2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Urban Driving with Conditional Imitation Learning

Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov
Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, Alex Kendall

Abstract— **Hand-crafting generalised decision-making rules for real-world urban autonomous driving is hard. Alternatively, learning behaviour from easy-to-collect human driving demonstrations is appealing. Prior work has studied imitation learning (IL) for autonomous driving with a number of limitations. Examples include only performing lane-following rather than following a user-defined route, only using a single camera view or heavily cropped frames lacking state observability, only lateral (steering) control, but not longitudinal (speed) control and a lack of interaction with traffic. Importantly, the majority of such systems have been primarily evaluated in simulation - a simple domain, which lacks real-world complexities. Motivated by these challenges, we focus on learning representations of semantics, geometry and motion with computer vision for IL from human driving demonstrations. As our main contribution, we present an end-to-end conditional imitation learning approach, combining both lateral and longitudinal control on a real vehicle for following urban routes with simple traffic. We address inherent dataset bias by data balancing, training our final policy on approximately 30 hours of demonstrations gathered over six months. We evaluate our method on an autonomous vehicle by driving 35km of novel routes in European urban streets.**

I. Introduction

Driving in complex environments is hard, even for humans, with complex spatial reasoning capabilities. Urban roads are frequently highly unstructured: unclear or missing lane markings, double-parked cars, narrow spaces, unusual obstacles and other agents who follow the road rules to widely varying degrees. Driving autonomously in these environments is an even more difficult robotics challenge. The state space of the problem is large; coming up with a driving policy that can drive reasonably well and safely in a sufficiently wide variety of situations remains an open challenge. Additionally, while the action space is small, a good driving policy likely requires a combination of high-level hierarchical reasoning and low-level control.

There are two main established paradigms for solving the problem of autonomous driving: a traditional engineering-based approach and a data-driven, machine-learning approach. The former performs well in structured driving environments, such as highways or modern suburban developments, where explicitly addressing different scenarios is tractable and sufficient for human-like driving [1], [2], [3]. This approach is more mature, and the focus of commercial efforts today. However, is still unclear if this can scale to deploying fully autonomous vehicles world-wide, in complex urban scenarios with wider variation and visual diversity.

In comparison, data driven methods avoid hand-crafted rules and learns from human driving demonstrations by training a policy that maps sensor inputs, such as images, to control commands or a simple motion plan (e.g., steering and speed). Importantly, such an approach has greater potential to generalise to the large state space and address variations such as weather, time of day, lighting, surroundings, type of roads and behaviour of external agents. The attraction here is that we have seen learned approaches to many vision tasks outperform hand-crafted approaches. We expect that it will also be possible to do so for learned visual control.

Prior works primarily explore the data-driven approach in the context of imitation learning, but to date address only a subset of the state space [4], [5] or the action space [6]. In more recent work, [7] learns both steering and speed, conditioned on a route command (e.g. turn left, go straight, etc.), but evaluates the method on a small-sized toy vehicle in structured scenarios with no traffic. Similarly, [6] addresses control conditioned on a route with GPS localisation, though learns only lateral control (steering) for driving empty roads.

Our main contribution is the development of an end-to-end conditional imitation learning approach, which, to the best of our knowledge, is the first fully learned demonstration of complete control of a real vehicle, following a user-prescribed route in complex urban scenarios. Additionally, we include the first examples of a learned driving policy reacting to other traffic participants. Our method requires monocular camera images as input and can be conditioned on a route command (e.g., taking a turn at an intersection [7]), in order to follow the specified route.

Importantly, we utilise small amounts of driving data such that our final policy is trained on only 30 hours of demonstrations collected over six months, yet it generalised to routes it has not been trained to perform. We train and evaluate[1] on European urban streets: a challenging, unstructured domain in contrast to simulated environments, or structured motorways and suburban streets.

The novel contributions of this work are:

- the first end-to-end learned control policy able to drive an autonomous vehicle in dense urban environments,
- a solution to the causal confusion problem, allowing motion information to be provided to the model for both lateral and longitudinal control of the vehicle,
- demonstration of data efficiency, showing that it is possible to learn a model capable of decision making in urban traffic with only 30 hours of training data,
- comprehensive performance evaluation of end-to-end deep learning policies driving 35km on public roads.

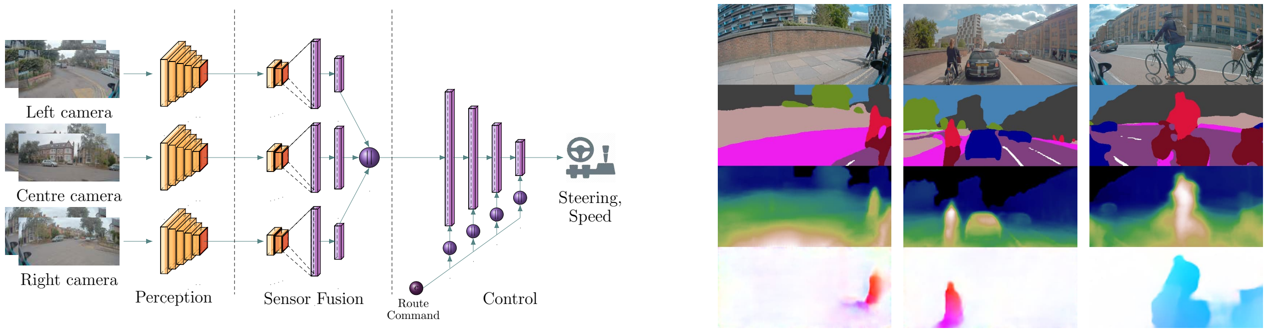
II. Related Work

The first application of imitation learning (IL) to autonomous driving was ALVINN [5], predicting steering from images and laser range data. Recently, IL for autonomous driving has received renewed interest due to advances in

The authors are with Wayve, in London, UK. Equal contribution. research@wayve.ai

[1] A video of the learned system driving a vehicle on urban streets is available at https://wayve.ai/blog/learned-urban-driving

978-1-7281-7395-5/20/$31.00 ©2020 IEEE



(a) Model structure, showing a conceptual grouping based on classical robot architectures. We frame this single fully differentiable architecture conceptually in terms of *perception*, *sensor fusion* and *control* components, though the division is purely semantic.

(b) An example of the intermediate learned perception features from the three forward-facing cameras on our vehicle in a typical urban scene (forward-left, forward, and forward-right). From top to bottom: RGB input, segmentation, monocular depth, optical flow.

Fig. 1: We demonstrate conditional route following using an end-to-end learned driving policy in dense urban driving scenarios, including simple traffic following behaviour. The model structure is outlined in Figure 1a. Each camera provides an image (or images where using optical flow) to the network, generating a learned computer vision representation for driving. We decode various perception outputs from this intermediate representation (semantics, geometry, and motion) shown in Figure 1b. The output features are compressed by sensor fusion and concatenated (|| denotes concatenation). Based on this representation and a route command **c**, the network outputs a short motion plan in steering and speed.

deep learning. [4], [8] demonstrated an end-to-end network, controlling steering from single camera for lane following on empty roads. [9] adds conditional navigation to this same approach. [7] learns longitudinal and lateral control via conditional imitation learning, following route commands on a remote control car in a static environment. [6] develops a steering-only system that learns to navigate a road network using multiple cameras and a 2D map for localisation, however it uses heavily cropped images, and focuses on localisation and route map generation.

The CARLA simulator [10] has enabled significant work on learning to drive. One example is the work in [11], which established a new behaviour cloning benchmark for driving in simulation. However, simulation cannot capture real-world complexities, and achieving high performance in simulation is significantly simpler due to a state space with less entropy, and the ability to generate near-infinite data. Several approaches have transferred policies from simulation to the real world. [12] uses semantic segmentation masks as input and waypoints as output. [13] learns a control latent space that allows domain transfer between simulation and real world. Sim-to-real approaches are promising, but policy transfer is not always effective and robust performance requires careful domain randomisation [14], [15], [16].

Rather than learning a driving policy from egocentric camera images, [17] and [18] present approaches to using imitation learning based on a bird's eye view (BEV) of a scene (using a fused scene representation and LiDAR data respectively, neither being fully end-to-end due to the need for prior perception and/or mapping). However, both approaches require additional infrastructure in the form of perception, sensor fusion, and high-definition (HD) mapping: in preference to this, we advocate for learning from raw sensor (camera) inputs, rather than intermediate representations.

Recently, model-based reinforcement learning (RL) for learning driving from simulated LiDAR data in [19], but it has yet to be evaluated in real urban environments. Approaches with low dimensional data have shown promising results in off-road track driving [20]. Model-free RL has also been studied for real-world rural lane following [21].

Our work fits in the end-to-end imitation learning literature and takes inspiration from the much of the work here. We extend the capabilities of a driving policy beyond what has been demonstrated to date: learning full lateral and longitudinal control on a vehicle with conditional route following and behaviour with traffic. We focus on novel urban routes, evaluated over 35km of driving.

III. Method

Consider a dataset of observation-action pairs $\mathcal{D}\{\langle \mathbf{o}_i, \mathbf{a}_i \rangle\}_{i=1}^{N}$, collected from expert demonstrations. The goal of IL is to learn a policy $\pi_\theta(\mathbf{o}_t) : \mathcal{O} \rightarrow \mathcal{A}$ that maps observations $\mathbf{o}_t$ to actions $\mathbf{a}_t$ at every time step and is able to imitate the expert. The parameters θ of the policy are optimized by minimizing a distance $\mathcal{L}$ between the predicted action and the expert action:

$$\min_\theta \sum_i \mathcal{L}\big(\pi_\theta(\mathbf{o}_i), \mathbf{a}_i\big) \tag{1}$$

Conditional IL (CIL) [7] seeks to condition the policy on some high-level command **c** that can convey the user intent at test time. Such a command can serve to disambiguate multi-modal behaviour: for example, when a car approaches an intersection, the camera input is not sufficient to predict whether the car should turn left or right, or go straight. Providing a route command **c** helps resolving the ambiguity. The CIL objective can be written as:

$$\min_\theta \sum_i \mathcal{L}\big(\pi_\theta(\mathbf{o}_i, \mathbf{c}_i), \mathbf{a}_i\big) \tag{2}$$

A. Model Architecture

Our method learns directly from images and outputs a local motion plan for speed and steering. The driving policy is a fully end-to-end neural network, but we form three conceptual components: *perception*, *sensor fusion* and *control*. We use a combination of pretrained and trainable layers to learn a robust driving policy. Figure 1 outlines the architectural details discussed below.

1) Perception: The perception component of our system consists of a deep encoder-decoder similar to multitask segmentation and monocular depth [22], trained to reconstruct RGB, depth and segmentation. For driving, we use the encoded features, which contain compressed information about the appearance, the semantics, and distance estimation. In principle, such representations could be learned simultaneously with the driving policy, for example, through distillation or by adding perception labels. However, to improve data efficiency and robustness when learning control, we pretrain the perception network on several large, heterogeneous, research vision datasets [23], [24], [25], [26], [27].

The perception architecture above does not have any temporal information about the scene. One solution is to use concatenated representations of the past n frames, but this did not perform well in our experiments. Instead, we use intermediate features from an optical flow model similar to [28], and concatenate this to form features rich in semantics, geometry, and motion information.

In some experiments we additionally use a left-facing and a right-facing view. In this case, the perception component treats each of these frames independently. Figure 1b provides an example of the perception output on input images.

2) Sensor Fusion: The purpose of the sensor fusion component is to aggregate information from the different sensors into a single representation of the scene for driving.

Singleview. In the case of a single camera, the model is composed of a simple feedforward architecture of convolutional and fully-connected layers.

Multiview. Based on single-frame observations from 3 cameras: front, left and right (Figure 1). Driving with a single view means the model must predict an action without full state observability (e.g., losing sight of the adjacent kerb at a junction). Enhancing observability with multiple views should improve behaviour in these situations. However, we found that naively incorporating all camera features led to over-reliance on spurious information in the side views. Thus, we occasionally corrupt the side view encoder outputs during training, using the augmentation procedure described below for flow. We also apply self-attention [29] on the perception output of each camera, allowing the network to model dependencies across the image and focus on the parts of the image that are important for driving.

Multiview and Flow. Additionally to the Multiview architecture above, optical flow is computed only for the front camera using the last two consecutive frames. The flow features are concatenated to the perception features for the front camera.

Access to more information, such as motion, can lead to worse performance. As discussed by [11], [30], imitation learning in particular can suffer from *causal confusion*: unless an explicit causal model is maintained, spurious correlations cannot be distinguished from true causes in the demonstrations. For example, inputting the current speed to the policy causes it to learn a trivial identity mapping, making the car unable to start from a static position. While [30] proposes a way to learn a causal graph, it assumes access to a disentangled representation and its complexity grows exponentially with the number of features.

Instead, we propose a method, which cannot determine the causal structure, but can overcome the causal confusion problem and scale to large number of features. Given an input $\mathbf{x}$, we add random noise, e.g. Gaussian, and apply dropout [31] on the noise with probability 0.5 for the full duration of training. This allows the model to use the true value of $\mathbf{x}$ when available and breaks the correlation the rest of the time, forcing the model to focus on other features to determine the correct output. Applying this approach on the flow features during training allowed the model to use explicit motion information without learning the trivial identity solution.

3) Control: The control module consists of several fully-connected layers that process the representation computed by sensor fusion. At this stage, we also input a driving route command $\mathbf{c}_t$, corresponding to one of *go-straight*, *turn-left* or *turn-right*, as a one-hot-encoded vector. We found that inputting the command multiple times at different stages of the network improves robustness of the model.

We selected this encoding to ground this work with that demonstrated in CARLA [7], [11], [12], [19]. In practice, this limits real world testing to grid-like intersections. We use this encoding for the approach outlined here, but we favour a richer learned route embedding similar to [6], [32].

The output of the control module consists of a simple parameterised motion plan for lateral and longitudinal control. In particular, we assume that the vehicle motion is locally linear and we output a line, parameterised by the current prediction $\mathbf{y}_t$ and a slope $\mathbf{m}_t$, for both speed and steering. During training, we minimise the mean squared error between the expert actions, taken over N timesteps into the future, and the corresponding motion predicted by the network:

$$\sum_{n=0}^{N-1} \gamma^{\Delta_n} \left(\mathbf{y}_t + \mathbf{m}_t \Delta_n - \mathbf{a}_{t+n}\right)^2 \tag{3}$$

where γ is a future discount factor, Δ_n is the time difference between steps t and $t+n$, $\mathbf{y}_t$ and $\mathbf{m}_t$ are the outputs of the network for $\mathbf{o}_t$, and $\mathbf{a}_{t+n}$ is the expert speed and steering control at timestep $t+n$. Predicting such a trend provides for smoother vehicle motion and demonstrated better performance both in closed loop and open loop testing.

B. Data

Driving data is inherently heavily imbalanced, where most of the captured data will be driving near-straight in the middle of a lane (Figure 3). To mitigate this, during training we sample data uniformly across lateral and longitudinal control dimensions. We found that this avoids the need for data augmentation [17] or synthesis [4].

We split the data into $k \geq 1$ bins by steering, defining x_0, x_k as the minimal and maximal values, and the bin edges $x_1, \ldots, x_{k-1}$. We define edges such that the product of the number of data in each bin and the width of that bin (i.e. $x_j - x_{j-1}$) are equal across all bins, finding these edges using gradient descent. We assign sampling weights to data:

$$w_j = \frac{x_j - x_{j-1}}{N_j} \cdot \frac{W}{x_k - x_0}, \tag{4}$$

where w_j denotes the weight of data in the j-th bin, N_j — the number of data in that bin and W — the total weight of the dataset (equal to the number of data). The total weight of each bin is proportional to its width as opposed to the number of data. The second factor is for normalisation: it ensures that the total dataset weight is W. We recursively apply this to balance each bin w.r.t. speed.

IV. Experiments

The purpose of our experiments is to evaluate the impact of the following features on the driving performance of the policy:

- Using computer vision to learn explicit visual representations of the scene, in contrast to learning end-to-end strictly from the control loss,
- Improving the learned latent representation by providing wider state observability through multiple camera views, and adding temporal information via optical flow,
- Understanding influence of training data diversity on driving performance.

To assess these questions, we train the following models:

- SV: Single view model. Based on the network described in Section III-A, using only the forward facing camera. The parameters of the perception module are pretrained and frozen while training the sensor fusion and the control modules. A local linear motion plan for speed and steering is predicted over 1s in the future. This model serves as our baseline.
- MV: Multiview model. As SV, but camera observations from forward, left, and right views are used, described in III-A.2.
- MVF: As MV, but with optical flow.
- SV[75, 50, 25]: As SV, but using only the first N% of the training data. The latter portion of the data was discarded, degrading the diversity while retaining comparable temporal correlation.
- SVE2E: As SV, but trained fully end-to-end with randomly initialised parameters.
- SVE2EFT: E2E fine-tuned perception. SV model with a second E2E fine-tuning stage.
- SVE2EPT: As SVE2E, but perception parameters pretrained as in SV.

We evaluate all of the above models in closed loop on real-world urban European streets using the procedure described in Section IV-A and the metrics discussed in IV-C. We note that we also evaluated the baseline model SV in simulation with an in-house simulator: its performance was significantly better compared to the real world, highlighting the simplification of the problem by simulation.

A. Procedure

Data Collection. We collected data from human demonstrations over the span of 6 months for a total of 30 driving hours in a densely populated, urban environment representative of most European cities. The drivers involved were not given any explicit instructions and were free to choose random routes. Collected data includes front, left and right cameras as well as measurements of the speed and steering controls from the driver. Images were captured from rolling shutter cameras, synchronised within 5ms with a frame rate of 15Hz. Scalar values (e.g., speed) were received at 100Hz. We used the front camera as the primary clock, associating this with the latest received values for the rest of the sensors. Route annotations were added during postprocessing of the data, based on the route the driver took. Table I and Figure 3 summarise the training and test datasets.

Dataset	Frames	Distance (km)	Time (hours)	Drivers
Training	735K	613	27.2	8
Test	91K	89	3.4	1

TABLE I: Training and test driving datasets, collected over a 6-month period in a typical European city.

Evaluation Procedure We evaluate the ability of a learned driving system to perform tasks that are fundamental to urban driving: following lanes, taking turns at intersections and interacting with other road agents. We select two routes not present in the training data, each approximately 1km in length, and measure the intervention rates while autonomously completing the manoeuvres required in order to drive the route. Each route was selected to have dense intersections, and be a largely Manhattan-like environment analogous to [7] to avoid ambiguity with the route encoding. Figure 2 shows the locations of these routes. Completing a route in a busy urban environment implicitly entails navigating around other dynamic agents such as vehicles, cyclists and pedestrians, to which the model has to respond safely through actions such as stopping or giving way.

Each model was commanded to drive the route in both directions, each of which counts as an attempt. Should a given model have five interventions within a single route attempt, the attempt was terminated early. Models with consistently poor performance were evaluated on fewer attempts.

We additionally evaluate these models interacting with another vehicle, following a pace car which periodically stopped throughout the driving route. During these tests, we measure intervention rates for stopping behind a stationary vehicle or failing to follow the pace car.

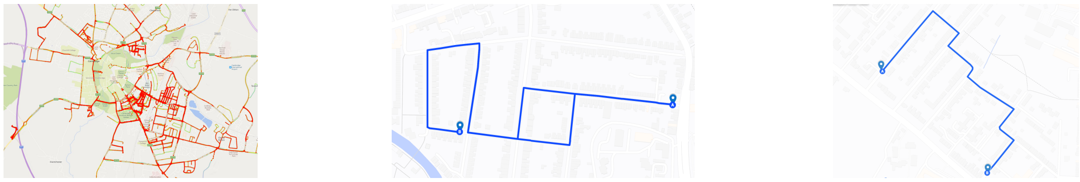
The interventions as well as their types are left to the judgment of the safety-driver, who is able to take over control whenever the model exhibits poor behaviour. Evaluating a driving policy's performance in terms of the number and type of interventions is preferable to computing low level metrics (e.g., cross-track error to a path) as there are multiple solutions to performing the task correctly.

We note that neither of the testing routes has been seen by the model during training, though the data has some overlap with some sections. While stopping for and following vehicles is a common occurrence in the data, the particular test vehicle used was not seen during training.

Finally, the testing environment will inevitably contain time-dependent factors beyond our control - this is expected for outdoor robot trials. Examples include the position and appearance of parked and dynamic vehicles, weather conditions, presence and behaviour of other agents. We have controlled for these factors as much as possible by running trials with batches of models in succession. Fundamentally, this real-world constraint forces a driving system to generalise: each attempt of the same route is novel in a different way, e.g., weather, lighting, road users, static objects.

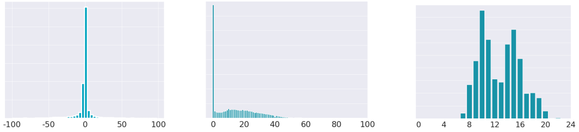
B. Training Procedure

The network described in Sections III-A is trained jointly using stochastic gradient descent using batches of size 256



(a) Training collection area (b) Test route A (c) Test route B

Fig. 2: We collect exemplar driving data in the European urban area in 2a. We evaluate the learned driving policy performance in two urban routes, each approximately 1km in length with multiple intersections. Both routes are dense urban scenes which broadly fall into a Manhattan-like grid structure, with a mix of road widths. Models tested on each route are evaluated in both directions: note that route A takes a slightly different route on the return due to visibility at an intersection.



(a) Steering (± %) (b) Speed (km/h) (c) Time (hour)

Fig. 3: We collect training data driving across a European city (see the heatmap in 2a, showing data density). The data, as is typical, is imbalanced, with the majority driving straight (3a), and a significant portion stationary (3b).

for $200k$ iterations. The initial learning rate is set to 0.01 and decayed linearly during training, while momentum and weight decay are set to 0.9 and 0.0001 respectively. In total, the network architecture used here consisted of $13-26$M trainable parameters, varying depending on the perception encoder used, and the number of cameras.

C. Metrics

We need a principled way of identifying which of the models may perform best under closed-loop test conditions. For this purpose we use open-loop metrics of the control policy, computed on a validation dataset. As discussed by [33], [34], the correlation between offline open-loop metrics and online closed-loop performance is weak. We observed similar results, but we found some utility in using weighted mean absolute error for speed and steering as a proxy for real world performance (referred to in Table II as Balanced MAE). We used this metric on the test dataset in Table I to select models for deployment. For the experimental results, we adopt the following closed-loop metrics:

- Intervention rate, computed as metres per intervention,
- Intervention rate only during lane following,
- Success rate of turning manoeuvres (left, right),
- Intervention rate while following a pace car, travelling at up to 15 km/h,
- Success rate of stopping behind a pace car.

We also assess the following intervention categories:

- *Obstacle*: e.g., prevention of crashing into an obstacle such as a parked car or road works,
- *Road position*: e.g., wrong side of traffic or dangerous proximity to the kerb,
- *Disobeyed navigation*: e.g., drove straight instead of turning left,
- *Failed manoeuvre*: e.g., attempted the correct turn, but failed to complete the manoeuvre,
- *Dynamic vehicle*: e.g., acted poorly with other traffic, such as not stopping behind a car.

Finally, while these metrics provide quantitative results, they fail to capture qualitative properties. We therefore provide some commentary on the models' observed behaviour.

D. Results

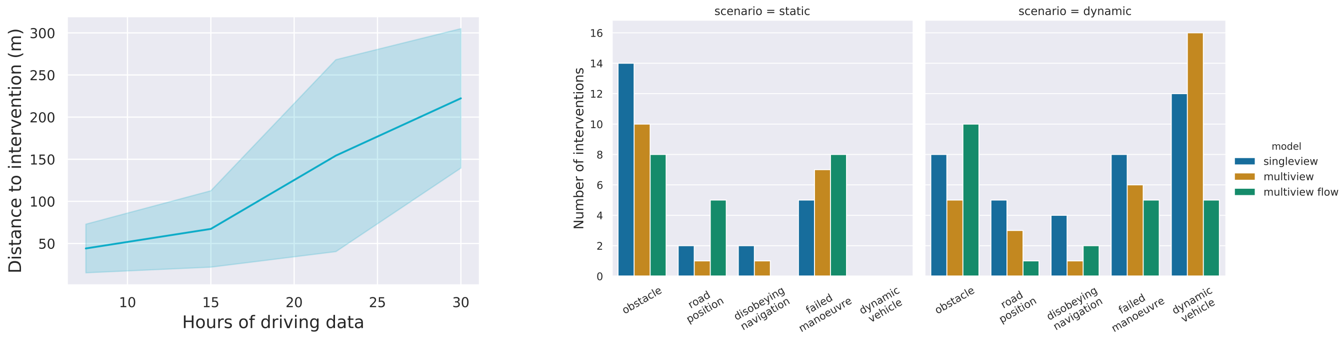
Table II outlines the experimental results. A total of 34.4km of driving was performed. We consider the performance in a static environment (*driving manoeuvres*) independently of that while following a pace car (*traffic following*).

1) Representation Learning with Computer Vision: Comparing the performance of SV to the E2E trained models (SVE2E, SVE2EFT, and SVE2EPT) suggests that learned scene understanding is a critical component for a robust representation for driving. Both fully E2E trained models (SVE2E, SVE2EPT) perform very poorly. We attribute this to the fact that our perception model is more robust to the diversity of real world images [35], in part influenced by the additional data seen during the perception training phase. In robotics we must consider methods which prioritise data efficiency: we will never have the luxury of infinite data. Here we train on 30 hours of driving: not nearly sufficient to cover the diversity in real world scenes. We observed that the E2E trained models performed comparably to SV in shaded environments, but poorly in bright sunlight. We attribute this to appearance change between our training data and the test environment as the data was gathered in winter, three months prior to testing in spring. Additionally, the fine-tuned model performs similarly to the baseline; on this basis we recommend using pretrained perception due to the wider benefits of interpretability.

2) Improving the learned state: observability & temporal information: Firstly we consider the performance of multi-view models compared to the singleview baseline. We see an improvement in driving performance when providing the model with a more complete view of the scene. This is apparent when considering the *disobeyed navigation* interventions: the SV model systematically fails to take certain turns. For example, if the side road is narrow, it tends to drive straight past ignoring the route. Equally, the model commonly cuts the near-side kerb while turning. Both these failure modes are attributable to the fact that the forward facing camera has no visibility of this part of the scene. Multiview models tend not to exhibit these failure modes. Qualitatively, we found the MV models provided a much larger buffer around the kerb, however both would exhibit the same failure mode of poor positioning behind parked cars around turns without space to correct. We attribute this to the low perception resolution and simple action representation used. Secondly, we consider

Model	Attempts		Distance (m)		Intervention rate (m/int)		Success Rate (%)			Open Loop	
	A	B	Manoeuvres (LF only)	Traffic follow	Manoeuvres (LF only)	Traffic Follow	Turn: Left	Turn: Right	Stop: Car	MAE	Bal-MAE
SV	14	12	5113 (4094)	3158	222 (372)	121	74% (N=31)	87% (N=31)	69% (N=36)	0.0715	0.0630
MV	14	12	5811 (4613)	3440	306 (659)	191	70% (N=23)	86% (N=36)	65% (N=40)	0.0744	0.0706
MVF	14	12	5313 (4086)	2569	253 (511)	161	78% (N=32)	82% (N=34)	81% (N=36)	0.0633	0.0612
SV75	2	2	2777 (2258)	-	154 (376)	-	57% (N=13)	80% (N=17)	-	0.0985	0.0993
SV50	2	4	2624 (2145)	-	67 (98)	-	54% (N=19)	50% (N=16)	-	0.0995	0.1015
SV25	2	-	926 (746)	-	44 (50)	-	33% (N=6)	71% (N=7)	-	0.1081	0.1129
SVE2E	2	2	823 (615)	-	30 (30)	-	42% (N=12)	100% (N=5)	-	0.1410	0.1365
SVE2EFT	2	2	1591 (759)	-	177 (379)	-	69% (N=16)	90% (N=21)	-	0.0769	0.0801
SVE2EPT	2	2	260 (177)	-	26 (59)	-	33% (N=3)	0% (N=5)	-	0.0966	0.0946

TABLE II: Driving performance metrics evaluated on routes A and B (see IV and IV-A). *Traffic Follow* and *Manoeuvres* present the metrics for following the routes with and without a pace car. The *LF only* metrics refer to lane following. In *Success Rate* metrics, N refers to the number of manoeuvre attempts.



(a) Performance trend with increasing data

(b) Interventions by type

Fig. 4: From the results in Table II we make a number of observations. In 4a we see a clear relationship between data and model performance (showing mean, s.d.). Inspecting the interventions in 4b we observe that the addition of flow improves dynamic vehicle behaviour, but potentially suffers due to a lower signal-to-noise ratio with increased dimensionality.

the benefit of augmenting the learned representation with temporal state, describing the scene motion with optical flow. MVF performs slightly worse than MV in these metrics. We attribute this to a lower signal-to-noise ratio given the increased dimensionality, though we consider these to be largely comparable. Where this motion information clearly has benefit is behaviour with other vehicles (stopping): the MVF model responds more robustly, demonstrated by a success rate increase from 65% to 81%.

3) Influence of data quantity & diversity on performance: The learned driving policies presented here need significant further work to be comparable to human driving. We consider the influence of data quantity on SV model performance, observing a direct correlation in Figure 4a. Removing a quarter of the data notably degrades performance, and models trained with less data are almost undriveable. We attribute this to a reduction in data diversity. As with the E2E trained models, we found that the reduced data models were able to perform manoeuvres in certain circumstances. Qualitatively, all models tested perform notably better during conditions closer to the collection period (concluded three months prior to these experiments). We observed that, in the best case, these models drove 2km without intervention. While we do not know the upper limit on performance for the effect of data, it is clear that more data than our training set — and more diversity — is likely to improve performance.

V. Conclusions

We present a conditional imitation learning approach which combines both lateral and longitudinal control on a real vehicle. To the best of our knowledge, we are the first to learn a policy that can drive on a full-size vehicle with full vehicle actuation control in complex real-world urban scenarios with simple traffic. Our findings indicate that learning intermediate representations using computer vision yields models which significantly outperform fully end-to-end trained models. Furthermore, we highlight the importance of data diversity and addressing dataset bias. Reduced state observability models (i.e., single view) can perform some manoeuvres, but have systematic failure modes. Increased observability through the addition of multiple camera views helps, but requires dealing with causal confusion.

The method presented here has a number of limitations which we believe are essential for achieving (and exceeding) human-level driving. For example, it does not have access to long-term dependencies and cannot 'reason' about the road scene. This method also lacks a predictive long-term planning model, important for safe interaction with occluded dynamic agents. There are many promising directions for future work. The time and safety measures required to run a policy in closed loop remain a major constraint. Therefore, being able to robustly evaluate a policy offline and quantify its performance is important area for research. Including the ability to learn from corrective interventions (e.g., [36], [21]) is another vital direction for future work, and for generalising to real-world complexities across different cities.

Acknowledgments

We thank our colleagues for their support for this work: Andrew Jones, Nikhil Mohan, Alex Bewley, Edward Liu, John-Mark Allen, Julia Gomes, and Yani Ioannou.

References

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the darpa grand challenge: Research articles," *J. Robot. Syst.*, vol. 23, no. 9, pp. 661–692, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1002/rob.v23:9

[2] E. D. Dickmanns, "The development of machine vision for road vehicles in the last decade," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 268–281.

[3] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[4] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[5] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in neural information processing systems*, 1989, pp. 305–313.

[6] A. Amini, G. Rosman, S. Karaman, and D. Rus, "Variational end-to-end navigation and localization," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[7] F. Codevilla, M. Mueller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[8] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017.

[9] C. Hubschneider, A. Bauer, M. Weber, and J. M. Zöllner, "Adding navigation to the equation: Turning decisions for end-to-end vehicle control," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–8.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on Robot Learning*, 2017, pp. 1–16.

[11] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," *CoRR*, vol. abs/1904.08980, 2019.

[12] M. Mueller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving Policy Transfer via Modularity and Abstraction," in *Proceedings of The 2nd Conference on Robot Learning*, vol. 87, 2018, pp. 1–15.

[13] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to drive from simulation without real world labels," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[14] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.

[15] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder *et al.*, "Domain randomization and generative models for robotic grasping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3482–3489.

[16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[17] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[18] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," *CoRR*, vol. abs/1810.06544, 2018.

[20] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1714–1721.

[21] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[22] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[24] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: https://www.mapillary.com/dataset/vistas

[25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[26] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, April 2018.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[29] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[30] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *CoRR*, vol. abs/1905.11979, 2019. [Online]. Available: http://arxiv.org/abs/1905.11979

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[32] S. Hecker, D. Dai, and L. Van Gool, "End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners," in *European Conference on Computer Vision (ECCV)*, 2018.

[33] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[34] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[35] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[36] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 627–635.