Urban Driving with Conditional Imitation Learning

Jeffrey Hawke* **Richard Shen*** **Corina Gurau*** **Siddharth Sharma***

Daniele Reda* **Nikolay Nikolov*** **Przemyslaw Mazur*** **Sean Micklethwaite***

Nicholas 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.

1 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 [1–3]. 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

*Equal Contribution. Contact information: research@wayve.ai, further information including videos at https://wayve.ai/blog/learned-urban-driving

Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

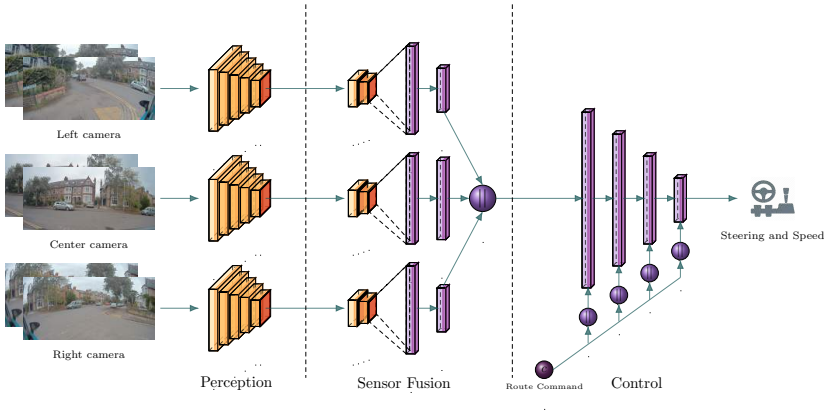


Fig. 1: We demonstrate conditional route following using an end-to-end learned driving policy in dense urban driving scenarios, including simple traffic following behaviour. We frame this single fully differentiable architecture conceptually in terms of *perception*, *sensor fusion* and *control* components. The division is purely semantic. Each camera provides an image (or images where using optical flow) to the perception encoders, generating a learned computer vision representation for driving. We generate various perception outputs from this intermediate representation, of semantics, geometry, and motion, shown in Figure 2. The output features are compressed by sensor fusion and concatenated ($||$ denotes concatenation). Based on this representation and a route command $\mathbf{c}$, the network outputs a short motion plan in steering and speed.

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] learn both steering and speed, conditioned on a route command (e.g. turn left, go straight, etc.), but evaluate the method on a small-sized toy vehicle in structured scenarios with no traffic. Similarly, [6] address control conditioned on a route with GPS localisation, though learn 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 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.

2 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 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] learn longitudinal and lateral control via conditional imitation learning, following route commands on a remote control car in a static environment. [6] develop 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 is dependent on localisation and route map generation.

The CARLA simulator [10] has enabled significant work on learning to drive. One example is the work of [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] use semantic segmentation masks as input and waypoints as output. [13] learn 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 [14–16].

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 by [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 develop new methods to deploy our system to real world environments. Specifically, we focus on urban streets, with routes not demonstrated during training, evaluated over 35km of urban driving.

3 Method

We begin with a brief introduction of imitation learning (IL) and conditional IL, followed by a discussion of our method and model architecture.

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 additionally condition the policy on some high-level command $\mathbf{c}$ 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 $\mathbf{c}$ helps resolving the ambiguity. The CIL objective can be written as:

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

3.1 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 conceptually structure it with three separate 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.

3.1.1 Perception

The perception component of our system consists of a deep encoder-decoder similar to multitask segmentation and monocular depth [22]. It receives an image as observation and is trained to

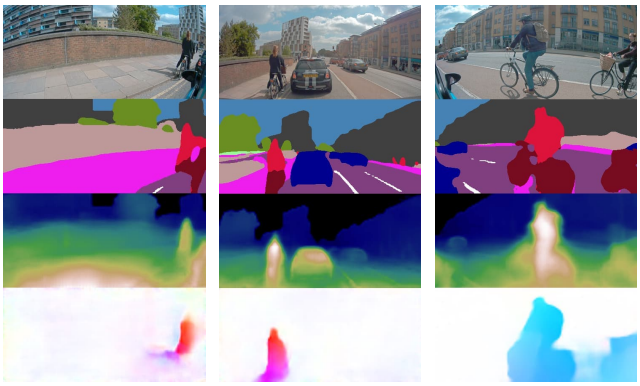


Fig. 2: An example of perception from the intermediate learned features of the three forward-facing cameras on our vehicle in a typical urban scene (forward-left, forward, and forward-right). From top to bottom: RGB input, segmentation, monocular depth, and optical flow.

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. Alternatively, the labelled control data could be labelled for perceptual tasks. However, to improve data efficiency and robustness when learning control, we pretrain the perception network on several large, heterogeneous, research vision datasets [23–27].

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.

Apart from the single forward-facing camera, in some of our 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 2 provides an example of the perception output on input images.

3.1.2 Sensor Fusion

The purpose of the sensor fusion component is to aggregate information from the different sensors and process them into a single representation of the scene, useful 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 observability of the necessary state (e.g., losing sight of the adjacent kerb while approaching 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 long-range dependencies across the entire 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 and then processed by sensor fusion.

Access to more information, such as motion, can actually 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] propose 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 solution of an identity mapping for speed and steering.

3.1.3 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 of a route to ground this approach with those demonstrated in CARLA [7, 11, 12, 32]. 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, 33].

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 actions predicted by the network motion plan:

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

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 trend into the future provides for smoother vehicle motion and demonstrated better performance both in closed loop and open loop testing.

3.2 Data

Driving data is inherently heavily imbalanced, where most of the captured data will be driving near-straight in the middle of a lane, as shown in Figure 5. 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], though these approaches can be helpful.

We split the data into $k \geq 1$ bins by steering value, defining x_0, x_k as the minimal and maximal steering values and the bin edges $x_1, \ldots, x_{k-1}$. We define the 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. Having defined the edges, we then assign 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.

4 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 3.1, 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 integrated by the sensor fusion module as described in 3.1.2.
- MVF: Multiview and Flow model. As MV, but with additional optical flow information.
- SV75: As SV, but using only the first 75% of the training data. The latter 25% of the data was discarded, degrading the diversity while retaining comparable temporal correlation.
- SV50: As SV75, but using only 50% of the training data (discarding the most recent 50%).
- SV25: As SV75, but using only 25% of the training data (discarding the most recent 75%).
- SVE2E: Same 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: Same 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 4.1 and the metrics discussed in 4.2. 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.

4.1 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 within the city. 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. For further information see Appendix C.

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 in the same city, each approximately 1km in length and measure the intervention rates while autonomously completing the manoeuvres required in order to reach the end of the route from the start. 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 4 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.

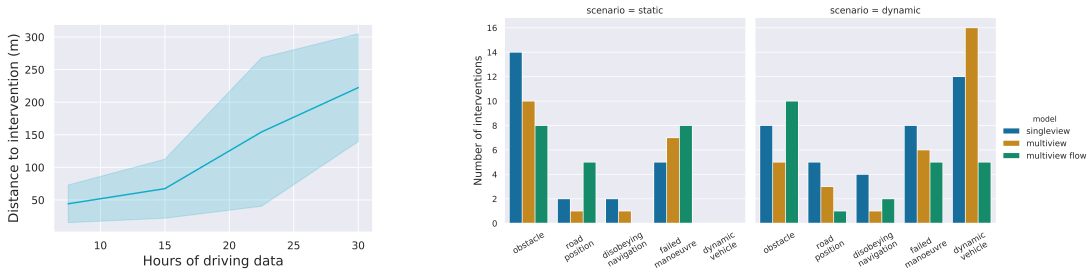
The interventions as well as their types are left to the judgment of the safety-driver, who is able to take over control whenever the model exhibits poor behaviour. Evaluating a driving policy's performance in terms of the number and type of interventions is preferable to computing low level metrics (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 mobile robot trials. Examples include the position and appearance of parked and dynamic vehicles, weather conditions, presence and behaviour of other agents, however, 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.

4.2 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 hold-out validation dataset. As discussed by [34] and [35], the correlation between offline open-loop metrics and online closed-loop performance is weak. We observed similar results, but we found some



(a) Performance w. increasing data

(b) Interventions by type

Fig. 3: From the results in Table 1 we make a number of observations. In 3a we see a clear relationship between data quantity and SV model performance (showing mean, s.d.). Inspecting the interventions types in 3b we observe that the addition of optical flow improves dynamic vehicle behaviour, but potentially suffers due to a lower signal-to-noise ratio with increased dimensionality.

utility in using open-loop weighted mean absolute error for speed and steering as a proxy for real world performance (referred to in Table 1 as Balanced MAE). In particular, we used this metric on the test dataset in Table 3 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 (a subset of the previous metric),
- 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.

In addition to the rate of interventions, we also provide their type using the following 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.

4.3 Results

Table 1 outlines the experimental results. A total of 34.4km of driving was performed under automation, prioritising the SV, MV, and MVF models. We consider the performance in a static environment (*driving manoeuvres*) independently of the performance following a pace car (*traffic following*).

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

Tab. 1: Driving performance metrics evaluated on routes A and B (see 4 and 4.1). *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*, N refers to the number of manoeuvre attempts.

4.3.1 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 significantly more robust to the diversity of appearance present in real world driving images [36], in part influenced by the additional data seen

during the initial perception training phase. In robotics we must consider methods which prioritise data efficiency: we will never have the luxury of infinite data. For example, here we train on 30 hours of driving, not nearly sufficient to cover the diversity of appearance in real world scenes.

We observed that the E2E trained models performed comparably to SV when operating in shaded environments, but poorly in bright sunlight. We attribute this to the appearance change between our training data and the test environment: the data was gathered in winter, three months prior to tests being conducted 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.

4.3.2 Improving the learned state: observability and temporal information

Firstly we consider the performance of multiview models compared to the 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 the forward facing camera having no visibility of this part of the scene. MV models tend not to exhibit these failure modes. Qualitatively, we found the MV models to provide a much larger buffer around the kerb, however both models would exhibit the same failure mode of poor positioning behind parked cars around turns without space to correct. We attribute this failure mode to the low perception resolution and simple action representation used.

Secondly, we consider the benefit of augmenting the learned representation with temporal state, describing the motion present in the scene 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 model with motion information responds to other agents more robustly, demonstrated by a success rate increase from 65% to 81%.

4.3.3 Influence of data quantity and diversity on performance

The learned driving policies 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 3a. 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 diversity. As with E2E trained models, we found that reduced data models were able to perform manoeuvres in certain circumstances. Qualitatively, all models perform notably better during environmental 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, it is clear that more diversity than our training set is likely to improve performance.

5 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 multiple camera views helps, but requires dealing with causal confusion.

The method presented here has a number of fundamental 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 remains a major constraint: being able to robustly evaluate a policy offline and quantify its performance is important area for research. Learning from corrective interventions (e.g., [37, 21]) is another direction necessary for generalising to real-world complexities.

References

[1] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, September 2006. ISSN 0741-2223. doi: 10.1002/rob.v23:9. URL `http://dx.doi.org/10.1002/rob.v23:9`.

[2] Ernst D Dickmanns. The development of machine vision for road vehicles in the last decade. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, pages 268–281. IEEE, 2002.

[3] John Leonard, Jonathan How, Seth Teller, Mitch Berger, Stefan Campbell, Gaston Fiore, Luke Fletcher, Emilio Frazzoli, Albert Huang, Sertac Karaman, et al. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *CoRR*, abs/1604.07316, 2016. URL `http://arxiv.org/abs/1604.07316`.

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

[6] Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. Variational end-to-end navigation and localization. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.

[7] Felipe Codevilla, Matthias Mueller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[8] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 2017.

[9] Christian Hubschneider, André Bauer, Michael Weber, and J Marius 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)*, pages 1–8. IEEE, 2017.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16, 2017.

[11] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *CoRR*, abs/1904.08980, 2019.

[12] Matthias Mueller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving Policy Transfer via Modularity and Abstraction. In *Proceedings of The 2nd Conference on Robot Learning*, volume 87, pages 1–15, 2018.

[13] Alex Bewley, Jessica Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, and Alex Kendall. Learning to drive from simulation without real world labels. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[14] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

[15] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489. IEEE, 2018.

[16] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[17] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019. doi: 10.15607/RSS.2019.XV.031.

[18] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *CoRR*, abs/1810.06544, 2018.

[20] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

[21] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[22] Alex Kendall, Yarin Gal, and Roberto 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] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[24] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017. URL `https://www.mapillary.com/dataset/vistas`.

[25] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[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*, 40(4):834–848, April 2018. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2699184.

[27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[28] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan 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*, pages 8934–8943, 2018.

[29] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[30] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *CoRR*, abs/1905.11979, 2019. URL `http://arxiv.org/abs/1905.11979`.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014.

[32] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*, 2018.

[33] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners. In *European Conference on Computer Vision (ECCV)*, 2018.

[34] Felipe Codevilla, Antonio M. Lopez, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[35] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[36] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. doi: 10.1177/0278364916679498. URL `http://dx.doi.org/10.1177/0278364916679498`.

[37] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

A Network Architecture

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.

Model	Perception	Sensor Fusion	Control	Total
SV	10.4M	2.2M	705K	13.3M
MV	10.4M	6.5M	442K	17.4M
MVF	19.1M	6.5M	442K	26.1M

Tab. 2: Number of trainable parameters used in the network architecture depicted here.

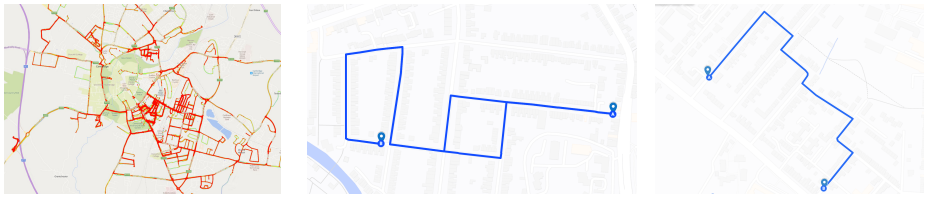
B Training Procedure

Modules described in Sections 3.1.2 and 3.1.3 are trained jointly using stochastic gradient descent (SGD) using batches of size 256 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.

C Data

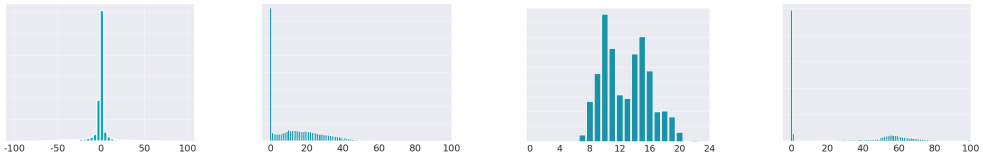
Dataset	Frames	Distance (km)	Time (hours)	Drivers
Training	735K	613	27.2	8
Test	91K	89	3.4	1

Tab. 3: Training and test driving datasets, collected over a 6-month period in a typical European city.



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

Fig. 4: We collect exemplar driving data by driving through the European urban area in 4a. 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 safety-driver visibility at an intersection. The driving environment is comparable to the scene shown in Figure 2.



(a) Steering (± %) (b) Speed (km/h) (c) Time of day (hour) (d) Throttle (%)

Fig. 5: We collect training data driving across a European city (see 4a). The data, as is typical, is imbalanced, with the majority driving straight (5a), and a significant portion stationary (5b).