Learning by Watching

Jimuyang Zhang Eshed Ohn-Bar
Boston University
{zhangjim, eohnbar}@bu.edu

Abstract

When in a new situation or geographical location, human drivers have an extraordinary ability to watch others and learn maneuvers that they themselves may have never performed. In contrast, existing techniques for learning to drive preclude such a possibility as they assume direct access to an instrumented ego-vehicle with fully known observations and expert driver actions. However, such measurements cannot be directly accessed for the non-ego vehicles when learning by watching others. Therefore, in an application where data is regarded as a highly valuable asset, current approaches completely discard the vast portion of the training data that can be potentially obtained through indirect observation of surrounding vehicles. Motivated by this key insight, we propose the Learning by Watching (LbW) framework which enables learning a driving policy without requiring full knowledge of neither the state nor expert actions. To increase its data, i.e., with new perspectives and maneuvers, LbW makes use of the demonstrations of other vehicles in a given scene by (1) transforming the ego-vehicle's observations to their points of view, and (2) inferring their expert actions. Our LbW agent learns more robust driving policies while enabling data-efficient learning, including quick adaptation of the policy to rare and novel scenarios. In particular, LbW drives robustly even with a fraction of available driving data required by existing methods, achieving an average success rate of 92% on the original CARLA benchmark with only 30 minutes of total driving data and 82% with only 10 minutes.

1. Introduction

Modern autonomous driving systems primarily rely on collecting vast amounts of data through a fleet of instrumented and operated vehicles in order to train imitation and machine learning algorithms [4, 7, 15]. This process generally assumes direct knowledge of the sensory state of an ego-vehicle as well as the control actions of the expert operator. As driving algorithms largely depend on such costly training data to learn to drive safely, it is regarded as a highly valuable asset.

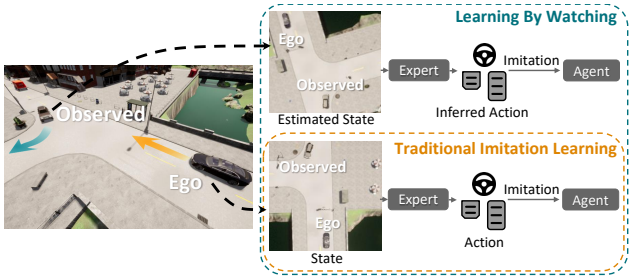


Figure 1: **Learning to Drive by Watching Others.** While existing imitation learning approaches solely utilize data from an ego-vehicle perspective, our proposed Learning by Watching (LbW) framework learns a more robust and data-efficient policy from all available demonstration sources in a scene.

Unfortunately, such data requirements have resulted in self-captured data being amassed in the hands of a few isolated organizations, thereby hindering the progress, accessibility, and utility of autonomous driving technologies. First, while technological giants such as Uber, Tesla, Waymo and other developers may spend significant efforts instrumenting and operating their own vehicles, it is difficult to completely capture all of the driving modes, environments, and events that the real-world presents. Consequently, learned agents may not be able to safely handle diverse cases, e.g., a new scenario or maneuver that is outside of their in-house collected dataset. Second, while small portions of driving data may be shared, i.e., for research purposes, the bulk of it is predominantly kept private due to its underlying worth. Finally, we can also see how this current practice may lead to significant redundancy and thus stagnation in development. ***How can we advance the underlying development process of safe and scalable autonomous vehicles?***

In an application where the lack of a data point could mean the difference between a potential crash or a safe maneuver, we sought to develop a more efficient and shared paradigm to ensure safe driving. Towards this goal, we propose the LbW framework for learning to drive by watching other vehicles in the ego-vehicle's surroundings. Motivated by how human drivers are able to quickly learn from demonstrations provided by other drivers and vehicles, our

approach enables leveraging data in many practical scenarios where ***watched vehicles may not have any instrumentation or direct capture means at all.***

Towards more effective use of a collected dataset, LbW does not assume direct knowledge of either the state or expert actions. For instance, in the scenario shown in Fig. 1, a non-instrumented human-driven vehicle or perhaps another company's autonomous vehicle is observed by our ego-vehicle as they turn and negotiate an intersection. LbW infers the observed agent's state and expert actions so that it may be used to teach our autonomous vehicle. By leveraging supervision from other drivers, our LbW agent can more efficiently learn to drive in varying perspectives and scenarios. The framework facilitates access to large amounts of driving data from human-driven vehicles that may not have been instrumented to directly measure and collect such data.

While offering several benefits for scalability, learning a driving model via indirect means of watching surrounding vehicles in a scene also poses several challenges which we address in this work. Specifically, to advance the state-of-the-art of robust autonomous driving agents, we make the following ***three contributions***: (1) We propose LbW, a new paradigm which can help facilitate a more efficient development of driving agents, thus aiding real-world deployment, (2) we develop an effective two-step behavior cloning approach which infers the states and actions of surrounding vehicles without direct access to such observations, and (3) we validate the impact of LbW on the resulting driving policy through a set of novel experiments on the CARLA and NoCrash benchmarks. While previous approaches tend to exclusively focus on driving policy performance, i.e., requiring many hours of collected ego-vehicle training data, we instead emphasize the benefits of our approach by varying and limiting the amount of available data. We also demonstrate LbW to enable adaption to novel driving scenarios and maneuvers, without ever having direct access to an operator of an ego-vehicle performing such maneuvers.

2. Related Work

Our LbW framework builds on several recent advances in learning to drive, in particular imitation learning from intermediate representations, in order to utilize the demonstrations provided by all expert drivers in a given scene.

Imitation Learning for Autonomous Driving: Over 30 years ago, Pomerleau [55] developed ALVINN, a neural network-based approach for learning to imitate a driver of an ego-vehicle. The approach requires ego-centric camera and laser observations and their corresponding operator steering actions. Since then, more elaborate approaches for learning to drive have emerged [42, 78, 37, 68, 12]. Yet, techniques for imitation learning to drive are still widely employed due to implementation and data collection ease [7, 52, 40, 75], particularly the offline and supervised learning version of behavior cloning [3, 55, 48, 60, 10]. Behavior cloning with conditional input is now a strong baseline for driving in urban settings [47, 42, 14, 15, 51, 11, 58, 57, 24]. However, approaches generally require complete access to the perceptual observations of the vehicle together with the expert driver actions. Building on recent advances in imitation learning to drive, our work takes a step towards extending the traditional formulation so that it may be applicable to challenging cases where direct access to measurements cannot be assumed. Due to the difficulty of our learning by watching task, the approach hinges on the choice of underlying data representation, as discussed next.

Intermediate Representations for Autonomous Driving: In parallel with steady progress in imitation learning to drive, the computer vision and machine learning communities have dedicated significant effort towards advancing perception and prediction tasks for autonomous vehicles. As pioneered by several researchers and developers, notably Dickmanns [19, 21, 20, 18], such tasks, e.g., semantic segmentation [17], 3D tracking of salient on-road objects from sensor input [26, 76, 72, 63, 71, 47, 70, 23, 27, 10], and future predictions [61, 8, 13, 34, 35, 38, 41, 62, 74], can be used in order to efficiently train a driving policy [76, 5]. Of particular relevance to our study are approaches that leverage a Bird's-Eye-View (BEV) of the scene [70, 23, 4, 73, 64, 25]. While researchers may focus on accurately obtaining components of the BEV through a variety of sensor configurations [30, 32, 59, 45, 56, 44, 28, 53, 54, 77, 1], in our work we emphasize employing the BEV as a compact intermediate representation for learning a driving policy. Crucially, a BEV enables to efficiently transform observations between varying points of view and estimate agent-centric states when learning by watching. In contrast, transforming other types of representations across views, e.g., [16, 46, 77], can be difficult for significantly differing perspectives.

Observational and Third-Person Imitation Learning: Several related recent studies tackle more general cases of imitation learning. For instance, methods may only require an observation of the state, such as video footage, without the underlying demonstrator actions. Kumar et al. [39] learns an inverse model that can generate pseudo-labels of actions for observed videos, and Murali et al. [49] explores a similar learning by observation formulation for surgical tasks. Nait et al. [50] presents a learning-based system using a pixel-level inverse dynamics model, i.e., for inferring actions that were taken by the human manipulator given a sequence of monocular images. However, such approaches still require access to the underlying state of the human expert, whereas we tackle the more general settings where direct access to neither the state nor the operator's actions is assumed. Another recent line of research analyzes third-

person imitation learning [65, 67], i.e., translating policies across slightly differing views of a shared workspace. Liu et al. [43] present an imitation learning method based on a context translation model that can convert a demonstration from a third-person viewpoint (i.e., human demonstrator) to the first-person viewpoint (i.e., robot). However, these recent studies focus on toy settings and simplified visual environments, e.g., an inverted pendulum control task [67]. In contrast, we are concerned with intricate multi-agent settings in the context of autonomous driving. Here, our task involves transforming and estimating demonstration data across drastically varying perspectives, frequent occlusion, dynamic scenes, and noisy agent tracking.

3. Method

In this section, we describe our LbW approach. We first define the learning setting as it has some key differences compared to existing approaches for imitation learning to drive (Section 3.1). Next, we describe our method for imitating the driving policies of not only our own instrumented and operated ego-vehicle, but of the surrounding vehicles as well. Our method consists of (1) an agent-centric input state representation that can be efficiently transformed to an observed agent frame of reference (Section 3.2), (2) a visibility encoding for handling missing state information due to occlusion and significantly differing viewpoints (Section 3.2), (3) a waypoint-based decomposition of the action representation aimed to efficiently learn from surrounding drivers (Section 3.3), and (4) a loss adjustment module for selectively refurbishing difficult and noisy samples inherent to the LbW task (Section 3.3).

3.1. Problem Definition

Imitation Learning Formulation: The objective of our driving agent is to generate a control action $\mathbf{a}_t \in \mathcal{A} = [-1, 1] \times [0, 1]^2$, i.e., the steering, throttle, and brake at each time step t, in order to arrive to a predefined destination. The input to the agent is the current state $\mathbf{s}_t \in \mathcal{S}$, which comprises of information about the vehicle and its surrounding context. The goal of the learning process is to learn a mapping

$$\pi_{\boldsymbol{\theta}} : \mathcal{S} \rightarrow \mathcal{A} \tag{1}$$

parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, also known as the policy function. In this work, we are concerned with learning $\pi_{\boldsymbol{\theta}}$ from driver demonstrations. Traditionally, the process of imitation learning [52] assumes access to the state and expert of an operated ego-vehicle, i.e., trajectories of the form

$$\boldsymbol{\tau}^{expert} = \{(\mathbf{s}_t, \mathbf{a}_t^*)\}_{t=1}^{T} \tag{2}$$

where $\mathbf{a}_t^*$ are expert actions. The trajectories of directly experienced states and corresponding demonstrator actions are then used to learn a policy. The goal of the policy is to produce driving behavior that is similar to the behavior of the demonstrator.

Imitation Learning by Watching: Despite ample related work studying our general decision-making problem, training LbW agents poses several unique challenges. Specifically, as in our case there is no direct access to either the state or action information of the watched agent, representing and inferring such information becomes a fundamental challenge[1]. While surrounding vehicles are also driven by expert drivers, their trajectories are perceived indirectly from the viewpoint of the ego-vehicle. In this case, the underlying state and demonstrator actions must therefore be estimated, such that

$$\boldsymbol{\tau}^{expert\text{-}LbW} = \{(\hat{\mathbf{s}}_t, \hat{\mathbf{a}}_t^*)\}_{t=1}^{T} \tag{3}$$

Due to the estimation task, training LbW agents is a very challenging problem. Specifically, such trajectory data must be properly represented, transformed, and inferred to avoid hindering driving performance. Next, we discuss how to effectively learn to drive from such potentially incomplete and noisy data.

3.2. Estimating Agent-Centric States

Estimating the agent-centric states of surrounding vehicles is key to using their demonstrations in Eqn. 3 to learn by watching. The complete state information is denoted as

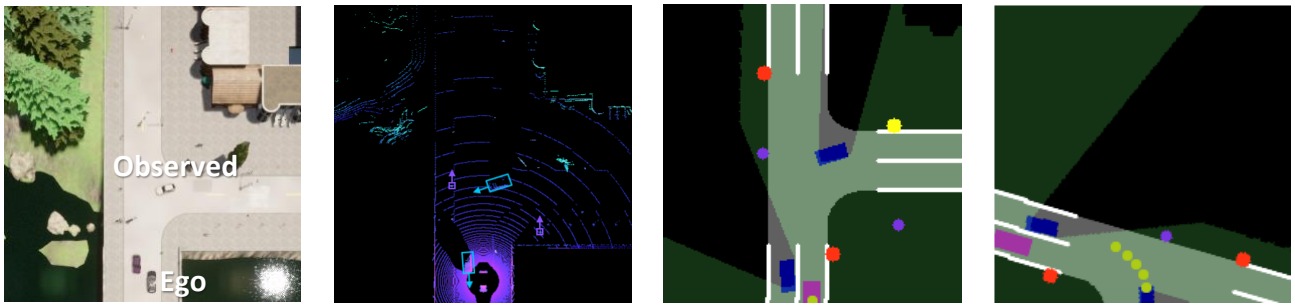
$$\mathbf{s}_t = [\mathbf{B}_t, \mathbf{M}_t, v_t, c_t] \tag{4}$$

where $\mathbf{B}_t$ is the current Bird's-Eye-View (BEV) image, $\mathbf{M}_t$ is a visibility map of the scene computed from the viewpoint of the ego-vehicle, v_t is the current speed, and c_t is a categorical variable specifying a high-level navigation command following the goal-driven CARLA navigation task [14, 11]. We further discuss these variables below.

Bird's-Eye-View State Representation: We propose to leverage a BEV image of the environment as it provides a compact and intuitive intermediate representation for learning a driving policy. The BEV representation will be crucial for addressing issues with differing viewpoints and missing observations when learning by watching.

As shown in Fig. 2, our BEV is a rendered tensor $\mathbf{B} \in \{0, 1\}^{W \times H \times 7}$ of a top-down view as perceived by the agent. The grid representation encodes each object type in a distinct channel, including the position of the road, lane marks, pedestrians, vehicles, and traffic lights and their state, i.e., red, yellow or green. The viewer position is fixed

[1]To clarify, it is only during LbW agent training that we have proxy and partial information regarding the states and expert actions of other agents. Once our LbW agent is trained, test-time driving is identical to the an agent learned via traditional ego-centric imitation learning, i.e., the state is directly observed.



(a) Reference scene (b) LiDAR map (c) Ego-centric BEV (d) Estimated BEV

Figure 2: **Bird's-Eye-View (BEV) Representation With Visibility Map Overlay.** We visualize (a) the reference top-down view of a scene, (b) its corresponding LiDAR map with vehicle and pedestrian detections, (c) the ego-centric BEV after integration with map information, including the ego-vehicle (magenta), surrounding vehicles (blue), pedestrians (purple circles), traffic lights (yellow, red), and visibility map (green overlay), and (d) the estimated BEV from the perspective of the observed vehicle. Dark yellow circles show future waypoints (Section 3.3) for the ego and observed vehicle.

at the BEV's middle-bottom portion with a perpendicular orientation to the x-axis. Further details regarding BEV computation can be found in the supplementary.

While there has been significant work in obtaining BEV representations from a variety of sensor setups and maps, our work primarily focuses on leveraging such representations, in particular [72], in order to learn a robust policy. We focus on the most essential state information for the BEV, such that other algorithms and more sophisticated representations, e.g., for agent dynamics and temporal context [30, 2, 9] can be easily integrated to extend our analysis in the future.

Observed Agent-Centric BEV: Observed drivers are reacting to the surroundings as perceived from their own perspective. The benefit of using a BEV representation, i.e., compared to low-level sensor information, is the crucial simplification it offers when estimating states from the perspective of other vehicles.

To obtain an agent-centric BEV, we employ the 3D location $\mathbf{l}$ and orientation α of an observed agent [72]. Next, any point $\mathbf{x} \in \mathbf{B}$ can be transformed through rotation and translation to an agent-centric frame of reference

$$\mathbf{x}_{\hat{\mathbf{B}}} = \mathbf{R}_{ego}^{observed}(\alpha)\mathbf{T}_{ego}^{observed}(\mathbf{l})\mathbf{x}_{\mathbf{B}} \tag{5}$$

where $\mathbf{x}$, written in homogeneous coordinates, is transformed to a new origin according to a rotation matrix calculated from α and a translation matrix calculated from l. This process produces a BEV for an observed vehicle $\hat{\mathbf{B}}$.

In addition to the agent-centric BEV, we estimate the speed and high-level command of watched agents ($\hat{v}$, $\hat{c}$) by tracking and comparing their 3D position over time, as observed from the ego-vehicle perspective. We analyze motion of observed vehicles relative to their entrance position to an intersection in order to obtain $\hat{c}$.

Visibility Map: Our choice of a BEV state representation facilitates a more efficient learning by watching task. However, in practice, the viewpoint transformation in Eqn. 5 can only roughly account for the state as it would have been perceived from the perspective of another vehicle. Due to occlusion and differing viewpoints in cases of complex urban settings, it is likely for the BEV estimation process to miss critical scene components. While missing scene context that could be perceived from a different perspective is a frequent issue in our experiments, Fig 3 demonstrates a more severe example. Here, a stopped vehicle drastically limits the BEV of the ego-vehicle. This results in a poor estimate of the BEV for the observed vehicle, i.e., entirely missing an unobserved vehicle which greatly influences actions by the driver in the observed vehicle. In general, cases of occlusion can produce trajectories with various driving actions that cannot be well-explained by our estimate of the agent-centric BEV $\hat{\mathbf{B}}$.

We propose to alleviate this critical issue of potentially missing information by utilizing a visibility map representation [31, 33]. The visibility map, as shown in Figs. 2 and 3, encodes areas in the BEV that were not directly perceived by the ego-vehicle. In this manner, the LbW agent can reason over crucial cues of unavailable information and learn to leverage the estimated BEV effectively during dense driving scenarios. While visibility and occupancy maps have been extensively studied in the general context of autonomous driving, its impact on learning policies from surrounding vehicles in complex, multi-agent settings with missing state information has not been previously analyzed. Moreover, it is important to note that we always compute the visibility map $\mathbf{M}$ via ray-casting in the perspective of the ego-vehicle. This step ensures that samples with low visibility in the *original frame of reference* are properly handled during LbW training. Eqn. 5 is then used to obtain an agent-centric visibility map $\hat{\mathbf{M}}$.

3.3. Inferring Demonstrator Actions

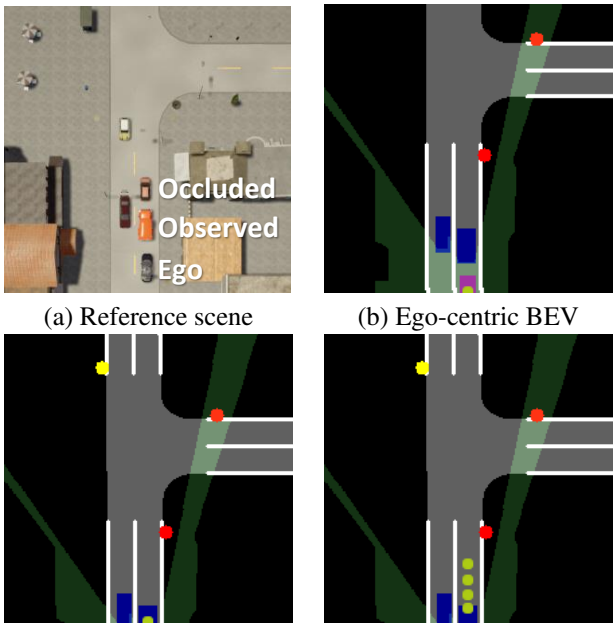
When learning by watching, our goal is to leverage demonstrations provided by other drivers as additional data for training the policy function $\pi_{\boldsymbol{\theta}}$. However, the expert actions taken by drivers of other vehicles must first be inferred to be used as learning targets for behavior cloning approaches [14, 15]. Nonetheless, inferring the steering, throttle, and brake control without having direct observation of such measurements is difficult. Therefore, as in the previous task of estimating agent-centric states, the LbW task hinges on the choice of a suitable underlying action representation. To effectively learn by watching, we propose to use a waypoint-based decomposition of the policy function.

Waypoint-Based Action Representation: The motion of watched vehicles as they are being driven alludes to the underlying operator actions. Motivated by this insight, our approach for inferring the actions of watched vehicles employs a representation based on waypoints. Each vehicle's observed motion can be unambiguously defined in terms of visited waypoints, i.e., a sequence of traversed positions by the expert in the BEV over time $\mathbf{w}^* = \{\mathbf{w}_0^*, \mathbf{w}_1^*, ..., \mathbf{w}_K^*\}$ [11]. During testing of the agent, the waypoints can then be passed as targets to a low-level controller g, e.g., a PID controller [69, 11]. The controller can then produce the final policy, such that

$$\pi_{\boldsymbol{\theta}}(\mathbf{s}) = g(f_{\boldsymbol{\theta}}(\mathbf{s})) \tag{6}$$

where $f_{\boldsymbol{\theta}}(\mathbf{s})$ is a learned waypoint prediction function. We do not use a parameter notation for g as it does not include trainable parameters in our implementation. This decomposition can also simplify the overall policy learning task [37, 47]. While we can use a control model to infer low-level actions and use these as policy output targets directly, in practice we found this task to be ill-posed. Instead, the high-level waypoint-based action representation provides a standardized framework for combining action data from multiple different agents. It also enables the explicit handling of incomplete and noisy data inherent to the inferred experts, as discussed next.

Waypoint Refurbishment for Observed Vehicles: One of the main challenges of using the indirectly observed training data is the ability to learn from difficult samples with missing or noisy information. For instance, the observed waypoints may contain significant noise due to the underlying sensor setup and perception algorithm in Section 3.2. Moreover, even in cases where the waypoints are observed accurately, the lack of state information to sufficiently explain their trajectory could potentially lead to ambiguity during agent training. In the visualized example of Fig. 3, the observed waypoints will not be fully explained by the available state information, i.e., expressing an inferred action to brake. Instead, the waypoints may vary arbitrarily



(a) Reference scene (b) Ego-centric BEV

(c) Estimated BEV (d) Refurbished waypoints

Figure 3: **Refurbishing Difficult Samples.** The example estimated view for an observed vehicle, shown in (c), is ambiguous due to occlusion. The proposed refurbishment process (Section 3.3) can correct the waypoint targets (d) to align with the estimated state and encourage proper agent behavior, i.e., to move forward and maintain a closer distance to the red light.

despite an identical estimated state. Although the presence of numerous difficult and noisy samples could hinder policy training (Section 3.4), discarding samples can remove potentially useful training data [66].

To take full advantage of the observed demonstrations, we employ a sample refurbishment process. We modify observed waypoints using a confidence hyper-parameter $\beta \in [0, 1]$ for determining the amount of refurbishment, i.e.,

$$\hat{\mathbf{w}}^* = \beta \mathbf{w}^* + (1 - \beta)\hat{\mathbf{w}} \tag{7}$$

where $\mathbf{w}^*$ are the observed waypoints and $\hat{\mathbf{w}}$ are correction values. β is used to specify the reliability of the waypoint extraction step. As discussed further in Section 3.4, the correction labels $\hat{\mathbf{w}}$ are predicted from the estimated state $\hat{\mathbf{s}}$ by an ego-only baseline model as it is trained using clean data, i.e., without any LbW data. In this manner, we can further mitigate noise and prevent overfitting to ambiguous or incorrect waypoint labels, thereby reducing their negative impact during training. We study the benefits of waypoint refurbishment for robust training with difficult cases of occlusion as well as overall perception noise in Section 4.

3.4. Training

Dataset: Motivated by the success of behavior cloning algorithms, e.g., [15, 14, 11], we pursue a supervised learning approach for learning from driver demonstrations. During training, as in previous work, we assume a collection of driver demonstrations comprising of sequences of ego-vehicle trajectories, $\mathbf{D}^*_{ego} = \{\boldsymbol{\tau}_i^{expert}\}_{i=1}^{N}$. Differently from previous work, the proposed LbW method can then be used to extract *new training data from the original ego-vehicle trajectories*. By leveraging human drivers which are operating the surrounding vehicles as expert demonstrators, we generate an additional trajectory for each agent j that is observed throughout a drive i such that $\mathbf{D}^*_{LbW} = \{\hat{\boldsymbol{\tau}}_{i,j}^{expert\text{-}LbW}\}_{j=1}^{M}$, where $\hat{\boldsymbol{\tau}}$ denotes refurbished trajectories. By incorporating the additional trajectories, our method enables learning robust policies with much smaller amounts of total collected driving time.

Robust Policy Learning: We learn a prediction function $f_{\boldsymbol{\theta}}$ by minimizing a waypoint-based behavior cloning (WBC) loss over samples from a dataset $\mathbf{D}$

$$\mathcal{L}_{\text{WBC}} = \mathbb{E}_{(\mathbf{s},\mathbf{w})\sim\mathbf{D}}\left[\ell_1(\mathbf{w}, f_{\boldsymbol{\theta}}(\mathbf{s}))\right] \tag{8}$$

The ℓ_1-loss is computed over the next K target waypoints that the autonomous vehicle should learn to predict.

We train a robust LbW policy using Eqn. 8 in three steps. First, we leverage $\mathbf{D}^*_{ego}$ and train a baseline behavior cloning model, $f_{\hat{\boldsymbol{\theta}}}(\mathbf{s})$. In general, we assume that samples in $\mathbf{D}^*_{ego}$ provide correct and clean supervision, as they were obtained through direct observation of driver actions. We then use the learned ego-only model to refurbish the LbW samples and obtain $\mathbf{D}^*_{LbW}$, i.e., by setting $\hat{\mathbf{w}} = f_{\hat{\boldsymbol{\theta}}}(\mathbf{s})$ in Eqn. 7. Thus, leveraging the baseline model to correct observed waypoints provides an explicit mechanism for removing overall noise and mislabeled samples. The refurbishment mechanism also aids in standardizing demonstrations across the various observed actors. Finally, the full model $f_{\boldsymbol{\theta}}$ is trained from scratch over the complete dataset $\mathbf{D}^* = \mathbf{D}^*_{ego} \cup \mathbf{D}^*_{LbW}$. In this final step, the refurbished waypoint targets improve robustness by effectively adjusting the loss in Eqn. 8 [66]. During testing, we use our learned function $f_{\boldsymbol{\theta}}$ to map states to waypoints. The waypoints are then given to the PID controller to obtain the policy $\pi_{\boldsymbol{\theta}}$, i.e., as an inverse dynamics model for mapping waypoints to actions.

3.5. Implementation Details

We implement $f_{\boldsymbol{\theta}}$ as a convolutional neural network with trainable parameters $\boldsymbol{\theta}$. We follow Chen et al. [11] and train a state-of-the-art BEV-based conditional behavior cloning agent (see supplementary for details).

Visibility Map Integration: We are interested in analyzing the impact of integrating the visibility with the BEV on policy learning as means to encode missing state information and reason over potentially ambiguous corresponding waypoint labels. As visibility maps have not been thoroughly studied in this specific context, we explore two network structures for integrating visibility cues, early and late fusion. In the early fusion scheme, the visibility map is leveraged as another channel concatenated to the BEV prior as an input to the backbone. However, the type of information encoded in the visibility map significantly differs in its purpose from the type of object position information that is found in the BEV. Therefore, we also experiment with a late fusion scheme, with a separate backbone for the visibility map. Both fusion schemes enable the network to weigh the two sources of information as needed, i.e., when waypoints are poorly explained by the BEV alone.

4. Experiments

We use the 0.9.9 version of the CARLA simulator [22] for generating diverse multi-agent driving scenes and evaluating the proposed LbW framework.

Original and NoCrash CARLA Benchmarks: The CARLA benchmarks [22, 14, 15] employ Town 1 of the simulation for training and Town 2 for testing under varying navigation tasks. The original benchmark is now obsolete [51, 11], yet previous analysis used older versions of the simulation (0.8.4-0.9.6). To analyze performance under more recently introduced functionalities, e.g., complex pedestrian behavior and changed graphics, we keep this benchmark for reference and only use the most challenging task of dynamic navigation settings [22]. We also consider the more challenging NoCrash benchmark settings [15], which employ various traffic density conditions (e.g., regular, dense) while not allowing for collisions. Previous approaches on CARLA generally require large amounts of training data, specified as ego-vehicle driving hours, e.g., between five hours in [11] to 10 hours in [15]. In contrast, our proposed method can effectively utilize much smaller datasets. Therefore, we focus on much lower total driving time of between 10 minutes and up to one hour. This experimental setup also ensures that our method is beneficial to low-data settings, e.g., for rare maneuvers.

Adaptation Benchmark: To further analyze the data-efficiency limitations of our approach, we also introduce a new benchmark defined by training over Town 1 and adapting the model to the most difficult town in the simulator, Town 3. The commonly used first two towns are relatively small and are comprised of similar features, e.g., simple two-lane roads with only orthogonal three-way intersections. In contrast, Town 3 has diverse multi-lane roads and intersections, with roundabouts, five-way intersections, and diagonal and curved turns. For adaptation, we define a disjoint set of maneuvers for training and testing and collect a small amount of data (a total of 10 minutes driving time) to

Table 1: **Ablation Study.** Comparison of driving success rate (%) for the proposed approach (LbW) with various visibility fusion schemes. The baseline model (Ego) is trained by traditional behavior cloning. Mean and standard deviation are shown over three runs using the original CARLA benchmark (OB) and the NoCrash benchmark (Regular: NC-R, Dense: NC-D).

	One Hour			**30 Minutes**			**10 Minutes**		
	NC-R	NC-D	OB	NC-R	NC-D	OB	NC-R	NC-D	OB
Ego (Baseline)	46 ± 1	18 ± 1	56 ± 1	26 ± 2	12 ± 1	68 ± 1	24 ± 1	0 ± 0	64 ± 1
LbW	64 ± 1	24 ± 1	74 ± 1	52 ± 0	$\mathbf{24 \pm 0}$	68 ± 1	34 ± 1	6 ± 1	$\mathbf{82 \pm 4}$
LbW + Visibility (Early)	52 ± 1	$\mathbf{24 \pm 0}$	76 ± 1	54 ± 1	18 ± 1	72 ± 3	28 ± 1	6 ± 1	64 ± 1
LbW + Visibility (Late)	$\mathbf{92 \pm 3}$	$\mathbf{24 \pm 0}$	$\mathbf{92 \pm 1}$	$\mathbf{74 \pm 2}$	$\mathbf{24 \pm 0}$	$\mathbf{92 \pm 0}$	$\mathbf{52 \pm 1}$	$\mathbf{20 \pm 2}$	68 ± 1

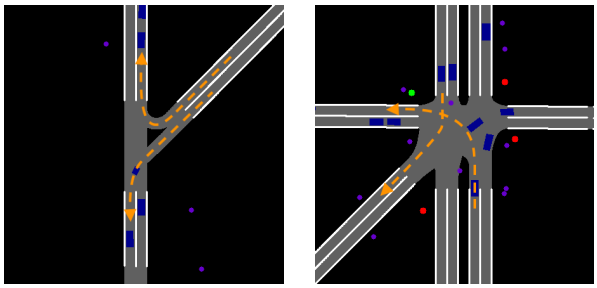


Figure 4: **Adaptation Benchmark.** We visualize example intersections from Town 3 together with agent testing routes (not seen in training) in orange arrows.

be used for adapting the model trained in Town 1. Example test routes are shown in Fig. 4. We define a CARLA and NoCrash-type benchmarks on Town 3 with various driving conditions. These challenging settings aim to analyze a realistic scenario where an autonomous vehicle must quickly learn to operate in a new situation or geographical location.

4.1. Results

We conduct four main experiments to analyze the benefits of different components in our proposed approach. First, we begin by comparing the performance of the LbW agent with the traditional behavior cloning baseline [11]. Second, we analyze the various underlying network structures for fusion of the visibility map. Third, we explore the performance of the LbW agent under a low-data regime. Finally, we investigate the impact of the proposed label refurbishment process when training robust driving agents.

LbW vs. Baseline: We first discuss our results using one hour of total driving data. The analysis in Table 1 shows the driving performance, in terms of success rate, for an agent trained with our proposed approach. Results are shown for both the NoCrash (NC) regular and dense benchmarks as well as the reference original CARLA benchmark (OB). With one hour of training data, we find the proposed LbW approach to consistently improve overall driving performance, i.e., by 39%, 33%, and 32% over the baseline agent for the NC-Regular, NC-Dense, and original CARLA settings, respectively. Notably, the consistent improvements are achieved by the LbW agent prior to integrating the visibility map or refurbishment step. We emphasize that the driving dataset collected for training both the LbW and baseline models is identical. Yet, the baseline model only leverages demonstration data that has been directly observed by the ego-vehicle. In contrast, through effective cross-perspective reasoning, our method increases the available training samples with respect to the number of surrounding agents observed. As such, the baseline's performance can match LbW asymptotically in principle, i.e., through exhaustively driving in various scenes, maneuvers, and perspectives. This is inefficient and costly in practice.

Integrating the Visibility Map: We sought to understand the benefits of different model choices in our learning by watching framework. Here, we analyze the underlying network structure for integrating the visibility map with respect to the ultimate driving task. Our findings with one hour of training data highlight the importance of effectively integrating visibility cues with BEV-based cues. Specifically, we achieve a 92% success rate for the late fusion-based LbW agent compared to 64% obtained by an agent without a visibility map on NC-Regular. We also nearly solve the original CARLA benchmark, achieving a 92% success rate. Early fusion of the visibility map is less successful, as our visibility cues differ in their function compared to the positional BEV cues. Thus, our model benefits from an architecture with distinct processing for the two types of cues. It is useful to contrast this finding with the 3D object detection study of Hu et al. [33], where an early fusion scheme was found to work best. One reason for this discrepancy could be due to the differences in the underlying learning task. In this study, we utilize the visibility cue to encode a more complex notion of missing cross-perspective information when training a driving policy.

Data-Efficiency in Training: Our primary goal is to enable researchers to easily train robust driving agents while minimizing operation and data collection costs. Towards this goal, in this experiment we deliberately consider an often neglected aspect of data-efficiency when training driving policies. Here, to highlight the benefits of our LbW

Table 2: **Adaptation Results.** Quantitative results for policy adaptation to novel maneuvers on Town 3.

	NC-R	NC-D	OB
Ego (Baseline)	40 ± 0	40 ± 0	60 ± 0
LbW	60 ± 1	$\mathbf{60 \pm 0}$	80 ± 1
LbW + Visibility (Early)	60 ± 1	60 ± 1	60 ± 1
LbW + Visibility (Late)	$\mathbf{100 \pm 0}$	40 ± 0	$\mathbf{100 \pm 0}$

(a) Ego (Baseline) (b) LbW (Proposed)

Figure 5: **Qualitative Adaptation Results.** While both agents leverage the same short adaptation route, the LbW agent effectively learns entirely new maneuvers solely through watching them performed by other vehicles. LbW results in more accurate and safe predicted waypoints.

approach, we use a much smaller amount of driving data than is generally considered, e.g., 30 and 10 minutes in total. This is a challenging experiment given that our models are initialized from scratch. Our key finding in Table 1 reveals how ***our LbW agent outperforms the baseline model trained with six times the amount of data***. For instance, for NC-Regular test settings, our agent trained with 10 minutes of data achieves a 52% success rate compares to 46% with traditional behavior cloning and one hour of data.

Data-Efficiency in Adaptation: We further explore the practical limitations of our proposed approach by adapting the driving agent to a town with completely different layout characteristics and maneuvers. In this realistic scenario, the agent encounters an environment differing from its original dataset and must efficiently adapt to the new environment in order to avoid an unsafe maneuver, e.g., as visualized in Fig. 5. Here, even a single trajectory watched at an intersection can potentially teach the ego-vehicle an entirely new maneuver, i.e., turning at a five-way intersection. As not utilizing such critical data would lead to poor driving performance on the most complex CARLA town, these settings provide an ideal test case. Table 2 shows how the baseline model generally struggles to adapt to the new environment. In contrast, our LbW agent can achieve up to 100% success rate on NC-Regular driving settings, solely through watching novel maneuvers being performed by other vehicles. As shown in Fig. 5, the LbW adaptation process results in improved waypoint predictions for the novel maneuvers.

Table 3: **Refurbishment Analysis.** Impact of waypoint refurbishment on driving success rate (%). Results are shown using the late fusion visibility integration scheme.

Town 2	NC-R	NC-D	OB
LbW + Visibility	64 ± 0	32 ± 3	86 ± 1
LbW + Visibility (Refurbishment)	$\mathbf{80 \pm 1}$	$\mathbf{36 \pm 0}$	$\mathbf{96 \pm 0}$
Town 3	**NC-R**	**NC-D**	**OB**
LbW + Visibility	40 ± 0	30 ± 1	$\mathbf{60 \pm 0}$
LbW + Visibility (Refurbishment)	$\mathbf{60 \pm 0}$	$\mathbf{40 \pm 0}$	$\mathbf{60 \pm 0}$

Nonetheless, the conditions on this new town, especially during dense settings, can create drastically different visibility patterns compared to the ones in Town 1. Hence, we find that the adaptation of our LbW agent without visibility cues is more efficient for this difficult task, motivating future work in learning generalized driving policies.

Robust Policy Learning: Given the analysis on the underlying network structure and choice of late fusion scheme, we now quantify the role of waypoint refurbishment when handling mislabeled samples inherent to LbW datasets. In particular, we sought to further analyze our proposed approach on more general settings, for instance, settings with a lower quality sensor, algorithm, or map, as such noise can impact the learning by watching process. To ensure our findings are relevant across various sensor and algorithm configurations, we further add waypoint noise into our one hour dataset when training our models. By improving robustness, i.e., through adjustment of the training loss, we show refurbishment to consistently improve driving success rates in Table 3. Refurbishment leads to further gains of 25%, 13%, and 11% on the NC-Regular, NC-Dense, and original benchmark settings, respectively.

5. Conclusion

In this work, we have presented the learning by watching framework. Based on our experiments, we showed that leveraging supervision from inferred state and actions of surrounding expert drivers can lead to dramatic gains in terms of driving policy performance, generalization, and data-efficiency. While our approach enables learning from incomplete data and noisy demonstrations, an important next step would be further validation under such realistic challenges. For instance, although existing approaches on CARLA generally assume expert optimality, this assumption may not hold in practice when learning to drive by watching diverse and imperfect drivers. The complementarity of the LbW approach to additional sample-efficient imitation learning approaches, e.g., [6, 36, 29], could also be studied in the future. More broadly, we hope that our work motivates others to pursue a more shared and efficient development of autonomous vehicles at scale.

References

[1] Syed Ammar Abbas and Andrew Zisserman. A geometric approach to obtain a bird's eye view from an image. In *ICCVW*, 2019. 2

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 4

[3] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence*, 1996. 2

[4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In *RSS*, 2019. 1, 2

[5] Aseem Behl, Kashyap Chitta, Aditya Prakash, Eshed Ohn-Bar, and Andreas Geiger. Label-efficient visual abstractions for autonomous driving. *IROS*, 2020. 2

[6] Lionel Blondé and Alexandros Kalousis. Sample-efficient imitation learning via generative adversarial nets. In *AISTATS*, 2019. 8

[7] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 1, 2

[8] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *ICRA*, 2020. 2

[9] Sergio Casas, Wenjie Luo, and Raquel Urtasun. IntentNet: Learning to predict intention from raw sensor data. In *CoRL*, 2018. 4

[10] Chenyi Chen, Ari Seff, Alain L. Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 2

[11] Dian Chen, Brady Zhou, and Vladlen Koltun. Learning by cheating. In *CoRL*, 2019. 2, 3, 5, 6, 7

[12] Yilun Chen, Chiyu Dong, Praveen Palanisamy, Priyantha Mudalige, Katharina Muelling, and John M Dolan. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *CVPRW*, 2019. 2

[13] Chiho Choi and Behzad Dariush. Looking to relations for future trajectory forecast. In *ICCV*, 2019. 2

[14] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, 2018. 2, 3, 5, 6

[15] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, 2019. 1, 2, 5, 6

[16] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Nova: Learning to see in novel viewpoints and domains. In *3DV*, 2019. 2

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2

[18] Ernst D Dickmanns. The development of machine vision for road vehicles in the last decade. In *IV*, 2002. 2

[19] Ernst Dieter Dickmanns. *Dynamic vision for perception and control of motion*. Springer Science & Business Media, 2007. 2

[20] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Machine vision and Applications*, 1(4):223–240, 1988. 2

[21] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):199–213, 1992. 2

[22] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017. 6

[23] Sudeep Fadadu, Shreyash Pandey, Darshan Hegde, Yi Shi, Fang-Chieh Chou, Nemanja Djuric, and Carlos Vallespi-Gonzalez. Multi-view fusion of sensor data for improved perception and prediction in autonomous driving. *arXiv preprint arXiv:2008.11901*, 2020. 2

[24] Angelos Filos, Panagiotis Tigas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? *ICML*, 2020. 2

[25] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding HD maps and agent dynamics from vectorized representation. In *CVPR*, 2020. 2

[26] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012. 2

[27] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *ICRA*, 2020. 2

[28] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. *arXiv preprint arXiv:2006.09917*, 2020. 2

[29] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *NeurIPS*, 2016. 8

[30] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *CVPR*, 2019. 2, 4

[31] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013. 4

[32] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020. 2

[33] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3D object detection. In *CVPR*, 2020. 4, 7

[34] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *ICCV*, 2019. 2

[35] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, 2019. 2

[36] Wonseok Jeon, Seokin Seo, and Kee-Eung Kim. A bayesian approach to generative adversarial imitation learning. In *NeurIPS*, 2018. 8

[37] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. 2, 5

[38] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*, 2019. 2

[39] Ashish Kumar, Saurabh Gupta, and Jitendra Malik. Learning navigation subroutines from egocentric videos. In *CoRL*, 2020. 2

[40] Guohao Li, Matthias Mueller, Vincent Casser, Neil Smith, Dominik L Michels, and Bernard Ghanem. Oil: Observational imitation learning. *RSS*, 2019. 2

[41] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. PnPNet: End-to-end perception and prediction with tracking in the loop. In *CVPR*, 2020. 2

[42] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. CIRL: Controllable imitative reinforcement learning for vision-based self-driving. In *ECCV*, 2018. 2

[43] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *ICRA*, 2018. 3

[44] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. *IEEE Robotics and Automation Letters*, 4(2):445–452, 2019. 2

[45] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Sai Shankar Narasimhan, Madhava Krishna, and Krishna Murthy Jatavallabhula. Monolayout: Amodal scene layout from a single image. In *WACV*, 2020. 2

[46] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020. 2

[47] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. In *CoRL*, 2018. 2, 5

[48] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun. Off-road obstacle avoidance through end-to-end learning. In *NeurIPS*, 2006. 2

[49] Adithyavairavan Murali, Siddarth Sen, Ben Kehoe, Animesh Garg, Seth McFarland, Sachin Patil, W Douglas Boyd, Susan Lim, Pieter Abbeel, and Ken Goldberg. Learning by observation for surgical subtasks: Multilateral cutting of 3D viscoelastic and 2D orthotropic tissue phantoms. In *ICRA*, 2015. 2

[50] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *ICRA*, 2017. 2

[51] Eshed Ohn-Bar, Aditya Prakash, Aseem Behl, Kashyap Chitta, and Andreas Geiger. Learning situational driving. In *CVPR*, 2020. 2, 6

[52] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018. 2, 3

[53] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. 2

[54] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020. 2

[55] Dean A Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *NeurIPS*, 1989. 2

[56] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. *ITSC*, 2020. 2

[57] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *ECCV*, 2018. 2

[58] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *ICLR*, 2020. 2

[59] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 2

[60] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 2

[61] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 2

[62] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In *CVPR*, 2019. 2

[63] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments. In *CoRL*, 2018. 2

[64] Samuel Schulter, Menghua Zhai, Nathan Jacobs, and Manmohan Chandraker. Learning to look around objects for top-view representations of outdoor scenes. In *ECCV*, 2018. 2

[65] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018. 3

[66] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 5, 6

[67] Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. *ICLR*, 2017. 3

[68] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *CVPR*, 2020. 2

[69] Antonio Visioli. *Practical PID control*. Springer Science & Business Media, 2006. 5

[70] Dequan Wang, Coline Devin, Qi-Zhi Cai, Philipp Krähenbühl, and Trevor Darrell. Monocular plan view networks for autonomous driving. *IROS*, 2019. 2

[71] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving. In *ICRA*, 2019. 2

[72] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D object detection and tracking. *arXiv preprint arXiv:2006.11275*, 2020. 2, 4

[73] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019. 2

[74] Zhishuai Zhang, Jiyang Gao, Junhua Mao, Yukai Liu, Dragomir Anguelov, and Congcong Li. Stinet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *CVPR*, 2020. 2

[75] Albert Zhao, Tong He, Yitao Liang, Haibin Huang, Guy Van den Broeck, and Stefano Soatto. Lates: Latent space distillation for teacher-student driving policy learning. *arXiv preprint arXiv:1912.02973*, 2019. 2

[76] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019. 2

[77] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li, and Dahua Lin. Generative adversarial frontal view to bird view synthesis. In *3DV*, 2018. 2

[78] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017. 2