Belief state separated reinforcement learning for autonomous vehicle decision making under uncertainty

Ziqing Gu[1†], Yujie Yang[1], Jingliang Duan[1], Shengbo Eben Li[1*], Jianyu Chen[2], Wenhan Cao[1], Sifa Zheng[1]

Abstract— **In autonomous driving, the ego vehicle and its surrounding traffic environments always have uncertainties like parameter and structural errors, behavior randomness of road users, etc. Furthermore, environmental sensors are noisy or even biased. This problem can be formulated as a partially observable Markov decision process. Existing methods lack a good representation of historical information, making it very challenging to find an optimal policy. This paper proposes a belief state separated reinforcement learning (RL) algorithm for decision-making of autonomous driving in uncertain environments. We extend the separation principle from linear Gaussian systems to general nonlinear stochastic environments, where the belief state, defined as the posterior distribution of the true state, is found to be a sufficient statistic of historical information. This belief state is estimated by action-enhanced variational inference from historical information and is proved to satisfy the Markovian property, thus allowing us to obtain the optimal policy using traditional RL algorithms for Markov decision processes. The policy gradient of a task-specific prior model is mixed with that of the interaction data to improve learning performance. The proposed algorithm is evaluated in a multi-lane autonomous driving task, where the surrounding vehicles are subject to behavior uncertainty and observation noise. The simulation results show that compared with existing RL algorithms, the proposed method can achieve a higher average return with better driving performance.**

Index Terms— **autonomous vehicle, Markov decision process, uncertain environment, partially observable**

I. Introduction

Decision-making is crucial for autonomous driving. However, the real-world driving environment is often aggravated by uncertain knowledge and imperfect perception about how the driving process evolves, making it hard to make the right decision. Generally, there are two kinds of uncertainties [1], [2], [3]: (i) process uncertainty and (ii) observation uncertainty. The former refers to the unmodelled high-order dynamics, parametric or structural errors, and behavior randomness of road users. The latter refers to the noisy sensor measurements. These problems of decision-making under uncertainties can be mathematically formulated as a partially observable Markov decision process (POMDP) [4]. This setting is different from POMDP in the motion planning field [5], which does not consider the uncertain parts in the transition model and observation model.

Obtaining the optimal policy in a partially observable environment is challenging. Some algorithms directly summarize the historical information (e.g., past observations and actions) into a hidden state with recurrent functions, such as the deep recurrent Q-network (DRQN) [6] and the variational RNN [7]. In essence, these methods estimate optimal value function and policy function from noisy perception data [8], [9]. However, they place a heavy burden on the recurrent functions, which should tackle the following two problems simultaneously: (i) learning state representation from the historical sequential information, and (ii) learning to maximize the expected return using the learned representation. Prior works [10], [11] have observed the "bottleneck of representation learning ," i.e., a large portion of the learning capability would be spent on obtaining a good representation of the observation space, which remarkably sacrifices the policy performance [4]. Therefore, training a recurrent policy stably and efficiently is still an open question in this end-to-end framework.

Other methods [12], [13], [14], [15] aim to learn an explicit latent dynamic model and then perform RL in the model's learned latent space, such as particle filtering RNN [16] and generative forward model [17]. Interpretability in decision-making is crucial under uncertain environments, especially in autonomous driving [18]. However, both the encoding result of RNN and the learned model's latent state are not interpretable. Furthermore, they provide no guarantee or analysis on the optimality of the learned policy.

This paper aims to extend the separation principle to consider both optimality and interpretability of decision-making under uncertainties. The linear quadratic Gaussian (LQG) is a special case of optimal control which satisfies the separation principle [19]. The optimal policy for such a linear Gaussian system can be separated into a Kalman filter and a linear quadratic regulator for state estimation and control, respectively. In this case, the feedback policy's optimality is theoretically guaranteed by the separation principle, which equals to apply the estimated state in the partially observable setting. However, this is not the case for nonlinear and non-Gaussian systems. In essence, the separation principle takes the sufficient statistic as a boundary to separate the state estimation and control. Therefore, if we can build a Bellman equation of sufficient statistics, the optimal policy could be solved by standard RL techniques, no matter the system is

This study is supported by National Key R&D Program of China with 2020YFB1600200. It is also partially supported by NSF China with 51575293, and U20A20334, and Tsinghua University-Didi Joint Research Center for Future Mobility. All correspondence should be sent to S. Eben Li.

[1]Z. Gu, Y. Yang, J. Duan, S. Eben Li, W. Cao, S. Zheng are with State Key Lab of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, 100084, China. Email: gzq_goodlee@163.com, yangyj17@mails.tsinghua.edu.cn, duanjl15@163.com, lishbo@tsinghua.edu.cn, cwh19@mails.tsinghua.edu.cn, zsf@mail.tsinghua.edu.cn.

[2]J. Chen is with is with Institute of Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. Email: jianyuchen@tsinghua.edu.cn.

linear or nonlinear.

Inspired by the basic principle of separation principle, this paper proposes a belief state separated RL algorithm for decision-making of autonomous driving in uncertain environments. The main contributions of this paper are summarized as follows:

1) A particular sufficient statistics, called belief state, is estimated by action-enhanced variational inference from historical information, which is proved to satisfy the Markovian property and thus allows us to obtain the optimal policy using standard RL algorithms for MDP. The physical meaning of the learned belief state is the distribution of the real state. Therefore, compared with other POMDP studies solved by RL algorithms [20], [21], our method can significantly improve the interpretability by directly learning the real state distribution.
2) Second, we propose the belief state separated RL algorithm to obtain a nearly optimal policy of a general stochastic system under partial observability. In particular, we take the belief state as the input of the value function and policy, and obtain the optimal policy by directly solving the sufficient statistic's Bellman equation. In this way, the problem with uncertainties is broken into two separate parts, a belief state estimator and a deterministic controller. Unlike most existing separated methods, such as LQG [19], the proposed algorithm extends the separation principle from linear Gaussian systems to general nonlinear stochastic systems.
3) Compared with existing POMDP studies [22], which directly take historical observations and actions as inputs, the belief-state based RL method can handle the problem with process uncertainty and observation uncertainty simultaneously. The proposed algorithm is evaluated in a multi-lane autonomous driving scenario with uncertain surrounding vehicles and shows better average return and driving efficiency.

The rest of this paper is organized as follows: Section II states the preliminaries. Section III introduces formulation and implementation of the proposed method. The simulation results are developed in Section IV. Section V summarizes the major contributions and concludes this paper.

II. Belief state separated reinforcement learning framework

This section will discuss how to formulate the problem and build Bellman equation of the sufficient statistic considering process and observation uncertainties.

A. Problem Formulation

A general discrete-time stochastic system with two kinds uncertainties is considered as the following:

$$\begin{aligned} x_{t+1} &= f(x_t,u_t,\xi_t), \\ y_t &= g(x_t,\zeta_t), \end{aligned} \tag{1}$$

where t is the current time, $x_t \in \mathcal{X} \subset \mathbb{R}^m$ is the state, $u_t \in \mathcal{U} \subset \mathbb{R}^n$ is the action, $\xi_t \in \mathbb{R}^m$ is the process uncertainty. $y_t \in \mathcal{Y} \subset \mathbb{R}^l$ is the measurement, $\zeta_t \in \mathbb{R}^l$ is the observation uncertainty, $f(\cdot)$ and $g(\cdot)$ describe the environmental dynamic and the observation model with uncertain parts, respectively.

Since the true state x_t cannot be directly attained, and the observation and action at the current time alone are not enough to recover the environmental information accurately, the policy π of systems in (1) should be represented as a function of historical information h_t, i.e., $u_t = \pi(h_t)$, where h_t contains discrete-time sequence of historical observations and actions:

$$h_t \overset{\text{def}}{=} \left\{y_{1:t}, u_{0:t-1}\right\}. \tag{2}$$

However, the length of the policy input h_t is time-variant, which increases the difficulties of policy representing and learning. Therefore, this paper replaces h_t with a fixed-length sufficient statistic as the state representation, extending the optimality of the separation principle to general systems. Moreover, the sufficient statistic is not unique. We choose the conditional probability distribution of the accurate state, denoted as b_t, as the sufficient statistic for it is meaningful and structurally concise. This kind of sufficient statistic is also called belief state, which satisfies:

$$b_t\left(x_t\right) \overset{\text{def}}{=} p\left(x_t|h_t\right). \tag{3}$$

Then the policy becomes $u_t = \pi(b_t)$. From (3), it further follows that:

$$\begin{aligned} &b_t\left(x_t\right) \\ &= \frac{p(x_t, y_t|h_{t-1}, u_{t-1})p(h_{t-1}, u_{t-1})}{p(y_t|h_{t-1}, u_{t-1})p(h_{t-1}, u_{t-1})} \\ &= \frac{\int p(x_{t-1}|h_{t-1})p(x_t|x_{t-1}, u_{t-1})p(y_t|x_t)\,\mathrm{d}x_{t-1}}{p(y_t|h_{t-1}, u_{t-1})} \\ &= \frac{\int b_{t-1}(x_{t-1})p(x_t|x_{t-1}, u_{t-1})p(y_t|x_t)\,\mathrm{d}x_{t-1}}{\iint b_{t-1}(x_{t-1})p(x_t|x_{t-1}, u_{t-1})p(y_t|x_t)\,\mathrm{d}x_{t-1}\,\mathrm{d}x_t}, \end{aligned} \tag{4}$$

where $p(x_t|x_{t-1}, u_{t-1})$ and $p(y_t|x_t)$ correspond to the environmental transition and the observation model in system (1). Considering the existing uncertainties, we add extra objective functions to make the belief update more suitable for the task in the partially observable environment, which will be introduced in the next section. Furthermore, the new b_t can be iteratively updated using b_{t-1},y_t and u_{t-1} as following:

$$b_t = \mathit{Update}(b_{t-1}, y_t, u_{t-1}). \tag{5}$$

Therefore, a belief-state model (BSM) is explicitly built as a posterior distribution inference model with an underlying state dynamical system, where the probabilistic sequential model is shown in Fig. 1. The probability distribution of belief state conditioned on the previous state and historical information obeys the following equation:

$$\begin{aligned} p(b_t|b_{t-1}, y_t, u_{t-1}) &= \mathbb{I}(b_t = \mathit{Update}(b_{t-1}, y_t, u_{t-1})) \\ &= p(b_t|b_{t-1}, ..., b_0, y_t, u_{t-1}), \end{aligned} \tag{6}$$

where $\mathbb{I}(\cdot)$ is the discrete Dirac function. Obviously, the belief state has the Markovian property, which can be regarded as the state of a belief-based MDP. The Bellman equation of belief state, which is a sufficient statistic learned with parameterized neural networks, is built and solved to attain the optimal policy.

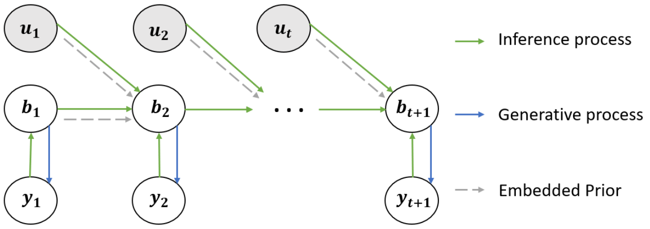


Fig. 1: The probability graph model of belief state

B. Bellman Equation

The belief state satisfies the self-consistent relationship and has the Bellman optimal equation:

$$V^*(b_t)=\min_{u_t}\left\{\mathbb{E}\{l_t|b_t\}+\gamma\mathbb{E}_{b_{t+1}\sim p(b_{t+1}|b_t,u_t)}\{V^*(b_{t+1})\}\right\}, \tag{7}$$

where the expectation, $\mathbb{E}(\cdot)$, represents the instant reward conditioned on historical information. Therefore, the objective function of the optimal control problem could be formulated as:

$$V^{\pi}\left(b_t\right)=\mathbb{E}\left\{\sum_{k=t}^{\infty}\gamma^{k-t}l\left(x_k,\pi\left(b_k\right)\right)|b_t\right\}. \tag{8}$$

Thus, starting from time t, we can compute optimal actions for each arbitrary history information, h_t, by using optimal value functions of belief state, b_t, which satisfies:

$$\begin{aligned}\pi^*(b_t)=\arg\min_{u_t}&\Big\{\mathbb{E}\{l_t|b_t\}+\\&\gamma\mathbb{E}_{b_{t+1}\sim p(b_{t+1}|b_t,u_t)}\{V^*(b_{t+1})\}\Big\}.\end{aligned} \tag{9}$$

C. Actor-Critic Framework

The optimal policy is calculated by approximating the solution of the belief state's Bellman equation under the actor-critic framework. Parameterized networks, $\pi_\omega\left(b_t\right)$ and $V_\theta\left(b_t\right)$, are utilized to approximate the policy and value function, corresponding to the actor and critic, respectively. They are optimized through the process of PEV and PIM. PEV drives the estimated value towards the true value for the current policy. PIM improves the policy according to the estimation of value. PEV and PIM iteratively roll forward and gradually converge to the optimal policy.

Remark 1: The proposed framework builds a connection between stochastic system optimal control and belief-based MDP. In the linear quadratic Gaussian setting, the separation principle separates estimation and control, where the estimation result is a kind of sufficient statistic and equals the belief state under the framework of belief-based MDP. In the latter context, the optimal policy could be directly attained by solving the Bellman equation of the belief state and is not limited by the form of the system. More generally, the sufficient statistic is not unique, where the encoded latent state is also an expression of the approximate sufficient statistic without explicit semantic description.

III. Algorithm

In this section, an internal BSM is explicitly learned and utilized to update the belief state, in which the historical information, interaction data, and empirical process dynamic are jointly considered. The primary motivation is to add a belief optimizer in the training process to reduce the instability only with the RL objective, as shown in Fig.2. In contrast to prior works, our method learns the real state distribution to improve the intrinsic interpretability and mixes the gradients of the model and data to further improve the algorithm performance.

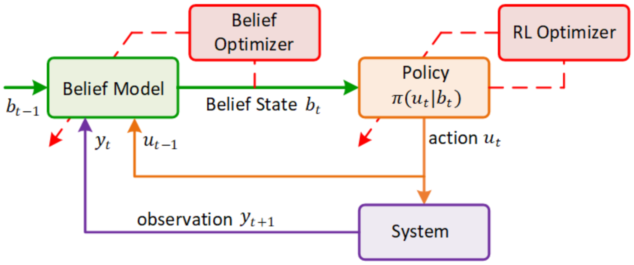


Fig. 2: Belief state separated reinforcement learning algorithm framework

A. Belief State Model

In order to apply the identified system in downstream tasks, we perform belief dynamical identification. In the partially observable setting, we do not have the full knowledge of the real state's transition model, x_t. Therefore, the BSM is trained by maximizing the log-likelihood of observation samples:

$$\begin{aligned}&\max\ (y_{0:t}|u_{0:t-1})\\&\quad=\log\mathbb{E}_{b_{1:t}\sim q(\cdot)}\left\{\frac{p(b_{1:t},y_{1:t}|u_{0:t-1})}{q(b_{1:t}|y_{1:t},u_{0:t-1})}\right\}\\&\quad\geq\mathbb{E}_{b_{1:t}\sim q(\cdot)}\left\{\log\frac{p(b_{1:t}|u_{0:t-1})p(y_{1:t}|b_{0:t-1})}{q(b_{1:t}|y_{1:t},u_{0:t-1})}\right\}\\&\quad=\mathbb{E}_{b_{1:t}\sim q(\cdot)}\{\log p(y_{1:t}|b_{0:t-1})\}-\\&\qquad\qquad\mathcal{KL}[q(b_{1:t}|y_{1:t},u_{0:t-1})||p(b_{1:t}|u_{0:t-1})],\end{aligned} \tag{10}$$

with

$$q(\cdot)=q(b_{1:t}|y_{1:t},u_{0:t-1}), \tag{11}$$

where the lower bound of (10) contains two terms: $\mathbb{E}[\cdot]$ describes the expectation of the mapping from belief state to observation and $\mathcal{KL}[\cdot]$ aims to shorten the distance between the posterior inference, $q(\cdot)$, and the prior transition, $p(\cdot)$. Since the observation is noisy, the second term, $\mathcal{KL}[\cdot]$, can be regarded as a regularization term to reduce the dependence on the generative term of observation. Our objective function of belief state has a similar structure as the basic evidence lower bound [23], but we take action sequence into account to enrich the historical information. We abbreviate the lower bound of (10) as $\mathcal{ELBO}$ and expand it along time series with parameterized form:

$$\begin{aligned}\mathcal{ELBO}(\Phi) = \mathbb{E}_{b_{1:t}\sim q_{\phi}(\cdot)}\Big\{ &\sum_{i=1}^{t} \log p_{\varphi}(y_i|b_i) + \log p_{\psi}(b_1) - \\ &\sum_{i=1}^{t-1} \log q_{\phi}(b_{i+1}|y_{i+1}, b_i, u_i) - \log q_{\phi}(b_1|y_1) + \\ &\sum_{i=1}^{t-1} \log p_{\psi}(b_{i+1}|b_i, u_i)\Big\},\end{aligned} \tag{12}$$

where Φ is the parameter set of the objective function, $\{\phi, \varphi, \psi\} \in \Phi$, and is updated by maximizing (12). p_{φ}, p_{ψ} and q_{ϕ} correspond to observation model, process dynamic model and inference model of belief state, respectively.

The information of task-specific prior is embedded in p_{ψ} so that the prior becomes the driving factor for shaping the belief state inference, rather than adjusting the real state's dynamic to the BSM. The prior satisfies $p_{\psi}(b_t) \sim p\left(x_t|x_{t-1}, u_{t-1}\right)$, as the grey line shown in Fig 1. This operation guarantees the interpretability of belief state. Note that b_t is an approximate distribution of x_t, and x_t only represents the prior deterministic knowledge of state.

B. Belief state separated reinforcement learning

Based on the belief state calculated by BSM, the value functions and policy can be optimized iteratively with PEV and PIM. Similar to Soft Actor-Critic (SAC) [24] algorithm, neural networks are utilized to approximate value function, Q-function and policy. In the process of PEV, value function and Q-function are iteratively updated the same as SAC. The loss function for value function is

$$\begin{aligned}J_V(\theta) = \mathbb{E}_{y_t\sim\mathcal{D}}\Big\{\frac{1}{2}(V_{\theta}(b_t) - (Q_{\nu}(b_t, u_t) - \\ \alpha \log \pi_{\omega}(u_t|b_t)))^2\Big\},\end{aligned} \tag{13}$$

where θ, ν, ω are the parameters of value network, Q-network and policy, respectively, $\mathcal{D}$ is the replay buffer and α is the temperature parameter. The loss function for Q-function is

$$J_Q(\nu) = \mathbb{E}_{(y_t,u_t)\sim\mathcal{D}}\Big\{\frac{1}{2}(Q_{\nu}(b_t, u_t) - \hat{Q}(b_t, u_t))^2\Big\}, \tag{14}$$

with

$$\hat{Q}(b_t, u_t) = l_t + \gamma V_{\bar{\theta}}(b_{t+1}), \tag{15}$$

where $\bar{\theta}$ is the parameter of target value network.

In the process of PIM, the gradient information is vital for the convergence of the RL algorithm. Although the known model here is not accurate, it can be appropriately utilized at the beginning to accelerate training. We adopt a mixed gradient method in PIM to balance the inaccurate model information and collected observation data. It mainly concentrates on the model-based gradient at the beginning of training while giving more weight to the data-based gradient in the later period. The former can be obtained with n-step Bellman recursion in (8):

$$\begin{aligned}J_{\pi,model}(\omega) = \mathbb{E}_{y_t\sim\mathcal{D},u_t\sim\pi_{\omega}}\Big\{\sum_{k=t}^{n+t}\gamma^{k-t} l\left(b_k, u_k\right) + \\ \gamma^{n+1} V(b_{t+n+1})\Big\}.\end{aligned} \tag{16}$$

The latter is calculated using the same method as SAC:

$$J_{\pi,data}(\omega) = \mathbb{E}_{y_t\sim\mathcal{D},u_t\sim\pi_{\omega}}\Big\{\alpha \log \pi_{\omega}(u_t|b_t) - Q_{\nu}(b_t, u_t)\Big\}. \tag{17}$$

Therefore, the policy objective could be written as:

$$J_{\pi,mix} = \rho_{pg} J_{\pi,data} + (1 - \rho_{pg}) J_{\pi,model}. \tag{18}$$

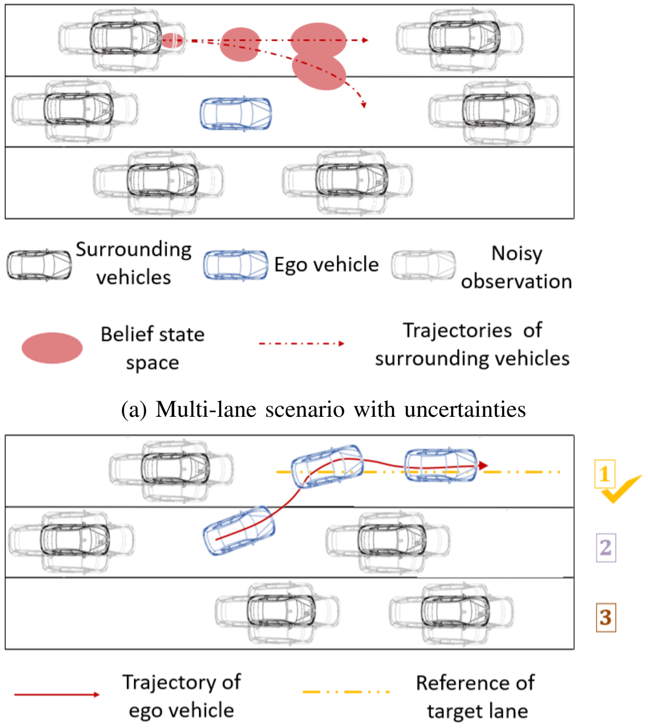
where the ρ_{pg} is the gradient weight between model and data. The complete algorithm is described in Algorithm 1.

Algorithm 1: Belief state separated algorithm

```
Initialize network parameters $\theta, \nu_1, \nu_2, \omega, \Phi$.
Initialize target network parameter $\bar{\theta} \leftarrow \theta$.
Initialize iterative step $k$, update interval m, batch
  size N, target smoothing coefficient $\tau$, learning rates
  $\beta_\theta$, $\beta_\nu$, $\beta_\omega$, $\beta_\Phi$, temperature coefficient $\alpha$, mixed
  weight $\rho_{pg}$.
Initialized the replay buffer $\mathcal{D}$.
Initialize the environment, get the initial observation
  $y_0$ for agent.
for each iteration do
    for each environment step do
        Collect a transition $(y, u, l, y')$ using policy
          $\pi_\omega$ and store it in the replay buffer $\mathcal{D}$.
    end
    for each update step do
        Sample N transitions from replay buffer $\mathcal{D}$.
        Update the belief model:
        $\Phi \leftarrow \Phi - \beta_\Phi \nabla_\Phi \mathcal{L}_{belief}$
        // PEV based on the belief
           state
        Update the value network and Q networks:
        $\nu \leftarrow \nu - \beta_\nu \nabla_\nu J_Q(\nu)$
        $\theta_i \leftarrow \theta_i - \beta_\theta \nabla_{\theta_i} J_V(\theta_i)$ for $i \in \{1, 2\}$
        // PIM based on the belief
           state
        Update the policy network:
        $\omega \leftarrow \omega - \beta_\omega \nabla_\omega J_{\pi,mix}(\omega)$
        Update target network:
        $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$ for $i \in \{1, 2\}$
        Adjust temperature coefficient $\alpha$ and mixed
          weight $\rho_{pg}$.
    end
end
```

IV. Experiment

This section compares the proposed method with other algorithms in the multi-lane scenario, where the uncertainties exist in the behavior and observation of surrounding vehicles.



(a) Multi-lane scenario with uncertainties

(b) Lane-changing process

Fig. 3: Multi-lane scenario (a) represents typical scenario with uncertainties existing in the surrounding vehicles. (b) describes the process of once lane-changing according to the reference target lane, where the number represents the lane index.

A. Simulation Environment

We focus on a typical three-lane scenario, as shown in Fig. 3 (a). We regard state transitions of ego vehicle and surrounding vehicles as a whole:

$$\begin{aligned} \begin{bmatrix} x_{t+1}^{self} \\ x_{t+1}^{surr} \end{bmatrix} &= \begin{bmatrix} f^{self}(x_t^{self} - x^{ref}, u_t) \\ f^{surr}(x_t^{surr}, \xi_t) \end{bmatrix}, \\ \begin{bmatrix} y_{t+1}^{self} \\ y_{t+1}^{surr} \end{bmatrix} &= \begin{bmatrix} g^{self}(x_t^{self}) \\ g^{surr}(x_t^{surr}, \zeta_t) \end{bmatrix}, \end{aligned} \tag{19}$$

where f^{self} is ego vehicle's (blue cars in Fig. 3) dynamic model. It consists of the bicycle model with a linear tire model and is discretized with the backward Euler method [25]. The corresponding observation model g^{self} is accurate. The partially observable settings are reflected in the facts that the model of surrounding vehicles (grey cars in Fig. 3) f^{surr} is uncertain and there exists observation noise in g^{surr}. The surrounding traffic flow is generated by the *Simulation of Urban Mobility* (SUMO) platform. Here, we assume the prior model of surrounding vehicles is a longitudinal model with uniform velocity, while in fact, they can both achieve longitudinal acceleration or deceleration and lateral movements.

For state space, the x^{surr} contains position and velocity of the nearest 6 surrounding vehicles with noise. If the number of surrounding vehicles is less than 6, virtual vehicles are added at the boundary of perception range. The x^{self} contains ego vehicle's relative position, velocity, heading angle and yaw rate with target lane, x^{ref}, where the x^{ref} is chosen by the trained value network, as shown in Fig. 3 (b).

We choose the acceleration and steering angle as actions and suppose that they are strictly limited to a reasonable bound. In total, we construct 27-dimensional continuous state space (18-dimension for surrounding vehicles and 9-dimension for ego vehicle) and 2-dimensional continuous action space.

The reward function is designed to simultaneously consider driving safety, stable speed and tracking performance of the target lane. This task is constructed in an episodic manner, where terminal conditions contain: collision, out of lane, steering angle, or tracking error out of bound and task completion.

B. Algorithm Details

We compare the proposed algorithm with the RNN-based method to evaluate the advantages of the belief state. The explicit semantic description of b_t is reflected in which the dimension of belief states is the same as the real states, corresponding to the state distribution. Table I describes the algorithm settings of the proposed method and baselines, i.e.,

- BFMIX: the belief state, b_t, is described by BSM, the gradient weight, ρ_{pg}, in (18) is time-variant from 0 to 1.
- BFSAC: the belief state, b_t, is described by BSM, and the gradient weight, ρ_{pg}, in (18) is 1.
- RNNSAC: the historical information, h_t, is encoded by the Gate Recurrent Unit (GRU), and the gradient weight, ρ_{pg}, in (18) is 1.

TABLE I: Algorithm Setting

Algorithm	Sufficient statistic	Gradient Information
BFMIX(variant ρ_{pg})	Belief Model	prior model and data
BFSAC($\rho_{pg} = 1$)	Belief Model	only data
RNNSAC	RNN encode	only data

The belief-based method has an explicit belief model to describe the real state distribution, approximated by multi-layer perceptron (MLP) with two hidden layers. The RNN-based method utilizes the GRU to encode historical information and generate latent states. The value function (of state and state-action) and policy function are approximated by MLP with three hidden layers. Besides, we use MLP as the feature net to encode surrounding vehicles with different input orders. See Table II for more details.

C. Result Analysis

1) Performance comparison: The comparative performance of algorithms are shown in Fig. 4. The results show that the belief-based methods (BFMIX and BFSAC) obtain a higher mean concerning the average return than the RNN-based method (RNNSAC). Compared with the encoded state

TABLE II: Training hyperparameters

Item	Value
Replay Buffer Size	1e6
Sample Batch Size	32
Hidden Layers Activation	ReLU
Number of Hidden Units	128
Optimizer Type	Adam
Actor Learning Rate	3e-4
Critic Learning Rate	3e-4
Learning Rate Schedule	Anneal linearly to 1e-5
Discount Factor γ	0.99
Target Update Rate τ	0.005
Temperature α	Init = 0.2, auto tune mode
Expected Entropy	-Action Dimensions

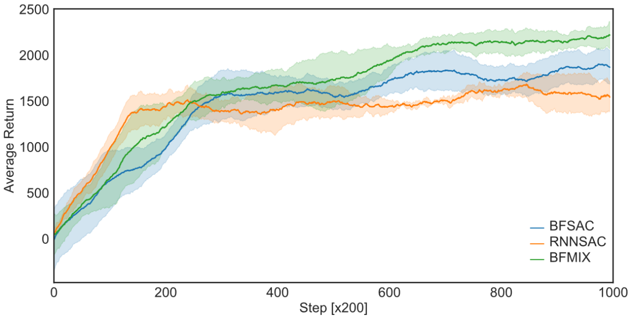
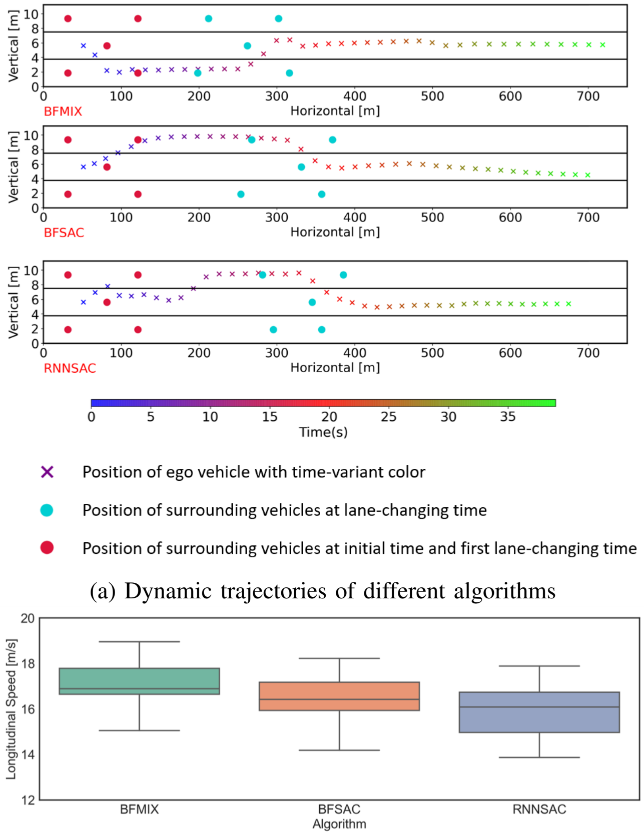


Fig. 4: Average return during the training process. The solid lines and the shaded regions correspond to the mean and 98% confidence interval over three runs, respectively.

of RNNs without explicit semantic description, the expressive ability of the belief state to real state improves the algorithm performance. Although the convergence rate of BFMIX is reduced in the early period affected by the gradient of the inaccurate prior model, compared with BFSAC, the average return is higher at the convergence stage. It could be explained that the task-specific prior model helps to accumulate more effective interactive data to some degree.

2) Simulation result: We compare different policies of the above algorithms in twice lane-changing processes, as shown in Fig. 5. The dynamic trajectory of the ego vehicle and the position of surrounding vehicles within the observation range at different times are shown in Fig. 5 (a). It can be figured out that in the same period, the policy of BFMIX can make the vehicle drive further distance by lane-changing decision. The policies of BFMIX and BFSAC can perform two complete lane-changing processes, while that of RNNSAC hesitates at the first time. According to the speed distribution of different algorithms in Fig. 5 (b), the average speed of the belief-based method is higher, where the average speed of BFMIX, BFSAC, and RNNSAC are 17.13m/s, 16.49m/s, and 16.02m/s, respectively. It can be concluded that the belief state could improve the algorithm performance with better driving efficiency.

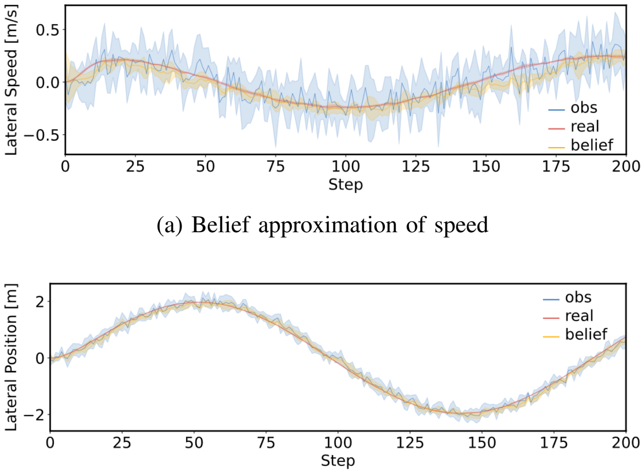
3) Belief: The advantage of the belief state is the explicit physical meaning, which is crucial to the interpretability in control. As shown in Fig. 6, the belief state can describe the distribution of real state better, while the encoded output of RNNs is not a one-to-one map of the real state and has no explicit physical meaning. Moreover, we utilize MLPs to express the belief state, the parameter number of which is much smaller than that of RNN.



(a) Dynamic trajectories of different algorithms

(b) Distributions of longitudinal speed

Fig. 5: State analysis in Multi-lane scenario



(a) Belief approximation of speed

(b) Belief approximation of position

Fig. 6: Interpretability of belief state

V. Conclusion

This paper proposed a belief state separated reinforcement learning algorithm for autonomous driving under uncertain environments, where an action-enhanced variational inference method was utilized to learn an explicit belief-state model from non-Markovian historical data. It performed sufficient statistics identification and extended the separation principle from linear Gaussian systems to general nonlinear systems, which allows obtaining the optimal policy for POMDP problems using standard RL algorithms for MDPs. Simulation results in the multi-lane autonomous driving tasks, which contained behavior uncertainty and observation noises of surrounding vehicles, showed that our algorithm improved the average learning return with better driving performance.

References

[1] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 392–399.

[2] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1671–1678.

[3] Y. Mu, B. Peng, Z. Gu, S. E. Li, C. Liu, B. Nie, J. Zheng, and B. Zhang, "Mixed reinforcement learning for efficient policy optimization in stochastic environments," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2020, pp. 1212–1219.

[4] S. E. Li, *Reinforcement Learning and Control*. Tsinghua University Lecture Notes, 2020. [Online]. Available: http://www.idlab-tsinghua.com/thulab/labweb/publications.html

[5] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[6] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *AAAI Fall Symposia*, 2015.

[7] D. Han, K. Doya, and J. Tani, "Variational recurrent models for solving partially observable control tasks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=r1lL4a4tDB

[8] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber, "Solving deep memory pomdps with recurrent policy gradients," in *International conference on artificial neural networks*. Springer, 2007, pp. 697–706.

[9] S. Carr, N. Jansen, R. Wimmer, A. Serban, B. Becker, and U. Topcu, "Counterexample-guided strategy improvement for pomdps using recurrent neural networks," in *IJCAI*, 2019.

[10] Y. Ren, J. Duan, S. E. Li, Y. Guan, and Q. Sun, "Improving generalization of reinforcement learning with minimax distributional soft actor-critic," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[11] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell, "Loss is its own reward: Self-supervision for reinforcement learning," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[12] M. Watter, J. T. Springenberg, J. Boedecker, and M. A. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," in *NIPS*, 2015.

[13] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine, "Solar: Deep structured representations for model-based reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7444–7453.

[14] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=S1lOTC4tDS

[15] M. Okada, N. Kosaka, and T. Taniguchi, "Planet of the bayesians: Reconsidering and improving deep planning network by incorporating bayesian inference," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5611–5618, 2020.

[16] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for pomdps," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2117–2126.

[17] L. Buesing, T. Weber, S. Racanière, S. M. A. Eslami, D. J. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, and D. Wierstra, "Learning and querying fast generative models for reinforcement learning," *CoRR*, vol. abs/1802.03006, 2018.

[18] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[19] Y. Zhou, S. Ahn, M. Chitturi, and D. A. Noyce, "Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 83, pp. 61–76, 2017.

[20] T. Kim, S. Ahn, and Y. Bengio, "Variational temporal abstraction," in *NeurIPS*, 2019.

[21] D. Hafner, T. Lillicrap, I. S. Fischer, R. Villegas, D. R. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," *ArXiv*, vol. abs/1811.04551, 2019.

[22] M. Deisenroth and C. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *ICML*, 2011.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80. PMLR, 2018, pp. 1861–1870.

[25] Q. Ge, S. E. Li, Q. Sun, and S. Zheng, "Numerically stable dynamic bicycle model for discrete-time control," *arXiv preprint arXiv:2011.09612*, 2020.