ± 0.192	0.927 ± 0.159	0.785 ± 0.089
w/o Static	0.868 ± 0.119	1.109 ± 0.126	0.768 ± 0.070
Target Heatmap	0.018 ± 0.012	0.027 ± 0.011	0.858 ± 0.041
Predicted Route	0.484	0.574	0.883

First, while BEV perception models can reach very high IoU scores on static datasets, as in our generated dataset or the NuScenes [19], that performance does not always map to successful predictions during online RL training. As the RL agent explores, it ends up visiting previously unseen state-action pairs, especially in the initial phases of training, prior to achieving successful driving. Unseen states cause our BEV prediction model to fail much more severely than it did on our static validation set. This problem persists even when we apply data augmentation techniques that are commonly used in BEV segmentation. We believe that future work on BEV perception for autonomous driving should consider driving performance with online metrics in addition to segmentation metrics, as the two are rarely correlated [26].

The second critical insight that we acquired is related to our motivation to focus on the desired route. We realized that the agent is able to drive even when the predictions for the road and lane lines are critically low quality. We suspect that this is caused by the dependency of the agent on the desired route channel by ignoring road and lane line channels. Next, we investigate this claim and its repercussions.

3) Importance of Desired Route: We perform an experiment to confirm the dependency of the ROACH expert on the desired route component while ignoring road and lane line information. We first train an RL agent by removing the road and lane line channels from the input. This agent needs to learn to navigate by using only the desired route channel for road information. Second, we replace the desired route channel with a less informative but unprivileged alternative. We project the GNSS coordinates of the next target waypoint of the current route trajectory and render it as a heatmap of the target location. We train another RL agent with this heatmap representation instead of the privileged route.

We compare the driving performance of these two agents to the privileged agent in Table IV. These results confirm our suspicion regarding the importance of the desired route, as the agent learns faster and achieves higher driving performance with less variance when it only sees the desired route (w/o Static). Moreover, we see that a less privileged route representation (w/ Target Heatmap) causes the agent to fail despite ground truth information for everything else.

4) Route Prediction Results: We train an RL agent by replacing the desired route channel with the predicted route. We found that the model requires longer training (20M vs. 10M in ROACH) with the predicted route due to the

TABLE V

STOP ZONE RESULTS. PERFORMANCE OF RL AGENTS WITH DIFFERENT STOP ZONE REPRESENTATIONS.

Model	DS	RC	IS
Expert	0.778 ± 0.192	0.927 ± 0.159	0.785 ± 0.089
GT Binary	0.747 ± 0.073	0.908 ± 0.034	0.791 ± 0.069
Predicted Binary	0.659 ± 0.141	0.779 ± 0.093	0.892 ± 0.071

prediction errors. The driving results are shown in the last row of Table IV. Due to longer training, we cannot report the results over 3 runs. While there is a drop in performance compared to the privileged agent, the agent can learn a meaningful driving behavior with the predicted route. This is an important step towards developing RL agents that can learn to drive with less privileged information.

5) Stop Zone Prediction Results: We explore various ways of incorporating binary stop zone prediction into the input. We first remove the stop zone channel from the BEV input and add a binary variable to the measurements. with this straightforward approach, the agent performs poorly and fails to achieve a driving score over 0.05. We suspect that the low-dimensional measurement vector is ignored by the agent relying on the BEV representation for the most part. To test this, we replace the ground truth stop zone channel with a binary channel, which is set to all ones when the agent is affected by a red light. The experiment results are shown in Table V. We first test this new representation by using ground truth information from the simulator to determine if the agent is affected by a red light (GT Binary). We see that replacing the stop zone channel with a ground truth binary channel results in only a modest drop in performance, showing that the new representation is safe. The agent that uses predictions to fill the binary traffic light channel (Predicted Binary) still manages to achieve a good driving performance albeit with a drop compared to the expert.

V. DISCUSSION AND FUTURE WORK

In this paper, we investigated the reasons behind the success of privileged RL agent ROACH and addressed the challenges of replicating that success with a sensorimotor agent to bridge the gap between the two.

We first adapted a state-of-the-art BEV perception model, SimpleBEV, to efficiently output multiple classes on CARLA. Our evaluation showed impressive results for the static part of the scene but failed for small objects like pedestrians. However, we observed that the successful validation performance of the static part did not generalize to the out-of-distribution observations encountered in the online RL training, yet the agent was successful. Further investigation revealed a more important factor in the success of the RL agent: the desired route component. We then proposed two alternatives to replace the privileged desired route information. The heatmap representation failed but predicting the desired route from road topology and waypoints showed promising results.

We hope that our initial investigation in this paper, related to desired route prediction will lead to future research on better predictions of the desired route, ideally from raw images. This way, we can provide the agent with a better understanding of the path to follow. Without the need to plan a short-term path, the agent can focus on solving other aspects of driving such as infractions.

We also investigated whether a privileged representation of the "stop zone" areas affected by traffic lights is necessary for the success of RL agents. We were able to replace the privileged stop zone region channel with a binary traffic light detector. Incorporating the output of this detector directly as another measurement did not work. However, providing it as an entire channel in the BEV state, either complete zeros or ones, resulted in a close to expert driving performance. Our investigation led to an unprivileged representation of the stop zone that is still acceptable for driving.

We did not investigate vehicle and pedestrian related information and kept them as privileged throughout our work. Predicting pedestrians in the BEV space remains an open challenge. As an alternative, object detection methods can be used to detect these objects, which can then be projected to the BEV space. Another direction is to improve the computational efficiency of larger BEV methods like BEVFormer and incorporate it in RL.

On the other hand, even when these problems are solved on static datasets, offline task metrics do not directly translate to good driving behavior for the agent. There is a significant mismatch in the observed states between a driving dataset like NuScenes, where accidents or wild maneuvers are rare, and an online RL agent exploring different state-action pairs in the environment. This results in BEV predictions failing when the RL agent strays from paths available in the training dataset during policy rollouts. A challenging benchmark to test the robustness of BEV perception methods under unusual configurations could encourage the community to work on this misalignment issue.

Although RL agents fall behind behavior cloning agents on CARLA, we argue that there are still discoveries to be made. Even though our work does not present a state-of-the-art sensorimotor RL agent, it investigates why such an agent has not been discovered yet. We put forth the importance of desired route prediction for RL. We also pinpoint the needs of autonomous driving agents from BEV perception and highlight areas that need improvement. Computational efficiency constitutes a bottleneck in benefiting from the latest developments in computer vision. The distribution differences between offline BEV datasets and online RL training presents another challenge.

Overall, our results highlight the critical role of efficient, informative, and accurate state representations in handling complex driving environments. We argue that such representations hold the key to the discovery of successful sensorimotor RL agents for autonomous driving.

ACKNOWLEDGEMENTS

Ege Onat Özsüer is supported by the KUIS AI Center Fellowship.

REFERENCES

[1] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proc. Conf. on Robot Learning (CoRL)*, 2020.

[2] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[3] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[4] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-bev: What really matters for multi-sensor bev perception?," in *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[5] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[6] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[7] D. Chen, V. Koltun, and P. Krähenbühl, "Learning to drive from a world on rails," in *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. Conf. on Robot Learning (CoRL)*, 2017.

[9] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *arXiv.org*, vol. arXiv:2306.16927, 2023.

[10] H. Kumamoto, K. Tanaka, K. Inoue, and E. J. Henley, "Dagger-sampling monte carlo for system unavailability evaluation," *IEEE Transactions on Reliability*, vol. 29, no. 2, pp. 122–125, 1980.

[11] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger, "Label efficient visual abstractions for autonomous driving," in *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[13] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of the International Conf. on Machine learning (ICML)*, 2016.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] X. Liang, T. Wang, L. Yang, and E. Xing, "CIRL: Controllable imitative reinforcement learning for vision-based self-driving," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning.," in *Proc. of the International Conf. on Learning Representations (ICLR)*, 2016.

[17] M. Toromanoff, E. Wirbel, and F. Moutarde, "End-to-end model-free reinforcement learning for urban driving using implicit affordances," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2018.

[19] 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 *CVPR*, 2020.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. of the International Conf. on Machine learning (ICML)*, 2019.

[22] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[23] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018.

[24] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006, pp. 369–386, SPIE, 2019.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv.org*, vol. arXiv:1412.6980, 2014.

[26] F. Codevilla, A. M. López, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.