World model learning from demonstrations with active inference: application to driving behavior*

Ran Wei[1], Alfredo Garcia[1], Anthony McDonald[1,2], Gustav Markkula[3], Johan Engstrom[4], Isaac Supeene[4], and Matthew O'Kelly[4]

[1] Texas A&M University, USA
[2] University of Wisconsin, USA
[3] University of Leeds, UK
[4] Waymo LLC, USA

Abstract. Active inference proposes a unifying principle for perception and action as jointly minimizing the free energy of an agent's internal world model. In the active inference literature, world models are typically pre-specified or learned through interacting with an environment. This paper explores the possibility of learning world models of active inference agents from recorded demonstrations, with an application to human driving behavior modeling. The results show that the presented method can create models that generate human-like driving behavior but the approach is sensitive to input features.

Keywords: Active inference · inverse reinforcement learning · driving behavior modeling.

1 Introduction

Active inference proposes a unifying principle for perception and action as jointly minimizing the free energy of an agent's internal generative model [6]. It has been strongly influential in contemporary neuroscience and cognitive science. More recently, active inference has been proposed as a framework for modeling driving behavior, both at the conceptual [5, 10] and computational levels [31]. The framework is attractive for computational driver behavior modeling as it enables the learning of complex behaviors from large amounts of driving data while at the same time being grounded in a fundamental theory of cognition and behavior which guides model design and enables increased interpretability of machine-learned models. However, most existing active inference models in the cognitive neuroscience literature address relatively simple toy problems. Thus, the scaling of active inference by means of modern machine learning techniques

* Support for this research was provided in part by grants from the U.S. Department of Transportation, University Transportation Centers Program to the Safety through Disruption University Transportation Center (451453-19C36) and the UK Engineering and Physical Sciences Research Council (EP/S005056/1). The role of the Waymo employees in the project is solely consulting including making suggestions and helping set the technical direction.

is currently an active area of research [29]. The novel contribution of this paper is to explore the application of active inference models in the context of learning human driving behavior from recorded data (i.e., Learning from Demonstration; LfD).

LfD provides an efficient alternative to the current manual specification or trial-and-error learning approaches to active inference model design. Assuming the demonstrating agent is an active inference agent, we can instead estimate the agent's generative model, consisting of a world model and a preference model, from demonstrated behavior. This approach is similar to inverse reinforcement learning (IRL) [20, 33] with an important difference. Instead of using a single reward function, active inference explains the demonstrator with a world model-preference pair, which makes active inference more transparent about the agent's decision process than traditional IRL methods because we can introspect the learned world model. This allows us to understand variations in human behavior as "optimal inference in suboptimal models" [26, 31].

The closest approaches to the work presented here are [1,11,15,23]. We build on these works by jointly estimating agent world model and preference model from demonstration. However, our work differs from these approaches in that it does not assume the environment is fully observable as in [23], it makes no assumptions about the agent's world model's alignment with the environment in light of the active inference formulation [11,15], and it focuses on a large continuous environment rather than a small discrete environment [1]. We demonstrate our method in continuous car following scenarios recorded on highways [32]. The learned driving policy jointly models its own states, road geometry, and other vehicles (i.e., agents) using discrete abstract states and implements continuous vehicle control. We show that this approach can mimic human driving behavior in simple scenarios but that it may learn an incorrect model of the world, known as "causal confusion" in LfD [4], and occasionally deviate from the lane. We further show that this deviation can be corrected by revising the observation set based on grounded theory of driver steering [25], thus illustrating the how inductive biases and domain knowledge can be injected into LfD approaches.

2 Active Inference Model of Highway Driving

In this section, we propose a mixed discrete-continuous active inference model of driving behavior and present the update rules for driver perception and control by minimizing expected free energy.

2.1 World model

We model the driver's perceptual process using a discrete-time controlled hidden Markov process with discrete hidden states $s \in \mathcal{S}$, discrete actions $a \in \mathcal{A}$, and continuous observations $o \in \mathcal{O}$. The hidden states are the driver's *internal* representation of the driving environment which is used to guide action selection (e.g steering or braking). The discrete actions represent driving motor primitives (i.e., prototype actions as described in [16]). The continuous observations

are a vector of signals known to influence driving control behavior (e.g., visual looming of the lead vehicle [18]). The state evolves according to a Markov chain with transition probabilities $P(s_{t+1}|s_t, a_t)$. The driver cannot directly observe the state but a high dimensional continuous signal o_t with distribution $P(o_t|s_t)$. Importantly, the definition of states and the corresponding transition and observation probabilities are free to deviate from the actual environment as long as they explain the demonstrated behavior.

2.2 A POMDP formulation of active inference

Given the world model, the agent's perception-action loop at every decision epoch consists of inferring a belief distribution on the current hidden state and selecting an action controlling the evolution of the hidden state. Active inference posits the minimization of *free energy* as a unifying principle for describing the perception-action loop.

Let $h_t = \{o_t, ..., o_0, a_{t-1}, ..., a_0\} \in H_t$ denote the observable history of the dynamic decision process including all past and present revealed observations and all implemented actions up to time $t > 0$, where $H_t \triangleq \mathcal{O}^{t+1} \times \mathcal{A}^t$.

According to the free energy minimization principle, the agent's belief distribution at time $t > 0$ which we denote by $b_t(s_t)$ must correspond to the Bayes updated belief distribution on the state s_t, i.e. the conditional probability distribution of s_t given history h_t, i.e. $b_t(s_t) = P(s_t|h_t)$. The active inference model of the perception-action loop assumes the agent has preferences over hidden states s_{t+1} which are represented by a probability distribution $\tilde{P}(s_{t+1})$. The expected free energy associated with the choice of action a_t and current belief distribution b_t at time $t > 0$ can be written as [3]:

$$EFE(b_t, a_t) = \mathbb{E}\big[D_{KL}\big(b_{t+1}||\tilde{P}\big)\big] + \mathbb{E}[\mathcal{H}(o_{t+1})] \tag{1}$$

where the first expectation is taken with respect to

$$\begin{aligned} P(o_{t+1}|b_t, a_t) &:= \sum_{s_{t+1}} P(o_{t+1}|s_{t+1}) P(s_{t+1}|b_t, a_t) \\ &= \sum_{s_{t+1}} P(o_{t+1}|s_{t+1}) \sum_{s_t} P(s_{t+1}|s_t, a_t) b(s_t) \end{aligned} \tag{2}$$

and $D_{KL}\big(b_{t+1}||\tilde{P}\big)$ is the Kullback-Leibler divergence between the random belief distribution $b_{t+1}(\cdot) = P(\cdot|h_t \cup \{o_{t+1}, a_t\})$ and $\tilde{P}(\cdot)$. $\mathbb{E}[\mathcal{H}(o_{t+1})]$ is the entropy of the observables expected under the predictive distribution $P(s_{t+1}|b_t, a_t)$ defined in (2). The first term in (1) is a measure of the extent to which the belief distribution b_{t+1} (resulting from implementing action a_t and recording observation o_{t+1}) differs from the preferred one $\tilde{P}$. Let $\pi \in \Pi$ denote a randomized action selection policy conditioned on the history of the process, i.e. $\pi(a|h_t) \in [0, 1], a \in \mathcal{A}$ and $\sum_{a \in \mathcal{A}} \pi(a|h_t) = 1$ for all $h_t \in H_t$. An information processing cost is modeled as the Kullback-Leibler divergence between policy π and a default a priori control

policy π_0 which is oblivious to new information [21, 28] i.e.:

$$D_{KL}(\pi(\cdot|h_t)||\pi_0) := \sum_{a\in\mathcal{A}} \pi(a|h_t) \log \frac{\pi(a|h_t)}{\pi_0(a)}$$

With a uniform default distribution, $D_{KL}(\pi(\cdot|h_t)||\pi_0) = \mathbb{E}_{\pi(a|h_t)} \log \pi(a|h_t) - \log|\mathcal{A}|$. For a finite planning horizon T, the *active inference* controller is the solution to the problem:

$$\mathcal{G}^*_\tau(h_\tau) \triangleq \min_{\pi\in\Pi} \mathbb{E}\big[\sum_{t\geq\tau}^{T} (EFE(b_t, a_t) + \log \pi(a_t|h_t))\big] \tag{3}$$

The combination of additive structure and Markovian dynamics allows for a recursive characterization of the optimal policy as follows:

$$\begin{aligned} \mathcal{G}^*_t(h_t) = \min_{\pi\in\Pi} \Bigg\{ \sum_{a_t\in\mathcal{A}} \pi(a_t|h_t)\big[\\ EFE(b_t, a_t) + \log \pi(a_t|h_t) + \int_{\mathcal{O}} P(o_{t+1}|h_t, a_t)\mathcal{G}^*_{t+1}(h_{t+1})do_{t+1}\big] \Bigg\} \end{aligned} \tag{4}$$

where $h_{t+1} = h_t \cup \{o_{t+1}, a_t\}$. Note that with no loss of generality the recursive equation can be expressed in terms of belief states b_t as opposed to the history h_t. The following is a standard result characterizing the optimal solution to (4) [7].

Proposition 1 *Let $\mathcal{G}_t(b_t, a_t)$ be defined as:*

$$\mathcal{G}^*_t(b_t, a_t) := EFE(b_t, a_t) + \log \pi(a_t|b_t) + \int_{\mathcal{O}} P(o_{t+1}|b_t, a_t)\mathcal{G}^*_{t+1}(b_{t+1})do_{t+1}$$

The optimal policy is of the form:

$$\pi(a|b_t) = \frac{e^{-\mathcal{G}^*_t(b_t,a)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{-\mathcal{G}^*_t(b_t,\tilde{a})}} \tag{5}$$

2.3 Estimation of POMDP model

Given the model for the active inference controller described above, in this section, we describe the problem of estimating such a model given recorded sequences of actions and observables. This is akin to *inverse* learning a POMDP model (see section 4.7 in [22]).

In what follows we consider a parametrization of observation probabilities $P_{\theta_1}(o_{t+1}|s_{t+1})$ and state-dynamics $P_{\theta_1}(s_{t+1}|s_t, a_t)$ with $\theta_1 \in \mathbb{R}^{p_1}$ where $p_1 > 0$. Given data in the form of finite histories $h_{T,i} = \{(o_{t,i}, a_{t,i})\}_{t=0}^{T}$ for $i \in \{1, \ldots, N\}$, a sequence of belief trajectories $\{b_{t,\theta_1,i}\}_{t=0}^{T}$ can be recursively computed for a fixed value of θ_1.

Assuming preferences over hidden states are parametrized $\tilde{P}_{\theta_2}(s_{t+1})$ with $\theta_2 \in \mathbb{R}^{p_2}$ with $p_2 > 0$, the log-likelihood of observed actions can be written as:

$$\log \ell(\theta) = \sum_{i=1}^{N} \sum_{t=0}^{T-1} \log \pi_\theta(a_{t,i}|b_{t,\theta_1,i}) \tag{6}$$

where $\pi_\theta(\cdot|b_{\theta_1,t})$ is the optimal policy in (5) and $\theta := (\theta_1, \theta_2)$.

(6) can be optimized using a nested-loop algorithm alternating between **(i)** a parameter update step at iteration $k > 0$ in which we set θ^{k+1} as the solution to:

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \log \pi_\theta(a_{t,i}|b_{t,\theta_1^k,i}) \quad \text{s.t.} \quad \pi_\theta(a_t|b_t) = \frac{e^{-\mathcal{G}^*_{t,\theta^k}(b_t,a_t)}}{\sum_{\tilde{a}_t \in \mathcal{A}} e^{-\mathcal{G}^*_{t,\theta^k}(b_t,\tilde{a}_t)}}$$

where $\mathcal{G}^*_{t,\theta^k}$ denotes the current free energy function and **(ii)** solving for the free energy function $\{\mathcal{G}^*_{t,\theta^{k+1}}\}_t$ given the new parameter values.

3 Implementation

In this section, we first describe the signals assumed to be observed by the drivers during a car-following scenario and defer a detailed description of the dataset to appendix A.1. We then describe the model fitting process with an augmentation of the model to continuous braking and steering control. Finally, we describe the procedure for model comparison.

3.1 Driver observations

We leveraged prior works on driver behavior theory [17, 18, 25] to define the observation vector o used in the car-following task. Markkula et al. [17] proposed visual looming denoted by τ^{-1} as a central observation signal in human longitudinal vehicle control, which is defined as the derivative of the optical angle of the lead vehicle subtended on the driver's retina divided by the angle itself: $\tau^{-1} = \dot{\theta}/\theta$. Salvucci & Gray [25] proposed a two-point model of human lateral vehicle control where the human driver controls the vehicle by representing road curvature with a near-point, assumed at a fixed distance in front of the vehicle, and a far-point, assumed to be the lead vehicle in the car-following context, and steers to minimize the deviation from a combination of the near and far-points. Using these insights, we designed an observation vector consisting of three sensory modalities:

1. The state of the ego vehicle in ego-centric coordinate
2. Relationships with the lead vehicle in ego-centric coordinates
3. Road geometry

We featurized the ego state with the longitudinal and lateral velocity and relationship to the lead vehicle with relative distance and speed with longitudinal

and lateral components, and looming. To encode the road geometry in the two-point model, we used the lane center 30 m ahead of the current position as the near-point and the lead vehicle as the far-point and used as features the heading error from the near and far-points and lane-center distance to the current road position.

3.2 Model fitting

We parameterized the hidden state transition probabilities $P(s_{t+1}|s_t, a_t)$ and preference distribution $\tilde{P}(s_t)$ with categorical distributions and observation probabilities $P(o_t|s_t)$ with multivariate Gaussian distributions. For a fixed belief vector b_t, the expected KL divergence and entropy in (1) can be computed in closed-form. We used the QMDP method [14] to approximate the cumulative expected free energy assuming the states will become fully observable in the next time step: $\mathcal{G}^*(b_t, a_t) \approx \sum_{s_t} b(s_t)\mathcal{G}^*(s_t, a_t)$. This allows us to train the model in automatic differentiation frameworks (e.g., Pytorch) using Value-Iteration-Networks style implementations [9, 27].

In order to fit the discrete action model from Section 2 to continuous longitudinal and lateral controls, we extended the model with a continuous control module. Let u denote a multidimensional continuous control vector (longitudinal and lateral accelerations in the current setting), we modeled the mapping from a discrete action a to u using $P(u|a)$ parameterized as a multivariate Gaussian with its parameters added to vector θ_1. $P(u|a)$ thus automatically extracts primitive actions, such as different magnitudes of acceleration and deceleration [16], from data by adaptively discretizing the action space. We assume at a given time step t, the agent also performs a Bayesian belief update about the previous action realized with prior given by the policy $\pi(a_t|b_t)$ and the posterior $P(a_t|u_t) \propto P(u_t|a_t)\pi(a_t|b_t)$. The action log likelihood objective in (6) is modified as:

$$\log \ell(\theta) = \sum_{i=1}^{N} \sum_{t=0}^{T-1} \log \sum_{a_{t,i}} P_{\theta_1}(u_{t,i}|a_{t,i})\pi_\theta(a_{t,i}|b_{t,\theta_1,i}) \tag{7}$$

3.3 Model comparison

We measured the quality of the trained agents by using a combination of offline and online testing metrics on the held-out set. For offline metrics, we used mean absolute error (MAE). For online metrics, we first ran the trained agents in a simulator that replayed the recorded trajectories of the lead vehicles and then recorded the final displacement and average lane deviation for each trajectory tested. The final displacement is defined as the distance between the final position reached by the trained agents and the final position in the dataset. The average lane deviation is the agents' distance to the tangent point on the lane center line averaged over all time steps in the trajectory.

We varied three aspects of the agents to compare with the canonical agent described previously. First, we examined the importance of the chosen features

by replacing the near-point heading error and distance to lane center with distances to the left and right boundaries at the current road position, a feature set commonly used by driving agents for simulated testing [2, 13]. We label the agents trained with the original two-point observation as "TP". Next, we examined the importance of grounding the world model in actual observations by adding an observation regularizer to the training objective with a coefficient of 0.01:

$$\mathcal{L}_{obs} = \sum_{t=1}^{T} \log P(o_t|h_t) \tag{8}$$

This encourages the agent to have a more accurate belief about the world with higher observation likelihood under the agent's posterior beliefs. We label agents trained with this penalty "Obs". Finally, we examined the impact of agent planning objectives on the learned world model and behavior. We replaced EFE with an alternative objective called expected cross entropy (ECE):

$$ECE(b_t, a_t) = \mathbb{E}[\log \tilde{P}(o_{t+1})] \tag{9}$$

which is the expected marginal likelihood of the agent preference model.

We used 30 states and 60 actions for all agents as they were sufficient to produce reasonable behavior. As a baseline, we trained a behavior cloning (BC) agent consisting of a recurrent and a feed-forward neural network to emulate the belief update and control modules of the active inference agent. We provide more details of the BC agent in appendix A.2.

4 Results and Discussions

Fig. 1 shows the offline (left panel) and online (middle and right panels) testing metrics for each agent tested using the same set of 15 scenarios sampled from the held-out dataset, with the canonical agent labeled as "TP+EFE". The MAE of all active inference agents were between 0.11 and 0.14 m/s^2. The BC agent outperformed all agents with a MAE of 0.08, however the BC+TP agent had a higher MAE value of 0.135. This is likely due to the sensitivity to input features during training, despite better function approximation capability of neural networks. The final displacements were on average 13m, the average lane deviation was 1.37 m, and no collision with the lead vehicle was observed. These metrics show that the agents can generate reasonable behavior by staying in the lane and following the lead vehicle (see a few sample trajectories generated in Fig. 3a).

Comparing across different agents, Fig. 1 shows that adding an observation penalty increased offline MAE, however, it did not noticeably affect the agents' online performance. This might be related to the objective mismatch problem in model-based reinforcement learning where a model better fitted to the observations may not enhance control capabilities [12]. The middle and right panels show that some of the agents produced final displacements and lane deviation as

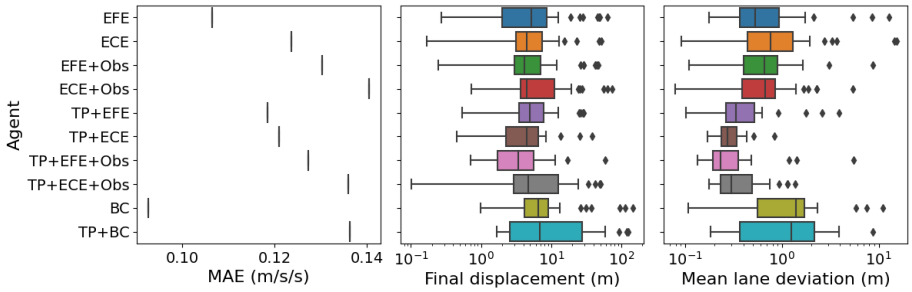


Fig. 1: Box plots of offline (column 1) and online (columns 2 & 3) performance metrics of the compared agents. Offline metrics are calculated on the entire held-out set. Each box plot in the online metrics shows the distribution of agent performance in 15 random held-out scenarios tested with 3 different random seeds.

large as 100 m and 15 m, respectively, as a result of deviating from the lane and failing to make corrections (see Fig. 3b). Interestingly, active inference agents using the two-point observations model generated noticeably less lane deviation than other agents (see Fig. 1 right with x axis in log-scale) despite similar performance in terms of offline metrics. This observation highlights the importance of incorporating generalizable features into agent world model.

Fig. 2 shows a subset of the parameters of the learned world models. All panels ordered the states by desirability so that states with lower EFE had smaller indices. The left panel plots the variance of the observation distribution for the relative distance feature against the states. The orange and blue lines represent the ECE and EFE objectives, respectively. This panel shows a clear increasing trend in the observation variance with the decrease of state desirability. The middle and right panels show the transition matrices controlled by the learned policy: $P^{\pi}(s'|s) = \sum_{a \in \mathcal{A}} P(s'|s,a)\pi(a|b = \delta(s))$. Whereas the transition probabilities of the ECE agent spread more uniformly across the state space, the transition matrix of the EFE agent has a block-diagonal structure. As a result, it is difficult to traverse to the desirable states in the upper diagonal (states 0-24) from the undesirable states (states 24 - 30) in the lower diagonal. We have empirically observed that when the EFE agent deviates from the lane, its EFE values also increase significantly without it taking any corrective actions. This shows that the increasing variance played a more important role in determining the desirability of a state than the KL divergence from the preferred states.

The observation made in Fig. 2 is similar to the "causal confusion" problem in LfD [4]. In [4], the authors found that the learning agent may falsely attribute the cause of an action to the previous actions in the demonstration rather than the external signals and its own goals. Our agent exhibited a different type of "causal confusion" similar to the model exploitation phenomena in reinforcement learning [8], where the cause of an action is attributed to a model with

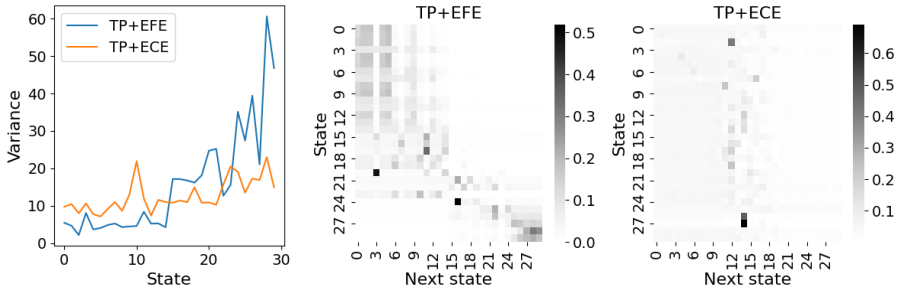


Fig. 2: Parameters of the learned world models. States are sorted by desirability (i.e., low expected free energy). **Left:** Observation variance vs. state. **Middle & right:** Heat map of controlled transition matrix. Darker color corresponds to higher transition probability.

incorrect counterfactual state and observation predictions. The consequence is that the agent does not have the ability to make corrections when entering these states. However, learning the correct counterfactual states from demonstration is difficult because these states are rarely contained in the demonstration as the demonstrating agents are usually experts who rarely visit undesirable states. Prior works addressed this by interacting with an environment [30] and receiving real-time expert feedback [24]. We have instead partially alleviated this by designing domain specific features (i.e., the two-point observation model) to reduce the probability of the agent deviating from desired states. However, given active inference strongly relies on counterfactual simulation of the world model in the planning step, future work should focus on discovering the correct counterfactual states from human demonstrations using approaches at the model level rather than at the feature level, e.g., by learning causal world models via environment interactions or constraining the model class [4].

References

1. Baker, C., Saxe, R., Tenenbaum, J.: Bayesian theory of mind: Modeling joint belief-desire attribution. In: Proceedings of the annual meeting of the cognitive science society. vol. 33 (2011) 2
2. Bhattacharyya, R., Wulfe, B., Phillips, D., Kuefler, A., Morton, J., Senanayake, R., Kochenderfer, M.: Modeling human driving behavior through generative adversarial imitation learning. arXiv preprint arXiv:2006.06412 (2020) 7
3. Da Costa, L., Parr, T., Sajid, N., Veselic, S., Neacsu, V., Friston, K.: Active inference on discrete state-spaces: a synthesis. Journal of Mathematical Psychology **99**, 102447 (2020) 3
4. De Haan, P., Jayaraman, D., Levine, S.: Causal confusion in imitation learning. Advances in Neural Information Processing Systems **32** (2019) 2, 8, 9
5. Engström, J., Bärgman, J., Nilsson, D., Seppelt, B., Markkula, G., Piccinini, G.B., Victor, T.: Great expectations: a predictive processing account of automobile driving. Theoretical issues in ergonomics science **19**(2), 156–194 (2018) 1

6. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G.: Active inference: a process theory. Neural computation **29**(1), 1–49 (2017) 1
7. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. pp. 1861–1870. PMLR (2018) 4
8. Janner, M., Fu, J., Zhang, M., Levine, S.: When to trust your model: Model-based policy optimization. Advances in Neural Information Processing Systems **32** (2019) 8
9. Karkus, P., Hsu, D., Lee, W.S.: Qmdp-net: Deep learning for planning under partial observability. Advances in neural information processing systems **30** (2017) 6
10. Kujala, T., Lappi, O.: Inattention and uncertainty in the predictive brain. Frontiers in Neuroergonomics **2** (2021) 1
11. Kwon, M., Daptardar, S., Schrater, P.R., Pitkow, X.: Inverse rational control with partially observable continuous nonlinear dynamics. Advances in neural information processing systems **33**, 7898–7909 (2020) 2
12. Lambert, N., Amos, B., Yadan, O., Calandra, R.: Objective mismatch in model-based reinforcement learning. arXiv preprint arXiv:2002.04523 (2020) 7
13. Leurent, E.: An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env` (2018) 7
14. Littman, M.L., Cassandra, A.R., Kaelbling, L.P.: Learning policies for partially observable environments: Scaling up. In: Machine Learning Proceedings 1995, pp. 362–370. Elsevier (1995) 6
15. Makino, T., Takeuchi, J.: Apprenticeship learning for model parameters of partially observable environments. arXiv preprint arXiv:1206.6484 (2012) 2
16. Markkula, G., Boer, E., Romano, R., Merat, N.: Sustained sensorimotor control as intermittent decisions about prediction errors: Computational framework and application to ground vehicle steering. Biological cybernetics **112**(3), 181–207 (2018) 2, 6
17. Markkula, G., Engström, J., Lodin, J., Bärgman, J., Victor, T.: A farewell to brake reaction times? kinematics-dependent brake response in naturalistic rear-end emergencies. Accident Analysis & Prevention **95**, 209–226 (2016) 5
18. McDonald, A.D., Alambeigi, H., Engström, J., Markkula, G., Vogelpohl, T., Dunne, J., Yuma, N.: Toward computational simulations of behavior during automated driving takeovers: a review of the empirical and modeling literatures. Human factors **61**(4), 642–688 (2019) 3, 5
19. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018) 11
20. Ng, A.Y., Russell, S.J., et al.: Algorithms for inverse reinforcement learning. In: Icml. vol. 1, p. 2 (2000) 2
21. Ortega, P.A., Braun, D.A.: Thermodynamics as a theory of decision-making with information-processing costs. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **469**(2153), 20120683 (2013) 4
22. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J.: An Algorithmic Perspective on Imitation Learning, Foundations and Trends in Robotics, vol. 7 (2018) 4
23. Reddy, S., Dragan, A., Levine, S.: Where do you think you're going?: Inferring beliefs about dynamics from behavior. Advances in Neural Information Processing Systems **31** (2018) 2
24. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth in-

ternational conference on artificial intelligence and statistics. pp. 627–635. JMLR Workshop and Conference Proceedings (2011) 9
25. Salvucci, D.D., Gray, R.: A two-point visual control model of steering. Perception **33**(10), 1233–1248 (2004) 2, 5
26. Schwartenbeck, P., FitzGerald, T.H., Mathys, C., Dolan, R., Wurst, F., Kronbichler, M., Friston, K.: Optimal inference with suboptimal models: addiction and active bayesian inference. Medical hypotheses **84**(2), 109–117 (2015) 2
27. Tamar, A., Wu, Y., Thomas, G., Levine, S., Abbeel, P.: Value iteration networks. Advances in neural information processing systems **29** (2016) 6
28. Tishby, N., Polani, D.: Information Theory of Decisions and Actions, pp. 601–636. Springer New York, New York, NY (2011) 4
29. Tschantz, A., Baltieri, M., Seth, A.K., Buckley, C.L.: Scaling active inference. In: 2020 international joint conference on neural networks (ijcnn). pp. 1–8. IEEE (2020) 2
30. Tschantz, A., Seth, A.K., Buckley, C.L.: Learning action-oriented models through active inference. PLoS computational biology **16**(4), e1007805 (2020) 9
31. Wei, R., McDonald, A.D., Garcia, A., Alambeigi, H.: Modeling driver responses to automation failures with active inference. IEEE Transactions on Intelligent Transportation Systems (2022) 1, 2
32. Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al.: Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088 (2019) 2, 11
33. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: Aaai. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008) 2

A Appendix

A.1 Dataset

We used the INTERACTION dataset [32], a publicly available naturalistic dataset recorded with drone footage of fixed road segments, to fit a model of highway car-following behavior. Each recording in the dataset consists of the positions, velocities, and headings of all vehicles in the road segment at a sampling frequency of 10 Hz. Specifically, we used a subset of the data[5] due to the abundance of car-following trajectories and relatively complex road geometry with road curvature and merging lanes. We defined car-following as the trajectory segments from the initial appearance of a vehicle to either an ego lane-change or the disappearance of the lead vehicle. Reducing the dataset using this definition resulted in a total of 1027 car-following trajectories with an average duration of 13 seconds and standard deviation of 8.7 seconds. We obtained driver control actions (i.e., longitudinal and lateral accelerations) by taking the derivative of the velocities of each trajectory. We then created a set of held-out trajectories for testing purposes by first categorizing all trajectories into four clusters based on their kinematic profiles using UMAP [19] and sampled 15% of the trajectories from each cluster.

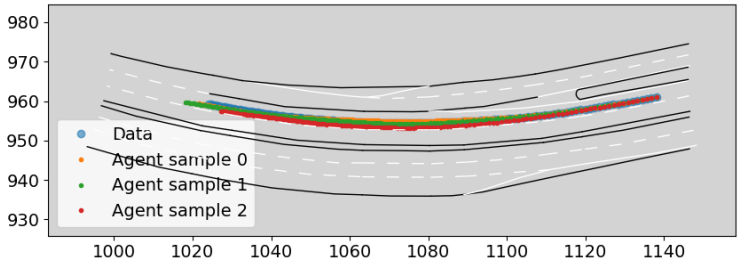
[5] Recording 007 from location "DR_CHN_Merging_ZS"

A.2 Behavior cloning agent

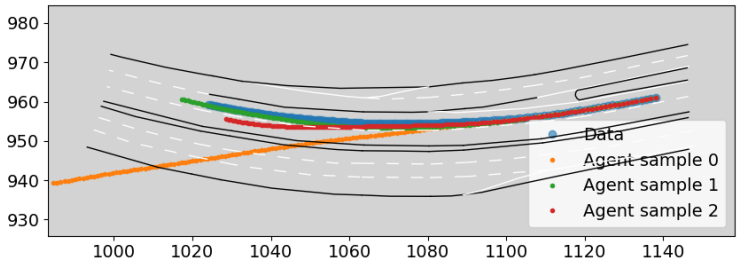
The behavior cloning agents consist of a recurrent neural network with a single gated recurrent unit (GRU) layer and a feed-forward neural network. The GRU layer compresses the observation history into a fixed size vector, which is decoded by the feed-forward network into a continuous action distribution model by a multivariate Gaussian distribution. To make the BC agents comparable to the active inference agents, the GRU has 64 hidden units and 30 output units and the feed-forward network has 30 input units, 2 hidden layers with 64 hidden units, and SiLU activation function. We used the same observation vector as input to the BC agents as to the active inference agents.

A.3 Sample path

Sample path generated by the agents with and without the two-point observation model.



(a) Trajectories generated by agent with the two-point observation model.



(b) Trajectories generated by agent without the two-point observation model.

Fig. 3