Learning Situational Driving

Eshed Ohn-Bar[1,3] Aditya Prakash[1] Aseem Behl[1,2] Kashyap Chitta[1,2] Andreas Geiger[1,2]
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen [3]Boston University
{firstname.lastname}@tue.mpg.de

Abstract

Human drivers have a remarkable ability to drive in diverse visual conditions and situations, e.g., from maneuvering in rainy, limited visibility conditions with no lane markings to turning in a busy intersection while yielding to pedestrians. In contrast, we find that state-of-the-art sensorimotor driving models struggle when encountering diverse settings with varying relationships between observation and action. To generalize when making decisions across diverse conditions, humans leverage multiple types of situation-specific reasoning and learning strategies. Motivated by this observation, we develop a framework for learning a situational driving policy that effectively captures reasoning under varying types of scenarios. Our key idea is to learn a mixture model with a set of policies that can capture multiple driving modes. We first optimize the mixture model through behavior cloning and show it to result in significant gains in terms of driving performance in diverse conditions. We then refine the model by directly optimizing for the driving task itself, i.e., supervised with the navigation task reward. Our method is more scalable than methods assuming access to privileged information, e.g., perception labels, as it only assumes demonstration and reward-based supervision. We achieve over 98% success rate on the CARLA driving benchmark as well as state-of-the-art performance on a newly introduced generalization benchmark.

1. Introduction

Realizing highly accurate and fail-safe autonomous vehicles that can handle the range of perceptual and situational complexities of driving has challenged researchers for decades. For instance, the systems' perception-to-action reasoning must flexibly accommodate both normal highway driving on a sunny day, as well as driving in a busy intersection full of pedestrians on a rainy day, where lane markings may not even be visible. To drive in such diverse scenarios, humans leverage different types of situation-specific strategies and contextual cues [11], e.g., identifying the need to slow-down and follow scene-level cues if lane information is not available. Moreover, drivers leverage combinations of driving strategies, in particular when encountering a novel scenario [26]. How can we endow machines with similar reasoning and learning capabilities, crucial for operating under the vast diversity of all possible visual, planning, and control scenarios?

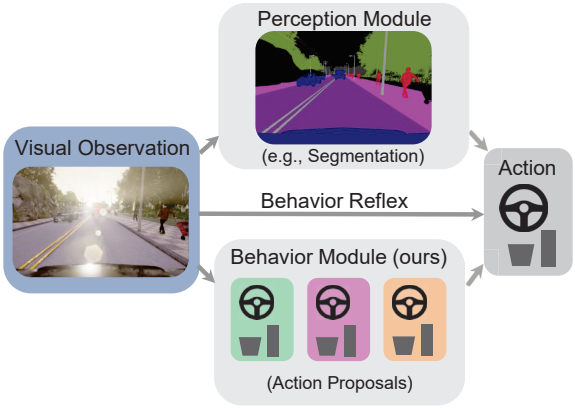

Figure 1: **Situational Driving.** To address the complexity in learning perception-to-action driving models, we introduce a situational framework using a behavior module. The module reasons over current on-road scene context when composing a set of learned behavior policies under varying driving scenarios. Our approach is used to improve over behavior reflex and privileged approaches in terms of robustness and scalability.

Towards addressing this question, several learning paradigms have been previously proposed. On one hand, the complex task of mapping visual observations to a control action can be decomposed into modules or subtasks using dedicated auxiliary loss functions, i.e., addressing the perception and action tasks as two modules (e.g., [4,28,37]). Leveraging prior and domain knowledge through hand-engineered modular structures can improve generalization under certain conditions [40], but the training requires additional annotations and the representations might not be optimal when not learned with respect to the actual navigation task. On the other hand, learning sensorimotor

driving directly from visual observations (e.g., with behavior cloning [8, 33]) has recently re-emerged as a compelling solution to autonomous driving because it can leverage flexibly learned representations and easily scale to large corpus of data. However, even with a large corpus of data, the learned representations may fail to generalize beyond the training set, partly due to the minimal structural prior [50, 52]. Moreover, commonly employed behavior cloning techniques [52] optimize a surrogate loss with respect to the driving task, while task-driven reinforcement learning techniques are difficult to employ, e.g., due to sample inefficiency [10, 13].

We seek to decompose the perception-action learning task in a way that best facilitates generalization, e.g., over varying situations, and scalability, i.e., with minimal supervision. Motivated by the observation that the aforementioned perception-action frameworks may be seen as orthogonal to some degree, we propose a module that attempts to leverage the benefits of incorporating compositional structure, and do so without requiring additional annotations beyond demonstrations and rewards. Towards this goal, we make the following *three contributions*: (1) To improve modeling capacity in behavior cloning models, we develop a mixture of experts (MoE) framework for composing a set of situation-specific policy predictors specialized to different components of the driving task, (2) we further analyze the benefits of the situational policy through refinement with task-driven optimization i.e., with respect to the driving task reward, and (3) we demonstrate state-of-the-art performance in vision-based single frame driving on the CARLA benchmark [10].

2. Related Work

We propose learning a driving policy that can effectively leverage different types of perception-action strategies, i.e., a mixture model that is learned to combine the predictions of specialized expert models. Hence, our work is related to research in learning sensorimotor policies for driving through behavior cloning, reinforcement learning, and hierarchical techniques.

Sensorimotor Navigation: Recognizing the fundamental inflexibility in manually structured and fixed representations, Pomerleau [33] explored an end-to-end neural network for sensorimotor driving, in an imitation learning technique that became known as the behavior reflex. This method learns the perception-to-action mapping via supervised learning from driver demonstrations, i.e., using behavior cloning [22, 29, 30, 49]. Due to ease of training, it is employed in several state-of-the-art approaches on the open-source CARLA simulator [8, 10, 25], as shown in Table 1. However, based on our experiments, increasing the difficulty of the perception-action learning task by introducing multiple relationships between observation and control during training leads to models that generalize poorly. Our MoE framework aims to address such issues in model capacity and optimization.

Issues in Behavior Cloning: Recently, Codevilla et al. [8] demonstrated that behavior cloning achieves state-of-the-art performance on the CARLA [10] benchmark. However, even with ample data, learning representations from high-dimensional visual data for perception, planning, and action with a single end-to-end network can be difficult to optimize. The presence of several dataset phenomena, such as bias [8], lack of on-policy experience [5, 35], multiple data modalities, or an expert that is difficult to imitate [5, 15] can all result in poor modeling and generalization performance [12, 40, 46, 50].

Task-Driven Policy Optimization: The optimization of a surrogate imitation loss with respect to the task can result in several undesirable learned driving behaviors. For instance, Codevilla et al. [8] discuss an 'inertia problem,' where the imitation agent gets stuck and never recovers. As the model was not trained with respect to the driving task itself, i.e., timely arrival to the destination, there is no supervisory signal that prevents the learning of such behaviors. We employ an explicit task-based optimization process in addition to imitation learning as it can alleviate such modeling issues. Liang et al. [25] proposed a driving agent learned with reinforcement learning with weights initialized by behavior cloning. Our model significantly outperforms that of [25] and the architecture is quite different as we learn a hierarchical policy where only the compositional module is learned in a task-driven manner while the imitation learning agents are kept frozen. This process greatly improves sample-efficiency since it only updates the composition of the learned policies. In addition to optimized training, our approach aids in encouraging the learned agent to adhere to traffic rules, which is essential for real-world driving.

Structure and Modularity in Driving Policies: Several studies demonstrate the benefit of incorporating hierarchical, situational reasoning in computer vision, e.g., boundary detection [47] and indoor navigation in static environments [42, 53]. The hierarchy enables to effectively decompose the overall learning task into manageable components that can potentially be combined to improve performance in novel settings [42]. Several hierarchical policy learning frameworks have been previously proposed, e.g. option learning [32, 41, 45] and action primitives [9, 45]. Li et al. [24] learns an optimal policy from an ensemble of imperfect teaching drivers, however they do not employ an MoE objective. A close study to ours is by Kipf et al. [20], showing hierarchical reasoning to enable imitation learning models that generalize to new environments and tasks using grid-world navigation and reaching tasks. However, [20]

Table 1: **Comparison with Representative Related Work.** For each approach we show the type of data and supervision assumed. Control refers to whether the agent outputs the control command directly or not, e.g., waypoints for a PID controller.

	Input			Output	Supervision				
Approach	Image	Speed	Video	Control	Image Annotations	Reconstruction	Demonstrations	On-Policy	Reward
CIL [7]	●	●	-	●	-	-	●	-	-
CAL [37]	●	●	●	-	●	-	-	-	-
CIRL [25]	●	●	-	●	-	-	●	●	●
CILRS [8]	●	●	-	●	-	-	●	-	-
LBC [5]	●	●	-	-	●	-	●	●	-
LSD (this work)	●	●	-	●	-	●	●	-	-
LSD+ (this work)	●	●	-	●	-	●	●	●	●

does not employ a mixture density network nor a task-driven optimization process. Moreover, the aforementioned studies have focused on highly simplified visual and situational environments. In contrast, our driving task involves realistic scenes of diverse weathers and dynamic obstacles.

Leveraging Privileged Supervision: Related studies in autonomous driving alleviate the issue of lack of structure through stronger supervision in the form of explicit perception labels and more structured representations (e.g., affordances [4, 37] and perception modules [2, 23, 28, 38, 48, 51, 52]). Sauer et al. [37] learns a low-dimensional intermediate representations set of affordances which are then inputted to a PID controller. However, the approach is not trained end-to-end and performs worse than the behavior cloning baseline of CILRS. Recently, Chen et al. [5] utilized environment layout and traffic participant annotations in order to train a privileged agent for coaching a non-privileged sensorimotor agent, i.e., an instantiation of imitation by coaching [15]. In contrast, our approach assumes no access to such extensive privileged information, while also performing task-driven optimization. Moreover, we directly learn to map to a control command, while [5] relies on a hand-tuned, separate control module. Nonetheless, we do explore visual representations which can be learned without such explicit supervision, i.e. a Variational Auto-Encoder [13, 19] (VAE). Related to this line of research is a study by Srivastava et al. [44], showing that image reconstruction and prediction tasks improve classification performance. Moreover, our MoE approach is complementary to privileged methods, e.g., training a behavior cloning model over intermediate representations with an MoE objective .

3. Method

In this section, we formulate our approach for learning a situational driving model which accommodates multiple types of on-road reasoning and decision-making processes.

Problem Definition: The goal-directed driving task is formulated as a sequential-decision making problem, defined in the context of the CARLA [10] simulator. The objective of the driving agent is to produce a sequence of control actions that result in timely arrival at a predefined destination. The environment provides the current observations $\mathbf{o}_t = [\mathbf{I}_t, v_t] \in \mathcal{O}$ which comprise an image from a front-facing camera and the ego-vehicle speed at the current time step t. In addition, it supplies a categorical variable defining a high-level navigation command $c_t \in \mathcal{C} = \{left, right, straight, follow\}$ which determines the vehicle path at the next intersections. The action space $\mathcal{A} = [-1, 1]^2$ defines the range of the continuous longitudinal and lateral control values. Our goal is to learn a policy $\pi_{\mathbf{\Theta}} \colon \mathcal{O} \times \mathcal{C} \rightarrow \mathcal{A}$ parameterized by $\mathbf{\Theta}$ that determines which action to take at every time step. Once an action is chosen, the environment provides the next observation $\mathbf{o}_{t+1} \sim p(\mathbf{o}_{t+1}|\mathbf{o}_t, \mathbf{a}_t)$.

3.1. Situational Driving Model

We now describe our situational driving model which facilitates efficient learning of diverse driving behaviors, e.g., fast driving in an empty road vs. driving cautiously in dense urban environments. Our policy takes the following form

$$\pi_{\mathbf{\Theta}}(\mathbf{a}|\mathbf{o}, c) = \sum_{k=1}^{K} \underbrace{\alpha_{\boldsymbol{\theta}}^{k}(\mathbf{o}, c)}_{\substack{\text{Mixture} \\ \text{Weights}}} \underbrace{\pi_{\boldsymbol{\theta}}^{k}(\mathbf{a}|\mathbf{o}, c)}_{\substack{\text{Expert} \\ \text{Models}}} + \mathbf{\Psi} \underbrace{\begin{bmatrix} q_{\boldsymbol{\phi}}(\mathbf{I}) \\ v \\ c \end{bmatrix}}_{\substack{\text{Context} \\ \text{Embedding}}} \quad (1)$$

and comprises two main components:

- A mixture model of probabilistic **expert policies** $\Pi = \{\pi_{\boldsymbol{\theta}}^{1}, \ldots, \pi_{\boldsymbol{\theta}}^{K}\}$ with weights $\alpha_{\boldsymbol{\theta}}^{k}$ for combining multiple diverse driving behaviors.
- A **context embedding** $q_{\boldsymbol{\phi}}$ which provides additional image-based context during model optimization and when regressing the final action.

We implement the mixture of experts model and the context embedding model using neural networks with trainable parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. In addition, we learn the

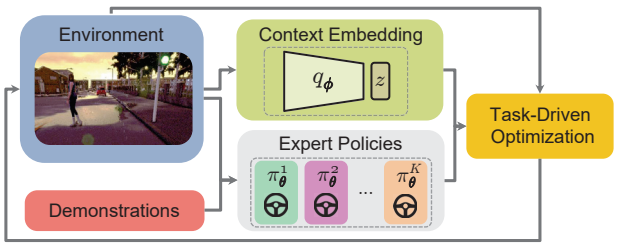

Figure 2: **Approach Overview.** The agent learns to combine a set of expert policies in a context-dependent, task-optimized manner to robustly drive in diverse scenarios.

matrix $\mathbf{\Psi}$ that projects the context features into the two-dimensional action space $\mathcal{A}$. An overview of our framework is provided in Fig. 2, with details on our architecture found in the supplementary.

We now discuss how we decompose the learning problem to learn the model parameters $\mathbf{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{\Psi}\}$ in a data efficient manner.

3.2. Training

Optimizing for the parameters of the driving policy $\pi_{\mathbf{\Theta}}$ is a difficult learning task [21]. In particular, training requires learning to map high-dimensional visual observations to a two dimensional control output, i.e., implicitly and jointly learning representations for performing perception, planning, and control. Moreover, the policy should ideally be optimized directly for the task at hand, i.e., timely arrival to a destination in the map while minimizing infractions, through interaction with the environment. However, learning the policy in this manner is inefficient due to long rollout times in simulation and the large number of parameters that must be optimized. We therefore propose to learn our policy $\pi_{\mathbf{\Theta}}$ in three steps:

1. Learning expert policies $\{\alpha_{\boldsymbol{\theta}}^k, \pi_{\boldsymbol{\theta}}^k\}$ via imitation.
2. Learning of the context embedding $q_{\boldsymbol{\phi}}$.
3. Task-driven learning / refinement of $\mathbf{\Psi}$ and $\{\alpha_{\boldsymbol{\theta}}^k\}$.

While the first step uses expert demonstrations for supervision, the second step requires only raw image sequences. The third step, in contrast, refines the model wrt. the actual driving task using evolutionary optimization. We now describe each of the three steps in detail.

Learning a Mixture of Experts Model: A key part of the proposed model is learning of the expert models, $\pi_{\boldsymbol{\theta}}^K$. These models can specialize to certain scenarios and hence increase robustness within those scenarios when compared to a monolithic policy that must learn to handle all modes of the data with a single prediction branch. As the parameter set of the expert network $\boldsymbol{\theta}$ is large, we train it via behavior cloning [1, 33] which solves the perception-action mapping using supervised learning, assuming access to an off-line collection of expert driving demonstrations. Given its sample-efficiency, this technique is the primary workhorse for many state-of-the-art sensorimotor driving models [7, 8, 25], yet existing approaches do not learn data-driven situational policies with a mixture model.

We formulate the following loss function for training our MoE model from demonstrations:

$$\mathcal{L}_{\text{MoE}} = \beta_0 \mathcal{L}_{\text{I}} + \beta_1 \mathcal{L}_{\text{V}} + \beta_2 \mathcal{L}_{\text{R}} \tag{2}$$

where β_i are scalar hyper-parameters which trade-off the three components of this loss function. The imitation loss is defined as the negative log-likelihood of the mixture density network [3, 13]

$$\mathcal{L}_{\text{I}} = -\log \left[\sum_{k=1}^{K} \alpha_{\boldsymbol{\theta}}^k(\mathbf{o}, c) \pi_{\boldsymbol{\theta}}^k(\mathbf{a}|\mathbf{o}, c) \right] \tag{3}$$

where we model each probabilistic expert policy $\pi_{\boldsymbol{\theta}}^k$ as a Gaussian distribution with mean and standard deviation determined by a neural network with parameters $\boldsymbol{\theta}$:

$$\pi_{\boldsymbol{\theta}}^k(\mathbf{a}|\mathbf{o}, c) = \mathcal{N}\left(\mathbf{a} \,\middle|\, \boldsymbol{\mu}_{\boldsymbol{\theta}}^k(\mathbf{o}, c),\, \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{\theta}}^k(\mathbf{o}, c))^2\right) \tag{4}$$

Behavioral cloning provides a sample-efficient way for training an initial driving model by optimizing an imitation loss that is surrogate to the actual driving task. However, the imitation objective only implicitly encodes the task objective [15, 35]. This is a significant issue that can be alleviated through task-driven policy refinement (see step 3 of our learning curriculum) as well as with auxiliary losses [2, 8].

Following Codevilla et al. [8], we incorporate a velocity prediction branch and an additional loss term in addition to the imitation loss for regularizing learning during this stage:

$$\mathcal{L}_{\text{V}} = ||\hat{v}_{\boldsymbol{\theta}} - v||_2^2 \tag{5}$$

We also add a reconstruction branch and loss which is useful for learning general purpose features [13, 44]:

$$\mathcal{L}_{\text{R}} = ||\hat{\mathbf{I}}_{\boldsymbol{\theta}} - \mathbf{I}||_2^2 \tag{6}$$

Here, $\hat{v}_{\boldsymbol{\theta}}$, $\hat{\mathbf{I}}_{\boldsymbol{\theta}}$ are the network predictions and v, $\mathbf{I}$ denote the measured velocity and the observation, respectively.

Learning the Context Embedding: The context embedding in Eq. (1) enables to integrate context information that is complementary to the learned expert policies as it is a shallow network trained independently from the experts using a different objective from the mixture model training. Moreover, due to the multi-step policy optimization process, the context embedding term can provide opportunities

to recover from sub-optimal solutions using the additional context [27, 43].

Due to known bias and generalization challenges on CARLA, e.g., overfitting to certain actions and the 'inertia problem' [8], we learn a general purpose embedding $q_{\boldsymbol{\phi}}(\mathbf{I})$ from image observations alone. As evaluation on CARLA has a diverse range of weathers not seen in training, e.g., from rainy to sunset weathers where large amounts of useful, task-specific visual scene information learned during training becomes unreliable in testing. Therefore, such an embedding provides additional diversity for learning a generalized policy. Following Ha and Schmidhuber [13] we train a shallow VAE [19, 34, 39] with encoder $q_{\boldsymbol{\phi}}$ and decoder $d_{\boldsymbol{\phi}}$ to produce a compact action-agnostic context embedding $\mathbf{z}$. While [13] employs a VAE to encode a highly simplified driving environment, we analyze its utility in more complex settings, i.e., textured and realistic rendering of autonomous driving scenes with CARLA. We minimize the variational lower bound

$$\mathcal{L}_{\text{VAE}} = \beta \, \text{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{I}) \parallel p_0(\mathbf{z})\right) + \|d_{\boldsymbol{\phi}}(\mathbf{z}) - \mathbf{I}\|_2^2 \quad (7)$$

of a β-VAE [17] where $p_0(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$ refers to the standard normal distribution, KL is the Kullback-Leibler divergence, $\mathbf{z}$ is sampled from the posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{I})$ and the hyper-parameter β provides a trade-off between reconstruction loss and the KL-divergence. Note that we have abbreviated the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{I})$ with $q_{\boldsymbol{\phi}}(\mathbf{I})$ in Eq. (1) to avoid clutter in the notation. At inference time, we draw a sample from this distribution and combine it with the current speed and the control command as context embedding, see right part of Eq. (1).

Task-Driven Policy Refinement: In the final step, we optimize the driving policy $\pi_{\boldsymbol{\Theta}}$ with respect to the actual driving task which we define in terms of a reward function. The reward takes into account sequence completion, collision avoidance and traffic infractions. In contrast to the first two steps, this refinement enables the policy to interact with the simulation and collect experience in an on-policy manner, further reducing the remaining co-variate shift of the expert demonstration training set. In particular, this step helps to encourage the learned agent to adhere to traffic rules and safety, an essential component for real-world driving. Unlike current state-of-the-art methods on CARLA [5, 8], optimization wrt. the task enables the agent to go beyond imitation of the driving expert to compose the expert models and the context embedding in a way that generates a more robust and safe driving behavior.

For efficiency, we only update the parameters $\boldsymbol{\Psi}$ and the head of the expert network that predicts the mixture weights $\alpha_{\boldsymbol{\theta}}$. Intuitively, this step combines the pre-trained experts and context embedding with the goal of improving the policy $\pi_{\boldsymbol{\Theta}}$ for the actual driving task. We will use $\boldsymbol{\theta}$ to refer to the subset of the parameters $\boldsymbol{\theta}$ that belong to this part of the network architecture. The remaining parameters in $\pi_{\boldsymbol{\Theta}}$ are kept frozen. Note that unlike previous approaches that have trained reinforcement learning agents on CARLA by fine-tuning the entire perception stack of sensorimotor control policies [25], here we update only the mixture coefficients over *predictions* provided by pre-trained models. This expert-level optimization facilitates a sample-efficient training process (e.g., compared to Dosovitskiy et al. [10] which achieves poor performance even after million of interaction steps) as the predictions by the experts can guide exploration [42]. We experimentally demonstrate that a recombination of experts indeed leads to a more robust final policy.

More formally, our task-driven optimization step maximizes the expected reward when following the policy $\pi_{\boldsymbol{\Theta}}$ sequentially over T time steps

$$\mathcal{J}_{\text{TASK}}(\tilde{\boldsymbol{\theta}}, \boldsymbol{\Psi}) = \mathbb{E}_{\pi_{\boldsymbol{\Theta}}}\left[\sum_{t=0}^{T} r_t\right] \quad (8)$$

Motivated by recent works that reported successful learning of robust policies in a variety of tasks [13, 36], we optimize the objective wrt. $\tilde{\boldsymbol{\theta}}$ and $\boldsymbol{\Psi}$ using an evolution strategy-based algorithm [14].

3.3. Implementation Details

We utilize a ResNet-50 [16] backbone for our mixture model, trained from scratch with Adam [18] using an initial learning rate of 0.0001. We employ a 256×256 image resolution as we found that increasing the input resolution compared to [8] improves performance slightly. We employ several data augmentation techniques based on [7], such as pixel dropout and color perturbations. For validation we follow the procedure from [6].

We implement two architectures for the MoE model in the experiments, referred to as MoE-Branched (experts share the backbone network) and MoE (each expert has a separate backbone network). In both cases, the model architecture extends the CIL [7] and CILRS [8, 25] approaches. The main difference is that we do not employ hard gating based on c for the experts, but replace it with a MoE head. Instead, we encode the high-level command c as a one-hot vector and input it to the network, also introduced in [7]. This architectural modification allows us to analyze the benefits of combining a set of learned policy prediction heads. The other architecture components, e.g., the MLP for speed measurements are kept the same. The MLP maps the measurements to a non-linear embedding which improves performance as shown in [7]. For the policy refinement step, we follow the publicly available implementation and hyperparameter settings of [13] both for the β-VAE and for CMA-ES [14].

4. Experimental Evaluation

Evaluation Procedure: We employ the CARLA 0.8.4 benchmark [10] as it provides diverse weathers, towns, and dynamic obstacles for analyzing situational reasoning. The environment contains two towns, one for training (Town 1) and one for testing (Town 2). In total there are 14 types of weathers, out of which four are used for training the models on Town 1. These weathers are clear noon, wet noon (with after rain puddles), heavy rain noon, and clear sunset (challenging due to illumination conditions). In this paper, we focus on evaluation on Town 2 as it requires the agent to generalize to new conditions. During standard evaluation on Town 2, the agent is required to drive in the four previously seen weathers, as well as two weathers not seen in training time, wet cloudy noon and soft rain sunset. The evaluation performance metrics involve arrival to the goal within an allocated amount of time over 25 routes for each of the weathers. On the original CARLA benchmark, collisions are allowed to occur along with other types of infractions [10] such that the episode may still complete successfully. For evaluation, the best results out of five test runs are reported for four driving conditions: driving straight, short one turn routes, longer navigation routes, and long navigation routes with dynamic obstacles. In the last case, the number of cars and pedestrians on Town 02 is set to 15 and 50, respectively. Overall, the evaluation requires 600 episodes per test run.

We also employ the more recent *NoCrash* [8] evaluation procedure, which involves several modifications to the original benchmark. The driving conditions are categorized into empty roads, regular traffic, and dense settings, where the last condition involves a higher number of cars and pedestrians in Town 2, of 70 and 150, respectively. In addition to these significantly more challenging settings, any type of collision with pedestrians, cars, or static obstacles results in episode termination. Hence, this evaluation procedure provides a better measure for overall driving performance. Both mean and standard deviation obtained using three overall test runs are reported (the experiments are not entirely determinstic due to simulator randomness).

To fully analyze the ability of the models to generalize beyond the training settings to diverse conditions, we also introduce a new benchmark which we refer to as the *AnyWeather* benchmark. We follow the original CARLA 0.8.4 benchmark but increase the types of new weathers to also include drastically different weathers from training conditions, e.g., a sunset with heavy rain conditions. In this evaluation procedure the agent drives on the test town (Town 2) with all the new weather types which are unseen in training (see supplement for visualizations). The *AnyWeather* benchmark incorporates 10 novel weathers, some are particularly challenging in terms of visibility and weather artifacts. Given that generalization capability is crucial for real-world autonomous driving, this benchmark is used in order to highlight the limitations of existing models.

Baselines: The closest baseline to ours is the recent CILRS behavior cloning model [8]. CILRS uses demonstrations as supervision, and so can be compared directly with our mixture model, i.e., for the monolithic case of $K = 1$. We report CILRS navigation performance numbers by re-running the publicly available models provided by [8]. A concurrent work to ours is the recently proposed LBC model [5]. However, the work employs a highly privileged agent, i.e., assuming access to an agent trained with extensive 3D annotations. To ensure meaningful comparison to our approach which does not assume access to such information, LBC can be considered as an upper limit on performance.

Experiments: We demonstrate the benefits of the proposed situational driving framework over four main experiments. First, we motivate the approach by training behavior cloning models while varying the dataset. Second, we perform ablative analysis for the model choices, including the task-driven optimization stage for the MoE policy refinement. Third, we discuss the performance of our method in comparison with several baselines on the CARLA benchmarks. Fourth, we explore the limits of the generalization ability of the situational model in diverse conditions unseen in training.

4.1. Results

Mixture Model Performance: The goal of this initial experiment, shown in Table 2, is to motivate the need for employing a mixture model to learn more flexible sensorimotor driving models. Specifically, we demonstrate how training a monolithic behavior cloning policy as in the CILRS [8] baseline can lead to poor decision-making and generalization performance across navigation tasks. This issue can be analyzed by varying the training data to introduce an additional perception-action modality, and consequently analyzing model performance within each data modality.

As shown in Table 2, we train three different models. We focus on the MoE training without the refinement step as it leads to the most significant improvement in driving performance. First, a monolithic policy is trained over scenes containing no dynamic obstacles, referred to as Nav. Static. Remarkably, the model learns to solve the static scenes navigation task, even when driving in new town and new weather conditions, better than the baseline models. However, the model is unable to safely drive around dynamic obstacles as these were not observed in training. Nonetheless, this experiment shows the strong benefit of learning situation-specialized policy models.

The second model is a similar monolithic behavior cloning policy, with one difference. The model is now trained with a dataset that also contains dynamic obstacles,

referred to as Nav. Dynamic (K=1). The presence of dynamic obstacles requires the agent to learn to slow down and brake appropriately. As shown in Table 2, this model can better handle navigation in such settings, but this improvement comes with a trade-off in generalization performance over settings and weathers, i.e., dynamic vs. static scenes. For example, performance for the static navigation task are reduced from 96% to 78%.

Finally, we train an MoE model with three components with the same dataset, referred to as Nav. Dynamic (K=3). The model achieves a 98% episode success rate on navigation in static scenarios and 92% in dynamic scenarios. Learning a mixture model effectively addresses the aforementioned issue, while achieving state-of-the-art performance without utilizing on-policy data or privileged information (e.g., [5]). Because there are shared elements of driving behavior over both the dynamic and static scenes analyzed (e.g., during lane following and turning), the situational reasoning improves performance both within each driving scenario as well as across scenarios.

Ablation: Table 3 shows the contribution of different training steps in the situational model on overall navigation success. Our baseline monolithic model already improves over the performance of CILRS [8] for the Nav. Dynamic task. We then train the two variants of the MoE architecture, with most gains seen due to incorporating a $K = 3$ component model. Adding mixture components up to $K = 5$ leads to a minor improvement, with examples of learned experts visualized in Fig. 3. The branched architecture, which is more computationally efficient due to sharing of the backbone network, shows an absolute improvement of 14% over the monolithic baseline. We can see how the experts specialize into different components of the driving task with respect to throttle and brake control. Further gains in performance are observed by training experts that do not share the backbone network due to the increase in diversity between the experts (also discussed in [31]).

We also analyze the limitations of learning the MoE policy with behavior cloning in Table 3. Specifically, we can see how refinement of the MoE policy through interaction with the environment can further improve driving performance. Although we only update the final layer for predicting the mixing coefficients, this step alone leads to a more robust policy and a 4% improvement. The refinement step mostly leads to improved driving performance in dynamic scenarios, as shown in Table 3.

Comparison with State-of-the-Art: We now compare our full model performance with several previously proposed approaches on the original CARLA benchmark in Table 4 and the *NoCrash* benchmark in Table 5. Results are shown both without and with the task-driven refinement stage, referred to as **LSD** and **LSD+**, respectively. Our proposed

Table 2: **Monolithic vs. Mixture.** We analyze the driving performance for new town (Town 2) & new weather conditions when introducing dynamic obstacles into the training town (Town 1).

	Training Data and Model		
Task	Nav. Static (K=1)	Nav. Dynamic (K=1)	Nav. Dynamic (K=3)
Straight	99	64	**100**
One Turn	98	74	**100**
Navigation	96	78	**98**
Nav. Dynamic	40	78	**92**

Table 3: **Ablative Analysis.** Performance shown for the new town and dynamic obstacles (Nav. Dynamic) settings.

Model	**Success Rate** (%)
Monolithic (K=1)	75
MoE-Branched (K=3)	89
MoE-Branched (K=5)	90
MoE-Branched (K=8)	87
MoE (K=3)	94
MoE (K=5)	93
MoE (K=8)	93
MoE+Refinement (K=3)	**98**

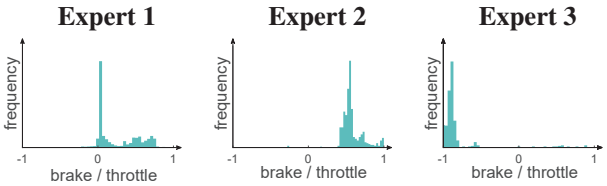

Figure 3: **Learned Experts' Statistics.** Acceleration behavior distribution of three different experts during testing.

model significantly improves over state-of-the-art driving performance on both test conditions of new town and new town with two new weathers. As previously mentioned, we can see how state-of-the-art models are unable to navigate empty road conditions well in some cases, e.g., CILRS achieves a 65% success rate Table 5. In contrast, our model is able to learn a policy that can handle such differing scenarios, achieving expert-level behavior. Moreover, the MoE approach also improves performance within driving tasks by combining the situation-specific policies. Our multi-stage learned policy also improves over CIRL [25], another approach incorporating reinforcement learning to optimize for the driving task. We find that enabling the model to learn through interaction experience and collisions facilitates significantly better behavior in dense traffic conditions. However, on *NoCrash*, even the provided expert is unable to solve the driving due to a variety of reasons unrelated to

Table 4: Comparison of success rates (%) with the state-of-the-art on the original CARLA 0.8.4 benchmark. A '*' indicates our independently performed evaluation using the publicly available model.

	New Town						New Town & Weather					
Task	CIRL [25]	CILRS [8]	CILRS*	LSD	LSD+	LBC [5]	CIRL [25]	CILRS [8]	CILRS*	LSD	LSD+	LBC [5]
Straight	**100**	96	96	**100**	**100**	100	98	96	78	**100**	**100**	100
One Turn	71	84	86	**99**	**99**	100	80	92	96	**100**	**100**	100
Navigation	53	69	67	**99**	**99**	100	68	92	96	98	**100**	100
Nav. Dynamic	41	66	64	94	**98**	99	62	90	94	92	**98**	100

Table 5: Comparison of success rates (%) with the state-of-the-art on the *NoCrash* CARLA 0.8.4 benchmark. Mean and standard deviation are shown over three runs.

	New Town					New Town & Weather				
Task	CILRS [8]	CILRS*	LSD	LSD+	Expert	CILRS [8]	CILRS*	LSD	LSD+	Expert
Empty	66 ± 2	65 ± 2	93 ± 2	$\mathbf{94 \pm 1}$	96 ± 0	90 ± 2	71 ± 2	$\mathbf{96 \pm 1}$	95 ± 1	96 ± 2
Regular	49 ± 5	46 ± 2	66 ± 2	$\mathbf{68 \pm 2}$	91 ± 1	56 ± 2	59 ± 4	61 ± 1	$\mathbf{65 \pm 4}$	92 ± 1
Dense	23 ± 1	20 ± 1	27 ± 2	$\mathbf{30 \pm 4}$	41 ± 2	24 ± 8	31 ± 3	29 ± 4	$\mathbf{32 \pm 3}$	43 ± 2

the agent. For instance, in dense settings, pedestrians and other cars may crash into the ego-vehicle or block an intersection indefinitely at no fault of the ego-vehicle, resulting in unsuccessful episode completion.

The *AnyWeather* Benchmark: As a final experiment we seek to quantify the ability of our model to operate under drastically diverse visual conditions, which is essential in real-world driving. While learning specialized policies can provide some flexibility under different settings, the analysis on 10 unseen weathers further highlights the benefits and limitations of our approach. Table 6 shows a summary of the results in this challenging settings over a total 1000 episodes, 250 for each condition. Due to this large number of episodes, even small improvements in success rates are significant. The results should be directly compared to the results in Table 4. Here, even simpler tasks such as driving straight in static scenes is no longer solved due to the harsh weather conditions. Surprisingly, several new weathers are so difficult that they result in zero success rate by state-of-the-art approaches (both CILRS and LSD), motivating future study of this challenging benchmark.

Table 6: **Generalization to Harsh Environments on the *AnyWeather* Benchmark.** Success rates (%) for new town (Town 2) and all 10 weathers unseen in training on the CARLA 0.8.4 benchmark.

	New Town & Weather		
Task	CILRS*	LSD	LSD+
Straight	83.2	85.2	**85.6**
One Turn	78.4	80.4	**81.6**
Navigation	76.4	78.8	**79.6**
Nav. Dynamic	75.6	77.2	**78.4**

5. Conclusion

We presented a situational policy model for driving in diverse scenarios. Based on our experiments, employing a mixture model when learning sensorimotor driving can lead to significant improvements in modeling capacity across different driving tasks. Moreover, directly optimizing for the driving task can provide additional performance gains, achieving state-of-the-art performance on the CARLA, *NoCrash*, and *AnyWeather* benchmarks. Although our approach does not require access to image-level annotations, the situational model can also be learned over a perception-module, providing a stronger visual prior and improving generalization capabilities further. Moreover, the situational formulation provides some interpretability, as the situation-specific predictions can be inspected at test time. Another future direction would be to evaluate the ability of the model to generalize to new traffic scenarios, i.e., through the composition of the expert policies. Given that our work takes a step towards learning robust, generalized driving policies, an important next step would be to further analyze the MoE model on challenging generalization settings, e.g., real-world datasets and Sim2Real [28].

Acknowledgements: This work was supported by the BMBF through the Tübingen AI Center (FKZ: 01IS18039B). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Kashyap Chitta and the Humboldt Foundation for supporting Eshed Ohn-Bar.

References

[1] M. Bain and C. Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, 1996.

[2] M. Bansal, A. Krizhevsky, and A. Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In *RSS*, 2019.

[3] C. M. Bishop. Mixture density networks. 1994.

[4] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015.

[5] D. Chen, B. Zhou, and V. Koltun. Learning by cheating. In *CoRL*, 2019.

[6] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy. On offline evaluation of vision-based driving models. In *ECCV*, 2018.

[7] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*.

[8] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *ICCV*, 2019.

[9] P. Dayan and G. E. Hinton. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*, 1993.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017.

[11] M. R. Endsley, D. J. Garland, et al. Theoretical underpinnings of situation awareness: A critical review. *Situation Awareness Analysis and Measurement*, 1, 2000.

[12] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017.

[13] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

[14] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[15] H. He, J. Eisner, and H. Daume. Imitation learning by coaching. In *Advances in Neural Information Processing Systems*, 2012.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] D. P. Kingma and M. Welling. Auto-encoding variational bayes. 2014.

[20] T. Kipf, Y. Li, H. Dai, V. Zambaldi, A. Sanchez-Gonzalez, E. Grefenstette, P. Kohli, and P. Battaglia. CompILE: Compositional imitation learning and execution. In *ICML*, 2019.

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

[22] J. Koutník, G. Cuccu, J. Schmidhuber, and F. Gomez. Evolving large-scale neural networks for vision-based reinforcement learning. In *Genetic and Evolutionary Computation*, 2013.

[23] D. Kuan, G. Phipps, A.-C. Hsueh, et al. Autonomous robotic vehicle road following. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10(5):648–658, 1988.

[24] G. Li, M. Mueller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem. Oil: Observational imitation learning. *RSS*, 2019.

[25] X. Liang, T. Wang, L. Yang, and E. Xing. CIRL: Controllable imitative reinforcement learning for vision-based self-driving. In *ECCV*, 2018.

[26] C. C. Macadam. Understanding and modeling the human driver. *Vehicle System Dynamics*, 40(1-3), 2003.

[27] D. Q. Mayne, M. M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[28] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun. Driving policy transfer via modularity and abstraction. *CoRL*, 2018.

[29] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in Neural Information Processing Systems*, 2006.

[30] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[31] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, 2016.

[32] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. MCP: Learning composable hierarchical control with multiplicative compositional policies. In *Advances in Neural Information Processing Systems*, 2019.

[33] D. A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, 1989.

[34] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[35] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

[36] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv*, 1703.03864, 2017.

[37] A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. In *CoRL*, 2018.

[38] A. Sax, B. Emi, A. R. Zamir, L. Guibas, S. Savarese, and J. Malik. Mid-level visual representations improve generalization and sample efficiency for learning active tasks. In *CoRL*, 2019.

[39] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019.

[40] S. Shalev-Shwartz and A. Shashua. On the sample complexity of end-to-end training vs. semantic abstraction training. *arXiv*, 1807.01622, 2016.

[41] A. Sharma, M. Sharma, N. Rhinehart, and K. M. Kitani. Directed-info GAIL: Learning hierarchical policies from unsegmented demonstrations using directed information. *ICLR*, 2019.

[42] W. B. Shen, D. Xu, Y. Zhu, L. J. Guibas, L. Fei-Fei, and S. Savarese. Situational fusion of visual representation for visual navigation. *ICCV*, 2019.

[43] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling. Residual policy learning. In *ICRA*, 2019.

[44] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[45] R. S. Sutton, D. Precup, and S. P. Singh. Intra-option learning about temporally abstract actions. In *ICML*, 1998.

[46] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, 2016.

[47] J. R. Uijlings and V. Ferrari. Situational object boundary detection. In *CVPR*, 2015.

[48] D. Wang, C. Devin, Q.-Z. Cai, P. Krähenbühl, and T. Darrell. Monocular plan view networks for autonomous driving. *IROS*, 2019.

[49] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *CVPR*, 2017.

[50] A. M. Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, 10(1):1–7, 2019.

[51] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[52] B. Zhou, P. Krähenbühl, and V. Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019.

[53] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017.