Distribution-aware Goal Prediction and Conformant Model-based Planning for Safe Autonomous Driving

Jonathan Francis [*] [1] [2] Bingqing chen [*] [1] Weiran Yao [*] [1] Eric Nyberg [1] Jean Oh [1]

Abstract

The feasibility of collecting a large amount of expert demonstrations has inspired growing research interests in learning-to-drive settings, where models learn by imitating the driving behaviour from experts. However, exclusively relying on imitation can limit agents' generalisability to novel scenarios that are outside the support of the training data. In this paper, we address this challenge by factorising the driving task, based on the intuition that modular architectures are more generalisable and more robust to changes in the environment compared to monolithic, end-to-end frameworks. Specifically, we draw inspiration from the trajectory forecasting community and reformulate the learning-to-drive task as obstacle-aware perception and grounding, distribution-aware goal prediction, and model-based planning. Firstly, we train the obstacle-aware perception module to extract salient representation of the visual context. Then, we learn a multi-modal goal distribution by performing conditional density-estimation using normalising flow. Finally, we ground candidate trajectory predictions road geometry, and plan the actions based on on vehicle dynamics. Under the CARLA simulator, we report state-of-the-art results on the CARNOVEL benchmark.

1. Introduction

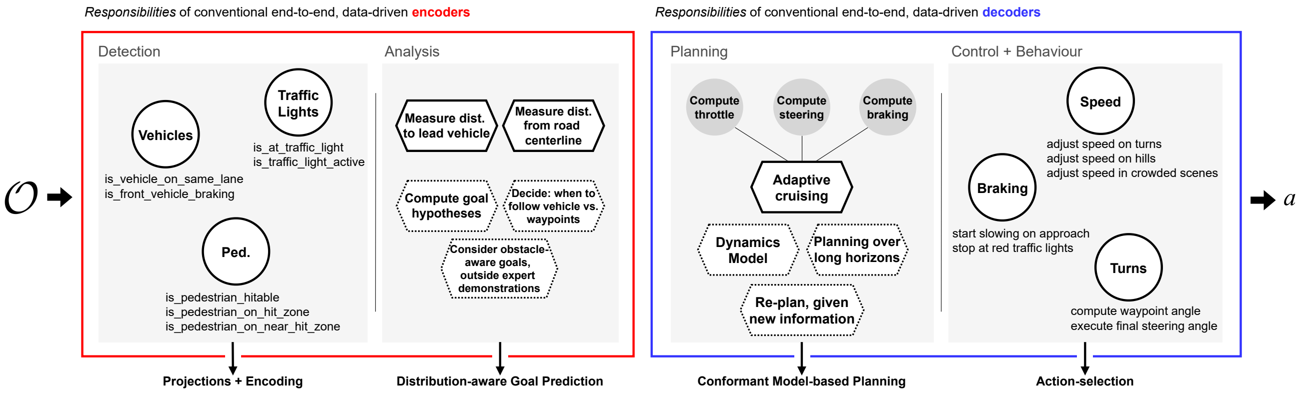
Achieving generalisability to novel scenarios in urban autonomous driving remains a challenging task for artificial intelligence (AI). Recent approaches have shown promising results in end-to-end imitation learning from expert demonstrations, wherein agents learn policies that replicate the experts' actions, at each time-step, given the corresponding observations (Bojarski et al., 2016; Codevilla et al., 2018; 2019; Muller et al., 2006; Pan et al., 2017; Pomerleau, 1989; Ohn-Bar et al., 2020; Chen et al., 2020b). Despite this progress, end-to-end imitative models often cannot capture the causal structures that underlie expert-environment interactions, leading models to misidentify the correct mappings from the observations (de Haan et al., 2019). Furthermore, if the coverage of expert demonstrations does not extend to *all* scenarios that the agent will encounter during test time, the agent will generate spurious actions in response to these out-of-distribution (OOD) observations (Filos et al., 2020).

In an effort to tackle *part* of this issue, recent works deviate from the end-to-end learning paradigm in their respective problem domains, opting instead to decompose learning into sub-modules, for trajectory forecasting (Rhinehart et al., 2018b; Filos et al., 2020), indoor robot navigation (Chen et al., 2020a), and learnable robot exploration (Chaplot et al., 2020; Chen et al., 2019). Here, the intuition is that, by breaking down the inference problem into smaller units, more control over the inference step is obtained and the causal misidentification issue is somewhat avoided, by factorising the problems via engineering judgement. The modularity of those approaches resonates with the proposed method, however those approaches use only the classical global-local hierarchical planning paradigm, where the responsibility of performing feature-extraction while also attempting to (implicitly) model the environment dynamics is still contained within a single unit, leading to spurious predictions in unseen environments. We proceed a step further, by defining modules in the learning-to-drive setting, such that each module's task is directly attributable to the components of behaviour expected of an expert agent (see Figure 1). In our decomposition, modules are given specialised roles (e.g., obstacle-awareness, goal/intention-prediction, and explicitly modelling environmental dynamics), improving the tractability of their respective sub-tasks and their complementarity towards the downstream objective.

However, the challenge of generalisability still remains. How can we effectively utilise the expert's prior experi-

[*] Equal contribution [1] School of Computer Science, Carnegie Mellon University; Pittsburgh, Pennsylvania, United States [2] Bosch Center for Artificial Intelligence; Pittsburgh, Pennsylvania, United States. Correspondence to: Jonathan Francis <jon.francis@us.bosch.com>.

Proceedings of the 1st Workshop on Safe Learning for Autonomous Driving, at the 39th International Conference on Machine Learning (ICML'22), Baltimore, Maryland, USA, PMLR 162, 2022.


Responsibilities of conventional end-to-end, data-driven encoders
Detection
Analysis
Vehicles
Traffic Lights
is_at_traffic_light
is_traffic_light_active
is_vehicle_on_same_lane
is_front_vehicle_braking
Ped.
is_pedestrian_hitable
is_pedestrian_on_hit_zone
is_pedestrian_on_near_hit_zone
Measure dist. to lead vehicle
Measure dist. from road centerline
Compute goal hypotheses
Decide: when to follow vehicle vs. waypoints
Consider obstacle-aware goals, outside expert demonstrations
$\mathcal{O}$
Projections + Encoding
Distribution-aware Goal Prediction
Responsibilities of conventional end-to-end, data-driven decoders
Planning
Control + Behaviour
Compute throttle
Compute steering
Compute braking
Adaptive cruising
Dynamics Model
Planning over long horizons
Re-plan, given new information
Speed
adjust speed on turns
adjust speed on hills
adjust speed in crowded scenes
Braking
start slowing on approach
stop at red traffic lights
Turns
compute waypoint angle
execute final steering angle
a
Conformant Model-based Planning
Action-selection

(a) (b)

Figure 1: Issues with end-to-end imitative pipelines. The (a) **red** and (b) **blue** boxes illustrate the scope of responsibilities of conventional data-driven encoders and decoders, respectively, in the overall pursuit of replicating human driving behaviour. These include obstacle detection and scene analysis, and planning and control. Entities with dotted lines indicate behavioural components that lie outside the support of the expert demonstrations in typical learning-to-drive AI tasks, as in CARLA simulation, such as: computing goal alternatives, in response to dynamic obstacle behaviour or re-planning over long horizons when presented with new route information. As a result, it is neither possible for end-to-end imitative models to recover these skills from data, nor for them to exhibit the necessary degree of internal specialisation, without role assignment through the adoption of modular training. Our modular decomposition scheme (bottom arrows) is motivated by this taxonomy, as well as by the shortcomings of alternative decomposition schemes.

ence (e.g., in the form of expert demonstrations), while also achieving generalisability to novel scenarios? Some recent works from the trajectory forecasting community formulate a dual-objective optimisation, coupling an imitation objective with a goal likelihood term (Rhinehart et al., 2018a; 2019; Park et al., 2020), arguing that the two ideals of using prior expert experience and generalising can be unified. A common issue with this formulation is that models are incentivised to *trade-off* the two objectives, rather than inherit their individual benefits. Samples from the goal distribution may not be sufficiently diverse, if the expert demonstrations did not provide sufficient coverage over the modes in the distribution over all possible predictions. Furthermore, predictions may not be admissible, discussed by Park et al. (2020); Francis et al. (2021), without some bias to adhere to, e.g., known physical constraints, as in *Verlet integration* (Verlet, 1967). In this work, we utilise expert demonstrations for pre-training sub-modules and for density estimation, but we also ground predictions on a differentiable vehicle kinematics model and we constrain predictions to respect road admissibility through geometrical projection of goal prediction.

As a summary of our contributions, we produce a framework for generating diverse multi-mode predictions, for the learning-to-drive setting, that achieves improved generalisability through modular task structures, more informed goal likelihood density-estimation, and explicit grounding on differentiable vehicle kinematics for trajectory generation. Our approach is summarised in Figure 2. First, (i) we define modular primitives, based on insights about the decomposable nature of human driving behaviour. Next, (ii) we pursue model generalisability by coupling an imitation prior objective with a goal likelihood term, enabling the agent to leverage expert knowledge, while modelling more diverse modes in the underlying distribution over all goal futures. Next, (iii) we ground candidate trajectories on vehicle kinematics. Finally, under CARLA simulation, (iv) we report new state-of-the-art results on the CARNOVEL benchmark.

2. Related Work

In this section, we describe prior art that is closely related to the core attributes of our approach.

Learning to drive. Pomerleau (1989) pioneered investigation of end-to-end imitation learning, for sensorimotor navigation in autonomous driving. Following some extensions (Muller et al., 2006; Silver et al., 2010; Bojarski et al., 2016) with applications in lane-following, highway driving, and obstacle avoidance, more recent works adapted the classic imitative modelling approach to urban driving scenarios (Codevilla et al., 2018; Bansal et al., 2018; Codevilla et al., 2019; Pan et al., 2017; Ohn-Bar et al., 2020; Chen et al., 2020b), with more complex road layouts and challenging dynamic obstacle interactions. Whereas the increased sample-efficiency from imitation allays much serious consideration of alternative learning paradigms, e.g., reinforcement, a common issue with imitative modelling arises from having to learn a representation from high-dimensional visual inputs, in highly-varying environments: even with sufficient data, models struggle to extract meaningful features from the input that are not confounded by high-frequency, label-independent variation (e.g., varied vehicle shapes, sensor miscalibration, different weather conditions, shadows, poor expert behaviour) (Bansal et al., 2018). In fact, access to more samples can actually yield worse performance, as low-quality data can lead the model to misidentify basic causal structures, underlying expert-environment interactions (de Haan et al., 2019). Following Codevilla et al. (2018; 2019), Ohn-Bar et al. (2020); Chen et al. (2020b); Sauer et al. (2018) use conditioning strategies, such as command variables, teacher networks, and mixtures of expert policies, in attempts to learn better conditional representations and thus reduce the search space for generating actions. However, a limitation of these works is that the number of modes that can be represented by these methods is limited by the number of pre-specified commands or experts—thereby limiting the model's generalisability to novel driving scenes.

Control strategies for autonomous vehicles. Aside from learning (e.g., neural) mappings from observations to actions, various works advocate for the use of feedback or model-based control: to simplify the learning process for the data-driven components of the framework, to replace the data-driven components entirely, or to ground neural predictions with explicit physical constraints. Chen et al. (2020b) utilise a proportional-integral-derivative (PID) controller to track the agent's target velocity, while Sauer et al. (2018) use a PID controller for longitudinal tracking and a Stanley Controller (SC) (Thrun et al., 2006) for lateral tracking, with respect to the road centerline. A feedback controller *myopically* and *reactively* determines its control actions based on deviations from the setpoints, whereas model-based controllers, such a model-predictive controller (MPC), can plan trajectories over long planning horizons by unrolling its model of the system dynamics. Kabzan et al. (2019); Herman et al. (2021); Chen et al. (2021); Francis et al. (2022) implement MPC controllers for their autonomous racing tasks, using ground-truth vehicle states. In this work, we integrate our perception and goal-prediction modules with an MPC, in the context of urban driving, enabling our system to generate trajectories that conform to vehicle kinematics.

Trajectory forecasting for autonomous driving. The notion of characterising distributions over all possible agent predictions has seen exciting growth in the domain of trajectory forecasting for autonomous driving (Lee et al., 2017; Rhinehart et al., 2018a;b; 2019; Park et al., 2020; Filos et al., 2020). Whereas Lee et al. (2017) use past trajectories and scene context as input for predicting future trajectories, and they score the 'goodness' of a trajectory as a learnable module, their method does not attempt to model the agent's predictive intent, e.g., as modes in a likelihood density. Rhinehart et al. (2018b) incorporated the concept of *goal-likelihood* into their Deep Imitative Model (DIM), and characterised the agent's objective via pre-specified geometric primitives: points, piece-wise linear segments, and polygons. However, their goal-likelihood is defined as simple set membership (i.e., within the pre-specified geometry or not). Intuitively, set membership is neither a necessary or sufficient condition for good driving behaviour (e.g., banking vs following waypoints; avoiding obstacles vs staying on waypoints; staying within drivable area vs. driving safely). Filos et al. (2020) aimed to improve on DIM by evaluating the trajectory on the basis of an ensemble of expert likelihoods; while that gives a more robust estimate of the 'goodness' of a trajectory, it neither considers dynamic obstacles nor more informative goal priors. Whereas multi-agent trajectory forecasting has slightly different intentions and implications than action planning in learning-to-drive settings, we nonetheless draw inspiration from the trajectory forecasting literature for their distributional interpretation of the ego-agent's intent, which we combine with information about the scene context, for improved obstacle-awareness in those predicted trajectories.

3. Problem Definition

We define, here, the terminology that we will use to characterise our problem. The ego-agent is a dynamic, on-road entity whose state is characterised by a 4D pose: the spatial position (consisting of x, and y in a Cartesian world coordinate frame), the speed v, and the yaw angle θ, which evolves over time. For the position of the ego-agent at control time-step k, we use the notation $\mathbf{x}_k = [x_k, y_k, v_k, \theta_k] \in \mathbb{R}^4$; for the agent's sequence of positions, from time-step k_1 to k_2, we use $\mathbf{x}_{k_1:k_2}$. For the full sequence of the ego-agent's positions, for a single episode in the training data, we use (bold) $\boldsymbol{X}$. Setting k_0 as the present state, we define the agent's historical trajectory $t \leq k_0$ to be $\boldsymbol{X}_{\text{past}}$ and the agent's future trajectory (again, from the expert demonstrations) $t \geq k_0$ to be $\boldsymbol{X}_{\text{future}}$. At each control time-step, k, the agent is provided with contextual information from the environment, such as a frontal camera view $\boldsymbol{\Phi} \in \mathbb{R}^{H \times W \times C}$ and a sequence of waypoints $\boldsymbol{\omega}$. Combining $\boldsymbol{X}_{\text{past}}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\omega}$ we have the agent's observation, or simply $\mathbf{O} \equiv \{\boldsymbol{X}_{\text{past}}, \boldsymbol{\Phi}, \boldsymbol{\omega}\}$.

At each time-step, the agent must take an action $\mathbf{u}_k$, defined as a tuple of *braking*, *throttling*, and *steering* control. Our objective is to learn a parameterised policy π_θ that maps observations to actions $\mathbf{u} \sim \pi_\theta(\cdot|\mathbf{O})$, such that, given a sequence of observations, an agent that begins at some initial location in the environment can drive to some destination.

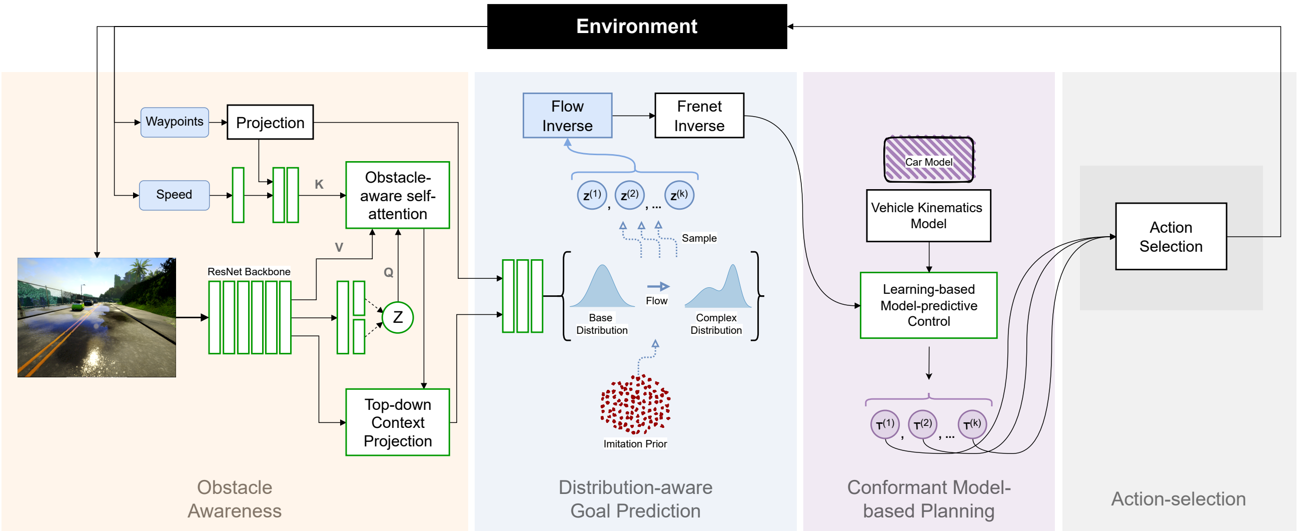

Environment
Waypoints
Projection
Speed
K
Obstacle-aware self-attention
V
Q
ResNet Backbone
Z
Top-down Context Projection
Obstacle Awareness
Flow Inverse
Frenet Inverse
$z^{(1)}$, $z^{(2)}$, ..., $z^{(k)}$
Sample
Base Distribution
Flow
Complex Distribution
Imitation Prior
Distribution-aware Goal Prediction
Car Model
Vehicle Kinematics Model
Learning-based Model-predictive Control
$\tau^{(1)}$, $\tau^{(2)}$, ..., $\tau^{(k)}$
Conformant Model-based Planning
Action Selection
Action-selection

Figure 2: Model architecture. Our framework uses the ego-centric sequence of RGB images, world-frame waypoints, and the agent's own current speed information to learn obstacle-aware attention maps and top-down visual representations. These scene encodings inform distribution-aware goal prediction, to leverage expert experience for generalisability to novel scenarios. A set of candidate goal predictions are realised as trajectories, each transformed to the *Frenet* road frame coordinate system and grounded to vehicle kinematics, using a differentiable MPC controller. Trajectories are pruned using a learnable ranking and refinement module.

4. Approach

Urban driving can be modelled as a composition of driving primitives, where, through decomposition of the conventional multimodal perception backbone into hierarchical units and through modular training, we enjoy lower sample-complexity and improved robustness and generalisability, compared to end-to-end policies.

4.1. Problem Formulation

We propose a modular pipeline that performs **D**istribution-aware **G**oal **P**rediction, with conformant model-based planning, in urban driving settings (DGP). It primarily consists of three components: an *obstacle-awareness* module, a *distribution-aware goal prediction* module, and a *conformant model-based planning* module, as illustrated in Figures 1 (decomposition) and 2 (overview). More formally, we factorise the predictive distribution over actions, as a more tractable mapping: $\mathcal{MPC} \circ \mathcal{GP} \circ \mathcal{OA}$. Here, $\mathbf{m} \sim \mathcal{OA}(\cdot|\mathbf{O})$ is an obstacle-awareness module, which generates an embedding $\mathbf{m} \in \mathbb{R}^{256}$, given an observation. $\mathbf{x}_{\text{goal}} \in \mathbb{R}^4 \sim \mathcal{GP}(\cdot|\mathbf{m})$ is a goal prediction module, whose samples are desired to be diverse in their coverage of the modes in the true, underlying goal distribution $p^\star(\mathbf{x}_{\text{goal}}|\mathbf{m})$. Here, $\mathbf{x}_{\text{goal}}$ is the predicted goal from the $\mathcal{GP}$ module and $\mathbf{x}^\star_{\text{goal}}$ is the true (unobserved) goal of the expert agent, which characterises its scene-conditioned navigational intent. We want $\mathcal{GP}$ to generate multiple samples, where each sample can be regarded as an independent hypothesis of what might have happened, given the same observation. $\mathbf{x}_{k+1:k+N}, \mathbf{u}_{k+1:k+N} = \mathcal{MPC}(\mathbf{x}_{\text{goal}})$ is a learning-based MPC, which takes a predicted goal from the goal distribution as input and generates a trajectory that reaches $\mathbf{x}_{\text{goal}}$ in N time-steps based on its embedded vehicle model.

Our framework uses the ego-centric sequence of RGB images, world-frame waypoints, and the agent's own current speed information to learn obstacle-aware attention maps and top-down visual representations. These scene encodings inform our goal prediction module, which combines an imitation prior and a goal likelihood objective, in order to leverage expert experience for generalisability to novel scenarios. A set of candidate goal predictions are realised as trajectories, each transformed to the *Frenet* road frame coordinate system and grounded to vehicle kinematics, using a differentiable MPC.

4.2. Obstacle Awareness: Projections and Encoding

We condition the learning of our goal distribution on crucial scene context — from projected topdown feature representation and obstacle self-attention. This perception module's task is to transform the front-view image observations into bird's eye view (BEV) semantic object attention maps.

In this work, we leverage the orthographic feature transform (OFT) technique developed by (Roddick et al., 2018). In particular, we extract obstacle semantic information by pre-training a variational autoencoder (Kingma & Welling, 2014) to reconstruct pixels, speed, and steering control in the next time step from current observations. It encourages the latent variables to attend to obstacle in front view (e.g., vehicles, pedestrians, traffic lights, curbs, etc.) which impact future vehicle control. The front-view feature map $f(u, v)$ is constructed by combining the learned self-attention maps (Vaswani et al., 2017) with multi-scale images features of pre-trained ResNet-18 front-end. Then, voxel-based features $g(x, y, z)$ are generated by accumulating image-based features $f(u, v)$ to a uniformly spaced 3D lattice $\mathcal{G}$ fixed to the ground plane a distance y_p below the camera and has dimensions W, H, D and a voxel resolution of r using orthogonal transformation. Finally, the topdown image feature representation $h(x, z)$ is generated by collapsing the 3D voxel feature map along the vertical dimension through a learned 1D convolution. In addition to image features, we interpolate waypoint sequence and create a topdown grid representation of waypoints with one-hot encoding. The final topdown feature representation is of dimension $[W/r, D/r, C]$ where the number of channels $C = C_{\text{attn}} + C_{\text{resnet}} + 1$.

4.3. Multi-mode Goal Distribution

We wish to approximate the true predictive distribution over all possible goal futures of the ego-agent, $p^{\star}(\mathbf{x}_{\text{goal}}|\mathbf{m})$, given the embedding vector $\mathbf{m}$ from the obstacle awareness module (§4.2) based on the observation $\mathbf{O}$ from the environment. Unfortunately, the predictive *intent* of the expert agent is not observable from the training data: the expert may make multiple manoeuvres given the same scenario, but we only observe the expert taking one of the options. As a proxy, we take a future state of the expert agent, at fixed time horizon N, to be the "ground-truth" ego-agent's goal, $\mathbf{x}_{\text{goal}} \equiv \mathbf{x}_{k_0+N}$, where N denotes the number of time-steps in the planning horizon.

Next, rather than learning a mapping to directly imitate these derived expert goals, we instead model an *approximation* $p_\theta(\mathbf{x}_{\text{goal}}|\mathbf{m})$ of the underlying goal distribution, by leveraging a bijective and differentiable mapping between a chosen *base distribution* p_0 and the aforementioned target approximate goal distribution p_θ. This technique is commonly referred to as a 'normalizing flow', which provides a general framework for transforming a simple probability density (base distribution) into a more expressive one, through a series of invertible mappings (Tabak et al., 2010; Rezende & Mohamed, 2015; Papamakarios et al., 2021; Kingma & Dhariwal, 2018; Park et al., 2020).

Formally, let f be an invertible and smooth function, with $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $\mathbf{x} = f(\mathbf{z})$, $\mathbf{z} \sim p_{\mathbf{z}}$, $f^{-1} = g$, and thus $g \circ f(\mathbf{z}) = \mathbf{z}$, for d-dimensional random vectors $\mathbf{x}$ and $\mathbf{z}$. In this case, $d = 4$ and $p_{\mathbf{z}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Further, we attribute to f the property of *diffeomorphism* (Milnor & Weaver, 1997), which ensures that $q_{\mathbf{x}}$ remains well-defined and obtainable through a change of variables, and ensures that $p_{\mathbf{z}}$ is uniformly distributed on the same domain as the data space (Liao & He, 2021) — insofar as both f and its inverse f^{-1} are differentiable and that $\mathbf{z}$ retains the same dimension as $\mathbf{x}$:

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{z}}(\mathbf{z}) \left|\det \tfrac{\partial f}{\partial z}\right|^{-1} \tag{1}$$

We can construct arbitrarily complex densities, by *flowing* $\mathbf{z}$ along the path created by a chain of K successive *normalizing* distributions $p_{\mathbf{z}}(\mathbf{z})$, with each successive distribution governed by a diffeomorphic transformation:

$$\mathbf{x} = \mathbf{z}_K = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z_0}) \tag{2}$$

Following this sequence of transformations, our main interfaces with the flow-based model are through either sampling or evaluating its density, where, in the former, we sample from $p_{\mathbf{z}}(\mathbf{z})$ and must compute the forward transformation f; in the latter, we must compute the inverse transformation f^{-1}, its Jacobian determinant, and the $p_{\mathbf{z}}(\mathbf{z})$ density evaluation. The learning objective of the goal prediction model is given by Eqn. 3, which is to minimise the KL divergence between the true and approximate goal distribution, or equivalently maximise the log-likelihood of the approximate goal distribution:

$$\begin{aligned} \min_\theta \mathrm{KL}(p^\star || p_\theta) &= \max_\theta \mathbb{E}_{x \sim p^\star} \log p_\theta(\mathbf{x}) \\ &\approx \max_\theta \sum_{x \sim \mathcal{D}} \log p_\theta(\mathbf{x}) \end{aligned} \tag{3}$$

The likelihood is given in in Eqn. 1, which is made tractable through the special design of the bijective transformation f. Since we are learning a conditional distribution, we also pass the context embedding $\mathbf{m}$ to the flow model, following prior works (Dinh et al., 2016; Papamakarios et al., 2017).

4.4. Grounding Action Predictions with Road Geometry and Vehicle Dynamics

Given $\mathbf{x}_{\text{goal}} \sim \mathcal{GP}(\cdot|\mathbf{m})$, we want to find a trajectory to $\mathbf{x}_{\text{goal}}$ that is conformant to both vehicle dynamics and road geometry. To conform to road geometry, we predict the goal state in Frenét coordinate (Figure 3) and invert to Cartesian coordinate. We parameterise the road geometry as a cubic spline along the way points, and calculate the Frenét and inverse-Frenét transformation accordingly.

To be conformant to the vehicle dynamics, we formulate reaching a desired goal state as a MPC problem (Eqn. 4), which embeds a vehicle model. The objective (Equation 4a) is to reach the goal with regularisation on actuations. $\mathbf{Q}$ and $\mathbf{R}$ are both diagonal matrices corresponding to cost weights for tracking reference states and regularising actions. At the

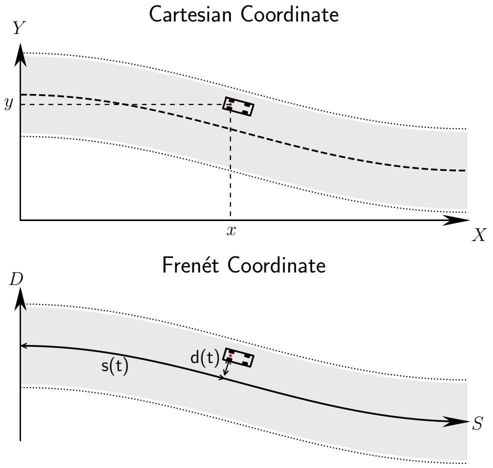


Figure 3: Comparison of Cartesian and Frenét coordinates.

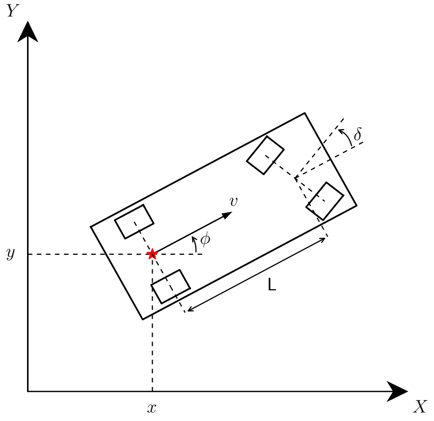


Figure 4: Illustration of the kinematic bike model.

same time, the MPC respects the system dynamics of the vehicle (Equation 4b), and allowable actuation (Eqn. 4c).

$$\min_{\mathbf{u}_{1:N}} \quad (\mathbf{x}_N - \mathbf{x}_{\text{goal}})^T \mathbf{Q} (\mathbf{x}_N - \mathbf{x}_{\text{goal}}) + \sum_{k=1}^{N} \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \tag{4a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \ldots, N \tag{4b}$$

$$\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \bar{\mathbf{u}} \tag{4c}$$

Specifically, we characterise the vehicle with the kinematic bike model (Kong et al., 2015) given in Eqn. 5[1] and visualised in Figure 4, where the state is $\mathbf{x} = [x, y, v, \phi]$, and the action is $\mathbf{u} = [a, \delta]$. a is the acceleration, and δ is the steering angle at the front axle. A key challenge is that the ground truth vehicle parameters were not known to us. Aside from L defined as the distance between the front and rear axle, the kinematic bike model expects actions, i.e. acceleration and steering, in physical units, while the CARLA simulator expects normalised commands. The mapping is unknown to us, and thus, we learn the mapping by minimising the prediction error between predicted and actual trajectories from expert demonstrations.

$$\dot{x} = v \cos(\phi) \tag{5a}$$

$$\dot{y} = v \sin(\phi) \tag{5b}$$

$$\dot{v} = a \tag{5c}$$

$$\dot{\phi} = v \tan \delta / L \tag{5d}$$

We solve the MPC problem with the iterative linear quadratic regulator (iLQR) proposed in (Li & Todorov, 2004), which iteratively linearises the non-linear dynamics along the current estimate of trajectory, solves a linear quadratic regulator (LQR) problem based on the linearized dynamics, and repeats the process until convergence. Specifically, we used the implementation for iLQR from (Amos et al., 2018).

5. Experiments

We evaluate our approach using the CARLA (Dosovitskiy et al., 2017) simulator, with specific focus on three challenge benchmarks: CARNOVEL (Filos et al., 2020), NoCrash (Codevilla et al., 2019), and the original CARLA benchmark (Dosovitskiy et al., 2017). In all settings, the agent is provided with an RGB image, a waypoint sequence, and vehicle speed; the agent must produce steering, throttle, and braking control, in order to navigation to a destination.

All experiments were conducted using CARLA simulator version 0.9.6, which is worth noting because this version introduced updates to the rendering engine and pedestrian logic, allowing for consistency across various benchmarks but making the results of contemporary approaches on previous simulator versions no longer comparable (Chen et al., 2020b). We used a dual-GPU machine, with the following CPU specifications: Intel(R) Core(TM) i9-9920X CPU @ 3.50GHz; 1 CPU, 12 physical cores per CPU, total 24 logical CPU units. The machine includes two NVIDIA Titan RTX GPUs, each with 24GB GPU memory.

In this section, we further describe the tasks, baselines, research challenges, and ablations.

Benchmark tasks. We follow Filos et al. (2020) in assessing the robustness of modelling approaches to novel, OOD driving scenarios in their CARNOVEL benchmark. Predicated on the CARLA simulator (Dosovitskiy et al., 2017), agents are first trained on the provided offline expert demonstration from `Town01` that were originally generated using a rules-based autopilot. Agents are then evaluated on various OOD navigation tasks, such as: abnormal turns, busy-town settings, hills, and roundabouts. Agents' performance are

[1] This set of equations is defined with respect to the back axle of the vehicle.

Table 1: Results of baseline models and our proposed approach on CARNOVEL (Filos et al., 2020). We report Success Rate (↑; M × N scenes, %), the Number of Infractions (Collisions, Lane Invasions) per kilometer (↓; $\times 1e^{-3}$), and Distance travelled (m), on four novel (unseen) scenarios.

	ABNORMALTURNS			BUSYTOWN		
Method	Success (↑) (M × N scenes, %)	Infra/km (↓) ($\times 1e^{-3}$)	Distance (m)	Success (↑) (M × N scenes, %)	Infra/km (↓) ($\times 1e^{-3}$)	Distance (m)
CIL (Codevilla et al., 2018)	65.71 ± 7.37	7.04 ± 5.07	128 ± 20	5.45 ± 6.35	11.49 ± 3.66	217 ± 33
LBC (Chen et al., 2020b)	0.0 ± 0.0	5.81 ± 0.58	208 ± 4	20.00 ± 13.48	3.96 ± 0.15	374 ± 16
DIM (Rhinehart et al., 2018b)	74.28 ± 11.26	5.56 ± 4.06	108 ± 17	47.13 ± 14.54	8.47 ± 5.22	175 ± 26
RIP (Filos et al., 2020)	**87.14** ± 14.20	4.91 ± 3.60	102 ± 21	62.72 ± 5.16	3.17 ± 2.04	167 ± 21
DGP (*ours*)	82.86 ± 6.39	**1.49** ± 1.44	108.74 ± 8.70	**76.36** ± 4.98	**5.51** ± 2.34	126.83 ± 6.73

	HILLS			ROUNDABOUTS		
Method	Success (↑) (M × N scenes, %)	Infra/km (↓) ($\times 1e^{-3}$)	Distance (m)	Success (↑) (M × N scenes, %)	Infra/km (↓) ($\times 1e^{-3}$)	Distance (m)
CIL (Codevilla et al., 2018)	60.00 ± 29.34	4.74 ± 3.02	219 ± 34	20.00 ± 0.0	3.60 ± 3.23	269 ± 21
LBC (Chen et al., 2020b)	50.00 ± 0.0	1.61 ± 0.15	514 ± 101	8.00 ± 10.95	3.70 ± 0.72	323 ± 43
DIM (Rhinehart et al., 2018b)	70.00 ± 10.54	6.87 ± 4.09	195 ± 12	20.00 ± 9.42	6.19 ± 4.73	240 ± 44
RIP (Filos et al., 2020)	87.50 ± 13.17	1.83 ± 1.73	191 ± 6	42.00 ± 6.32	**4.32** ± 1.91	217 ± 30
DGP (*ours*)	**100** ± 0.0	**0.0** ± 0.0	193.12 ± 0.03	**80.00** ± 14.14	4.44 ± 7.21	126.32 ± 11.83

measured according to the following metrics: success rate (percentage of successful navigations to the destination), infractions per kilometre (ratio of moving violations to kilometre driven), and total distance travelled.

Baselines models. We compare our model with the following baselines in the CARNOVEL benchmark:

Conditional imitation learning (CIL) (Codevilla et al., 2018) is an end-to-end behaviour cloning approach, which conditions its predictions on high-level commands and LiDAR information.

Learning by cheating (LBC) (Chen et al., 2020b) extends CIL through cross-modal knowledge distillation, from a teacher network (trained on privileged information — e.g., environment state, overhead images) to a sensorimotor navigation agent (the ego-agent).

Deep Imitative Model (DIM) (Rhinehart et al., 2018b) is a trajectory forecasting and control method, which combines an imitative objective with goal-directed planning.

Robust Imitative Planning (RIP) (Filos et al., 2020) is the method that was proposed alongside the recent CARNOVEL benchmark. RIP is an epistemic uncertainty-aware method, targeted toward robustness to OOD driving scenarios.

6. Results

Generalisation to OOD scenes. In Table 1, we report the success rate, number of infractions per kilometer, and distance travelled of our approach, alongside strong baselines from both the learning-to-drive and trajectory forecasting communities. We show significant improvements in unseen generalisation in visuomotor control for urban driving from our approach, which jointly estimates the ego-agent's action distribution while learning to predict and score intermediate goals. Notably, we observe most significant improvements over the next-best model, Robust Imitative Planning (RIP; Filos et al. (2020); an epistemic uncertainty-aware model) when transferring models to completely unseen traffic layouts, such as Roundabouts.

Qualitative results on agent intention. A notable benefit of our factorising the urban driving problem—into encoding, distribution-aware goal prediction, conformant model-based planning, and pruning + action-selection—is that we obtain increased interpretability in our agent's prediction pipeline. Specifically, by way of our goal-prediction module, we obtain the ability to reveal agent's intent, during its simulated task execution. Figure 5 offers qualitative results from our agent's interaction with the environment during inference. Each frame captures the agent's prediction (in red), given its own speed information, ego-vision context, and short waypoint sequences (in mustard).

We observed that waypoint sequences, provided by the environment, may sometimes change, as agents approach locations where some decision must be made (e.g., turns). Coupled with our models estimation of the underlying action distribution, conditioned on offline dataset samples, our model correctly makes multiple admissible predictions of turning (a), going straight (b), changing lanes (c), or stopping/slowing (d) as it approaches the first intersection. After committing and executing the turn (e-f), the agent once again considers multiple possible futures (g), before once again following the rightward arcing waypoint sequence. On

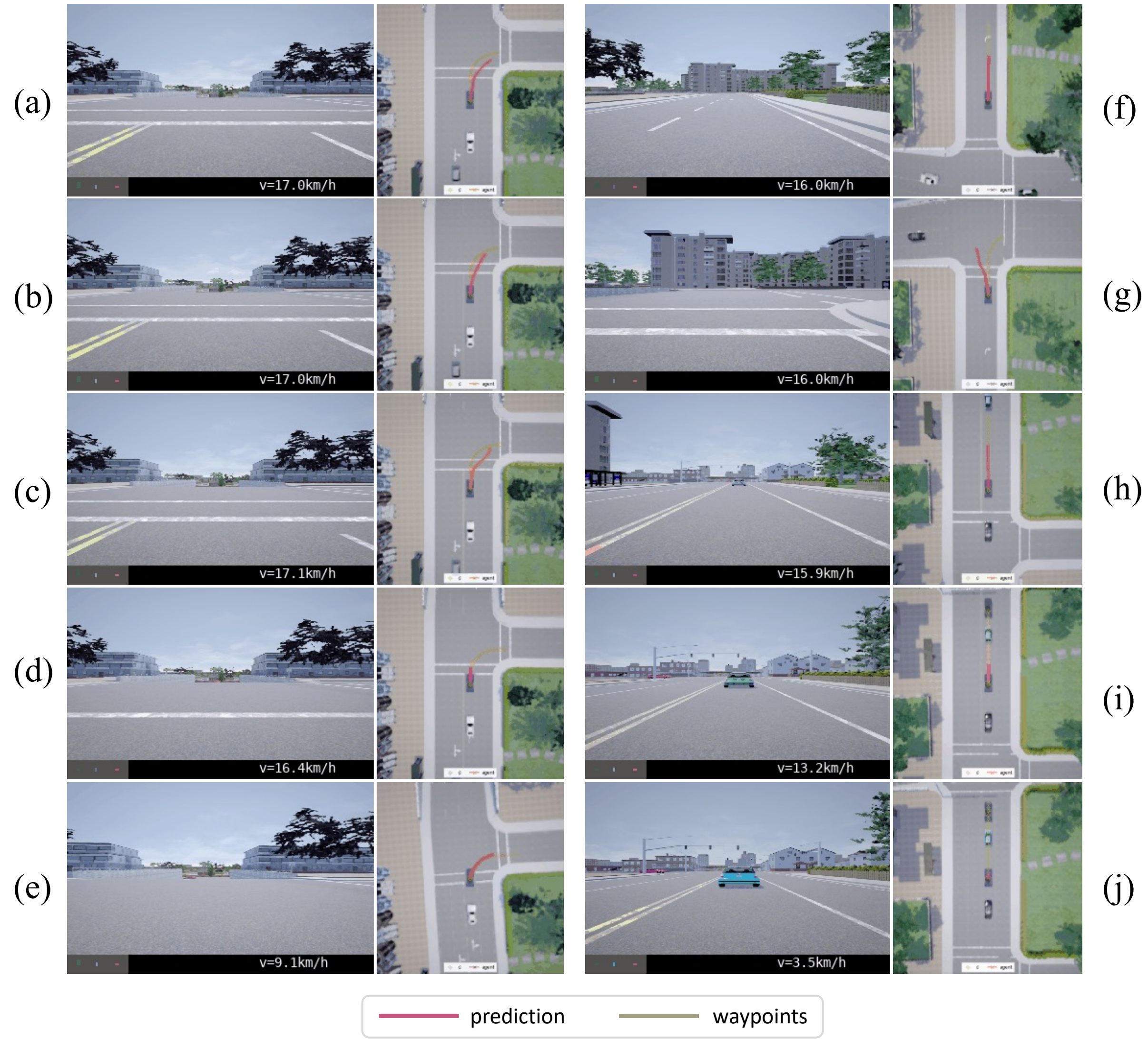


Figure 5: Qualitative results from simulated task execution.

straight sections, the agent shows strong belief on forward movement, indicated by distant and straight goal predictions (h), but correctly slows (i) and stops (j) for obstacles.

7. Discussion and Conclusion

In this paper, we introduce a distribution-aware trajectory generation mechanism that remains conformant to both road geometry and vehicle kinematics. We show how our agent uses this learned prior to generalise to completely out-of-distribution driving scenarios, such as busy towns, abnormal turns, hills, and unseen traffic patterns such as roundabouts.

References

Amos, B., Rodriguez, I. D. J., Sacks, J., Boots, B., and Kolter, J. Z. Differentiable mpc for end-to-end planning and control. *arXiv preprint arXiv:1810.13400*, 2018.

Bansal, M., Krizhevsky, A., and Ogale, A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Chaplot, D. S., Gandhi, D., Gupta, S., Gupta, A., and Salakhutdinov, R. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

Chen, B., Francis, J., Oh, J., Nyberg, E., and Herbert, S. L. Safe autonomous racing via approximate reachability on ego-vision, 2021.

Chen, C., Majumder, S., Al-Halah, Z., Gao, R., Ramakrishnan, S. K., and Grauman, K. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, 1(2):6, 2020a.

Chen, D., Zhou, B., Koltun, V., and Krähenbühl, P. Learning by cheating. In *Conference on Robot Learning*, pp. 66–75. PMLR, 2020b.

Chen, T., Gupta, S., and Gupta, A. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*, 2019.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

Codevilla, F., Santana, E., López, A. M., and Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9329–9338, 2019.

de Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning, 2019.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Filos, A., Tigkas, P., McAllister, R., Rhinehart, N., Levine, S., and Gal, Y. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2020.

Francis, J., Kitamura, N., Labelle, F., Lu, X., Navarro, I., and Oh, J. Core challenges in embodied vision-language planning. *arXiv preprint arXiv:2106.13948*, 2021.

Francis, J., Chen, B., Ganju, S., Kathpal, S., Poonganam, J., Shivani, A., Vyas, V., Genc, S., Zhukov, I., Kumskoy, M., Oh, J., Nyberg, E., and Herbert, S. L. Learn-to-race challenge 2022: Benchmarking safe learning and cross-domain generalisation in autonomous racing, 2022.

Herman, J., Francis, J., Ganju, S., Chen, B., Koul, A., Gupta, A., Skabelkin, A., Zhukov, I., Kumskoy, M., and Nyberg, E. Learn-to-race: A multimodal control environment for autonomous racing. *arXiv preprint arXiv:2103.11575*, 2021.

Kabzan, J., Hewing, L., Liniger, A., and Zeilinger, M. N. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4): 3363–3370, 2019.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Kong, J., Pfeiffer, M., Schildbach, G., and Borrelli, F. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099. IEEE, 2015.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–345, 2017.

Li, W. and Todorov, E. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, pp. 222–229. Citeseer, 2004.

Liao, H. and He, J. Jacobian determinant of normalizing flows. *arXiv preprint arXiv:2102.06539*, 2021.

Milnor, J. and Weaver, D. W. *Topology from the differentiable viewpoint*. Princeton university press, 1997.

Muller, U., Ben, J., Cosatto, E., Flepp, B., and Cun, Y. L. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pp. 739–746. Citeseer, 2006.

Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., and Geiger, A. Learning situational driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11296–11305, 2020.

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., and Boots, B. Agile autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Park, S. H., Lee, G., Bhat, M., Seo, J., Kang, M., Francis, J., Jadhav, A. R., Liang, P. P., and Morency, L.-P. Diverse and admissible trajectory forecasting through multimodal context understanding. In *European Conference on Computer Vision (ECCV)*, 2020.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY ..., 1989.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rhinehart, N., Kitani, K. M., and Vernaza, P. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 772–788, 2018a.

Rhinehart, N., McAllister, R., and Levine, S. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*, 2018b.

Rhinehart, N., McAllister, R., Kitani, K., and Levine, S. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2830, 2019.

Roddick, T., Kendall, A., and Cipolla, R. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

Sauer, A., Savinov, N., and Geiger, A. Conditional affordance learning for driving in urban environments. In *Conference on Robot Learning*, pp. 237–252. PMLR, 2018.

Silver, D., Bagnell, J. A., and Stentz, A. Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12):1565–1592, 2010.

Tabak, E. G., Vanden-Eijnden, E., et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9): 661–692, 2006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Verlet, L. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.