

Mixed Reinforcement Learning with Additive Stochastic Uncertainty

Yao Mu, Shengbo Eben Li*, Chang Liu,
Qi Sun, Bingbing Nie, Bo Cheng, and Baiyu Peng

Abstract—Reinforcement learning (RL) methods often rely on massive exploration data to search optimal policies, and suffer from poor sampling efficiency. This paper presents a mixed reinforcement learning (mixed RL) algorithm by simultaneously using dual representations of environmental dynamics to search the optimal policy with the purpose of improving both learning accuracy and training speed. The dual representations indicate the environmental model and the state-action data: the former can accelerate the learning process of RL, while its inherent model uncertainty generally leads to worse policy accuracy than the latter, which comes from direct measurements of states and actions. In the framework design of the mixed RL, the compensation of the additive stochastic model uncertainty is embedded inside the policy iteration RL framework by using explored state-action data via iterative Bayesian estimator (IBE). The optimal policy is then computed in an iterative way by alternating between policy evaluation (PEV) and policy improvement (PIM). The convergence of the mixed RL is proved using the Bellman's principle of optimality, and the recursive stability of the generated policy is proved via the Lyapunov's direct method. The effectiveness of the mixed RL is demonstrated by a typical optimal control problem of stochastic non-affine nonlinear systems (i.e., double lane change task with an automated vehicle).

Index Terms—Reinforcement learning, Bayesian estimation, Policy evaluation (PEV), Policy improvement (PIM), Dynamic model

I. INTRODUCTION

REINFORCEMENT learning (RL) has been successfully applied in a variety of challenging tasks, such as Go game and robotic control [1], [2]. The increasing interest in RL is primarily stimulated by its data-driven nature, which requires little prior knowledge of the environmental dynamics, and its combination with powerful function approximators, e.g. deep neural networks. In spite of these advantages, many purely data-driven RL suffers from slow convergence rate in continuous action space of stochastic systems, which hinders its widespread adoption in real-world applications [3], [4].

To alleviate this problem, researchers have investigated the use of model-driven RL algorithms, which searches the

optimal policy with known environmental models by employing the principle of Bellman optimality [5]–[8]. Model-driven RL has shown faster convergence compared to the data-driven counterparts, since environmental models provide the information of environmental evolution in the whole state-action space. Thus, gradient calculation can be easier and more accurate than merely using data samples [9]. To solve the Bellman equation in the continuous action space, most existing RL methods adopt an iterative technique to gradually find the optimum. One classic framework is called policy iteration RL, which consists of two steps: 1) policy evaluation (PEV), that aims at solving self-consistency condition equation and evaluating the current policy, and 2) policy improvement (PIM) that seeks to optimize the corresponding value function [10], [11].

A number of prior works focus on improving the PEV step by using model-driven value expansion, which corrects the cumulative return or the approximated value function by using environmental models [12], [13]. However, due to the inherent model inaccuracy, this technique is not suitable for long-term PEV. To partly solve this problem, model-based value expansion algorithm proposed a hybrid algorithm that uses environmental dynamic model only to simulate the short-term horizon, and utilizes the explored data to estimate the long-term value beyond the simulation horizon [14]. Nevertheless, the inaccuracy problem hinders the application of environmental model in PEV.

So far, the environmental model has limited applications in the PIM step due to two main issues: 1) the inaccuracy and overfitting of environmental dynamic models and 2) policy oscillation caused by the time-varying models, since the system model is iteratively learned or updated in the training process [15]–[17]. Prior works provide the model ensemble technique for solving these problems. For example, the model-ensemble trust region policy optimization (TRPO) algorithm [18] limits model over-training by using an ensemble metric during policy search. The stochastic ensemble value expansion [19], which is an extension to the model-based value expansion, interpolates between many different horizon lengths and different models to favor models that generate more accurate estimates. Although the ensemble techniques effectively avoid over-fitting, it brings extra computational overhead.

Facing the aforementioned challenges of RL algorithms, this paper proposes a mixed reinforcement learning (mixed RL) algorithm that utilizes the dual representations of environmental dynamics to improve both learning accuracy and training speed. The environmental model, either empirical or theoretic-

This study is supported by International Science & Technology Cooperation Program of China under 2019YFE0100200, Beijing Natural Science Foundation with JQ18010. All correspondence should be sent to S. Li.

Y. Mu, S. Li, Q. Sun, B. Nie, B. Cheng and B. Peng are with State Key Lab of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, 100084, China. (Email: {muy18@mails.;lishbo@qisun@mail.}tsinghua.edu.cn).

C. Liu is with Sibley School of Mechanical and Aerospace Engineering, Cornell University, New York, 14853, USA. (Email: cl775@cornell.edu).

cal, is used as the prior information to avoid overfitting, while the model error is iteratively compensated by the measured data of states and actions using Bayesian estimation. Precisely, the contributions of this paper are as follows,

- 1). A dual representation of environmental dynamics is utilized in RL by integrating the designer's knowledge with the measured data. An iterative Bayesian estimator (IBE) with explored data is designed for improving the model accuracy and computation efficiency.
- 2). A mixed RL algorithm is developed by embedding the iterative Bayesian estimator into the policy iteration. We propose the sufficient recursive stability and convergence condition which limits the estimated difference of iterative Bayesian estimator between two consecutive iterations. And we proved that the sufficient condition holds with probability one after sufficient iterations.

The rest of this paper is organized as follows. Section II defines a mixed RL problem. Section III introduces the mixed representation of environmental dynamics. Section IV and Section V presents the mixed RL algorithm, as well as the parametrization of the policy and value function. Section VI evaluates the effectiveness of mixed RL problem using the double lane change task with a automated vehicle, and Section VII concludes this paper.

II. PROBLEM DESCRIPTION

We consider a discrete-time environment with additive stochastic uncertainty and its actual dynamics is mathematically described as

$$\begin{aligned} x_{t+1} &= f(x_t, u_t) + \xi_t, \\ \xi_t &\sim N(\mu, \mathcal{K}) \end{aligned} \quad (1)$$

where t is the current time, $x_t \in \mathcal{X} \subset \mathbb{R}^n$ is the state, $u_t \in \mathcal{U} \subset \mathbb{R}^m$ is the action, $f(\cdot, \cdot)$ is the deterministic part of environmental dynamics, $\xi_t \in \mathbb{R}^n$ is the additive stochastic uncertainty with unknown mean $\mu \in \mathbb{R}^n$ and covariance $\mathcal{K} \in \mathbb{R}^{n \times n}$. In this study, we assume that the additive stochastic uncertainty ξ_t follows the Gaussian distribution and $\mathbb{E}\{|\xi_t|\} < \infty$. Parameters μ and \mathcal{K} can be completely independent of (x, u) or form a functional relationship with (x, u) .

As shown in Fig. 1, actual environmental dynamics contains both deterministic part $f(\cdot, \cdot)$ and uncertain part ξ_t , where $p(\xi_t)$ is the probability density of ξ_t and $p(x_{t+1})$ is the probability density of x_{t+1} under given (x_t, u_t) .

The objective of mixed RL is to minimize the expectation of cumulative cost under the distribution of additive stochastic uncertainty ξ , shown as (2):

$$\min_{\pi} V(x_t) = \mathbb{E}_{\xi} \left\{ \sum_{k=0}^{\infty} \gamma^k l(x_{t+k+1}, u_{t+k}) \right\}, \quad \forall x_t \in \mathcal{X} \quad (2)$$

where π is policy, $V(\cdot)$ is the state value, which is a function of initial state x_t , $l(\cdot, \cdot) \geq 0$ is the utility function, which is positive definite, γ is the discounting factor with $0 < \gamma < 1$,

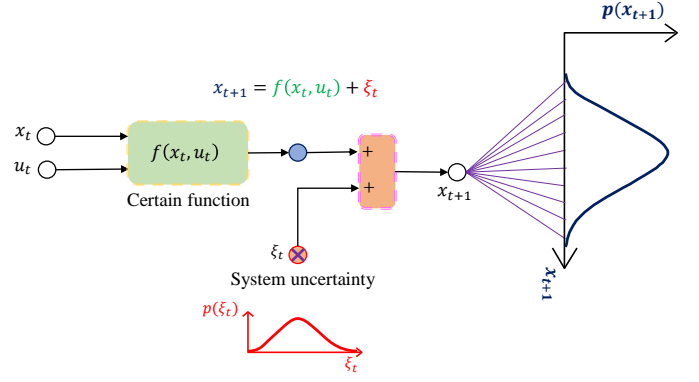


Fig. 1. Dynamics for the stochastic environment.

1, and $\mathbb{E}_{\xi}(\cdot)$ is the expectation w.r.t. the additive stochastic uncertainty ξ . Here, the policy is a deterministic mapping:

$$u_t = \pi(x_t) \quad (3)$$

The optimal cost function is defined as

$$V^*(x_t) = \inf_{\{u_t, u_{t+1}, \dots, u_{\infty}\}} V(x_t) \quad (4)$$

where $\{u_t, u_{t+1}, \dots, u_{\infty}\}$ is the action sequence starting from time t . In mixed RL, the self-consistency condition (5) is used to describe the relationship of state values between current time and next time:

$$V(x_t) = \mathbb{E}_{\xi} \{ l(x_{t+1}, u_t) + \gamma V(x_{t+1}) \} \quad (5)$$

By using Bellman's principle of optimality, we have the well-known Bellman equation:

$$V^*(x_t) = \min_{u_t} \{ \mathbb{E}_{\xi} \{ l(x_{t+1}, u_t) + \gamma V^*(x_{t+1}) \} \} \quad (6)$$

The Bellman equation implies that optimal policy can be calculated in a step-by-step backward manner. Therefore, optimal action is

$$\pi^*(x_t) \stackrel{\text{def}}{=} \arg \min_{u_t} \{ \mathbb{E}_{\xi} \{ l(x_{t+1}, u_t) + \gamma V^*(x_{t+1}) \} \} \quad (7)$$

where $\pi^*(\cdot)$ represent the optimal policy that maps from an arbitrary state x to its optimal action u^* . Similar to other indirect RL problems, mixed RL aims to find an optimal policy by minimizing cost (2) while being subjected to the constraints of environmental dynamics. The searching procedure can be replaced by solving the Bellman equation in an iterative way. Obviously, the performance of the generated policy depends on the accuracy of the representation of the environmental dynamics. In fact, either an analytical model or state-action samples $(x_1, u_1, \dots, x_t, u_t, \dots)$ can be an useful representation, which corresponds to the so-called model-driven RL and data-driven RL, respectively. The analytical model is usually inaccurate due to environmental uncertainties, which will impair the optimality of the generated policy. The state-action samples, on the other hand, have low sampling efficiency and will slow down the training process.

III. DUAL REPRESENTATION OF ENVIRONMENTAL DYNAMICS

In mixed RL, the environmental dynamics are dually represented by both an analytical model \mathcal{M} and state-action data \mathcal{D} . The former represents the designer's knowledge about the environmental dynamics. It is defined in the whole state-action space and can be used to accelerate the training speed. The latter comes from direct measurement of state-action pairs during learning. It is generally more accurate than \mathcal{M} , and therefore can improve the estimation of the uncertain part in the analytical model. The mixed RL uses the dual representation of environmental dynamics, i.e., both analytical model \mathcal{M} and state-action data \mathcal{D} , to search for optimal policy. Such dual representation can have accelerated training compared to purely data-driven RL while achieving better policy satisfaction than purely model-driven counterpart.

The analytical model \mathcal{M} is similar to (1):

$$\mathcal{M} = \{x_{t+1} = f(x_t, u_t) + \xi_t^{\mathcal{M}}\} \quad (8)$$

$$\xi_t^{\mathcal{M}} \sim N(\mu_{\mathcal{M}}, \mathcal{K}_{\mathcal{M}})$$

where the mean $\mu_{\mathcal{M}}$ and covariance $\mathcal{K}_{\mathcal{M}}$ of $\xi_t^{\mathcal{M}}$ are given in advance by designers. The given distribution can be quite different from actual distribution due to the modelling errors. Here, $\mu_{\mathcal{M}}$ and $\mathcal{K}_{\mathcal{M}}$ are taken as the prior knowledge of environmental dynamics.

The state-action data, i.e., a sequence of triples (x_j, u_j, x_{j+1}) , is denoted by \mathcal{D} :

$$\mathcal{D} = \{(x_j^{\mathcal{D}}, u_j^{\mathcal{D}}, x_{j+1}^{\mathcal{D}}), j = 1, 2, \dots, N\} \quad (9)$$

where $x_j^{\mathcal{D}}$ is the j -th state in \mathcal{D} , $u_j^{\mathcal{D}}$ is the j -th action in \mathcal{D} , and N is the length of data samples. Obviously, the measured data also inherently contain the distribution information of ξ , and are taken as the posterior knowledge of environmental dynamics.

If the environmental dynamics is exactly known, optimal policy $\pi^*(\cdot)$ can be computed by only using the dynamic model, which is also the most efficient RL. However, exact model is inaccessible in reality, and thus the generated policy might not converge to $\pi^*(\cdot)$. Although collecting samples \mathcal{D} is less efficient, it can be quite accurate to represent the environment, thus being able to improve the generated policy. Therefore, the mixed representation is able to utilize advantages of both model \mathcal{M} and data \mathcal{D} to improve training efficiency and policy accuracy.

Improve model \mathcal{M} by using data \mathcal{D} :

We utilize data samples to improve the estimation of the additive stochastic uncertainty ξ in the analytical model \mathcal{M} . The uncertainty that inherently exists in a state-action triple is equal to

$$\xi_j^{\mathcal{D}} = x_{j+1}^{\mathcal{D}} - f(x_j^{\mathcal{D}}, u_j^{\mathcal{D}}) \quad (10)$$

A Bayesian estimator is adopted to fuse the distribution information of the additive stochastic uncertainty from both model \mathcal{M} and data \mathcal{D} . The Bayesian estimator aims to maximize the posterior probability $p(\mu, \mathcal{K}|\mathcal{D})$. In general, we

introduce $p(\mu)$ and $p(\mathcal{K})$ as the prior distribution of μ and \mathcal{K} , then the maximum likelihood problem becomes

$$\begin{aligned} & \max_{\mu, \mathcal{K}} \{p(\mu, \mathcal{K}|\mathcal{D})\} \\ & \Leftrightarrow \max_{\mu, \mathcal{K}} \{p(\mathcal{D}|\mu, \mathcal{K}) p(\mu) p(\mathcal{K})\} \end{aligned} \quad (11)$$

Under the assumption that data \mathcal{D} is iid, (11) can be rewritten into iterative form:

$$\begin{aligned} & \max_{\mu, \mathcal{K}} \{p(\xi_k^{\mathcal{D}}|\mu, \mathcal{K}) p(\mathcal{D}_{k-1}|\mu, \mathcal{K}) p(\mu) p(\mathcal{K})\} \\ & \mathcal{D}_{k-1} = \{\xi_1^{\mathcal{D}}, \xi_2^{\mathcal{D}}, \dots, \xi_{k-1}^{\mathcal{D}}\} \end{aligned} \quad (12)$$

Therefore, we can build an iterative Bayesian estimator IBE(\cdot) with the following general form,

$$\begin{bmatrix} \mu_k \\ \mathcal{K}_k \end{bmatrix} = \text{IBE}(\mu_{k-1}, \mathcal{K}_{k-1}, \xi_k^{\mathcal{D}}) \quad (13)$$

Here, we discuss two simplified cases of the Bayesian estimator:

Case 1: Assume that the covariance \mathcal{K} is known and μ is independent from x and u , we introduce $\mu \sim N(\mu_{\mathcal{M}}, \mathcal{K}_{\mathcal{M}})$ provided by model \mathcal{M} as the prior distribution of μ . Thus, the objective function \mathcal{L} of Bayesian estimation becomes,

$$\begin{aligned} \mathcal{L} &= \log \{p(\mathcal{D}|\mu) p(\mu)\} \\ &= \frac{1}{2} (\mu - \mu_M)^T \mathcal{K}_M^{-1} (\mu - \mu_M) \\ &\quad + \frac{1}{2} \sum_{j=1}^N (\xi_j^{\mathcal{D}} - \mu)^T \mathcal{K}^{-1} (\xi_j^{\mathcal{D}} - \mu) + \mathcal{C} \end{aligned} \quad (14)$$

where $p(\mu) = \mathcal{N}(\mu_M, \mathcal{K}_M)$ is the prior distribution and \mathcal{C} is a constant. The optimal estimation of μ is calculated by (15).

$$\hat{\mu} = (\mathcal{K}_M^{-1} + N\mathcal{K}^{-1})^{-1} \left(\mathcal{K}_M^{-1} \mu_M + \mathcal{K}^{-1} \sum_{j=1}^N \xi_j^{\mathcal{D}} \right) \quad (15)$$

The $\hat{\mu}$ can be iteratively computed by using IBE. Define $\Psi_k = \mathcal{K}_M^{-1} + k\mathcal{K}^{-1}$, and $m_k = \sum_{j=1}^k \xi_j^{\mathcal{D}}$, the iterative Bayesian estimator IBE(\cdot) is

$$\begin{aligned} \Psi_k &= \Psi_{k-1} + \mathcal{K}^{-1}, \quad m_k = m_{k-1} + \xi_k^{\mathcal{D}} \\ \hat{\mu}_k &= (\Psi_k)^{-1} (\mathcal{K}_M^{-1} \mu_M + \mathcal{K}^{-1} m_k) \end{aligned} \quad (16)$$

Case 2: Assume that both the mean μ and covariance \mathcal{K} are unknown. The same prior distribution in case 1 is applied to μ . The covariance \mathcal{K} is estimated by the maximum likelihood estimation, since the parameters of the prior distribution of \mathcal{K} are inconvenient to determine by human designer. Subsequently, the optimal estimation of μ and \mathcal{K} are as follows,

$$\begin{aligned} \hat{\mu} &= (\mathcal{K}_M^{-1} + N\hat{\mathcal{K}}^{-1})^{-1} \left(\mathcal{K}_M^{-1} \mu_M + \hat{\mathcal{K}}^{-1} \sum_{j=1}^N \xi_j^{\mathcal{D}} \right) \\ \hat{\mathcal{K}} &= \frac{1}{N} \sum_j (\xi_j^{\mathcal{D}} - \hat{\mu}) (\xi_j^{\mathcal{D}} - \hat{\mu})^T \end{aligned} \quad (17)$$

Define $\Psi_k = \mathcal{K}_M^{-1} + k\hat{\mathcal{K}}_k^{-1}$ and $m_k = \sum_{j=1}^k \xi_j^D$. Then $\hat{\mu}$ and $\hat{\mathcal{K}}$ can be iteratively computed by the following IBE,

$$\begin{aligned}\Psi_k &= \Psi_{k-1} + \hat{\mathcal{K}}_{k-1}^{-1} \\ m_k &= m_{k-1} + \xi_k^D \\ \hat{\mu}_k &= (\Psi_k)^{-1} (\mathcal{K}_M^{-1} \mu_M + \hat{\mathcal{K}}_{k-1}^{-1} m_k) \\ \hat{\mathcal{K}}_k &= \frac{1}{k} \left\{ (k-1)\hat{\mathcal{K}}_{k-1} + (\xi_k^D - \hat{\mu}_{k-1}) (\xi_k^D - \hat{\mu}_{k-1})^T \right\}\end{aligned}\quad (18)$$

For more general cases where μ is related to x and u , i.e. $\mu = \phi(x, u; w_\phi)$, where $\phi(\cdot)$ is a general function with parameter w_ϕ , the likelihood \mathcal{L} becomes (19) and the optimal estimation of w_ϕ is the minimum of \mathcal{L} .

$$\begin{aligned}p(\mu) &\propto p(w_\phi) \\ \mathcal{L} &= \log \{p(\mathcal{D}|\mu = \phi(x, u; w_\phi))p(w_\phi)\}\end{aligned}\quad (19)$$

IV. MIXED RL ALGORITHM

A. Mixed RL Algorithm Framework

Existing RL algorithms that compute the optimal policy via the use of Bellman equation are known as indirect RL and they usually involve PEV and PIM steps. Different from traditional indirect RL algorithms, mixed RL consists of three alternating steps, i.e., IBE, PEV and PIM, as shown in Fig. 2. IBE that is proposed in Section III is used to estimate the mean and covariance of the additive stochastic uncertainty iteratively. PEV seeks to numerically solve a group of algebraic equations governed by the self-consistency condition (5) under current-step policy π , and PIM is to search a better policy by minimizing a “weak” Bellman equation.

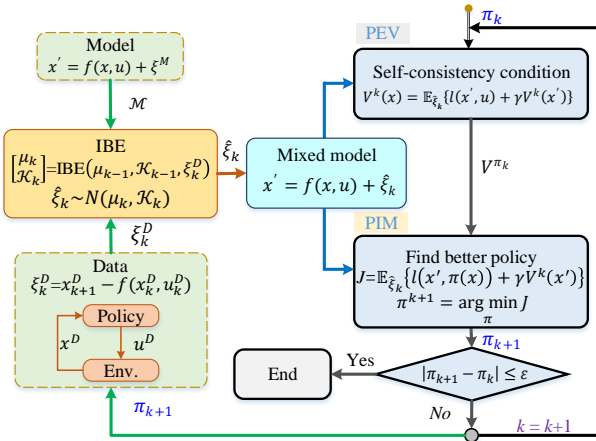


Fig. 2. The framework of the mixed RL algorithm.

In the first step, IBE calculates μ_k and \mathcal{K}_k with the latest data ξ_k^D and the mixed model is updated accordingly, i.e.,

$$x' = f(x, u) + \hat{\xi}_k, \quad \hat{\xi}_k \sim N(\mu_k, \mathcal{K}_k) \quad (20)$$

where $\begin{bmatrix} \mu_k \\ \mathcal{K}_k \end{bmatrix} = \text{IBE}(\mu_{k-1}, \mathcal{K}_{k-1}, \xi_k^D)$ is defined in (13). The optimal policy is searched by policy iteration with the

mixed model (20). In the second step, PEV solves (21) under the estimated distribution of ξ :

$$V^k(x) = \mathbb{E}_{\hat{\xi}_k} \{l(x', \pi^k(x)) + \gamma V^k(x')\}, \forall x \in X \quad (21)$$

where $\pi^k(x)$ is the current policy at k -step iteration, and $V^k(x)$ is the state value to be solved under policy $\pi^k(x)$. In the third step, PIM computes an improved policy by minimizing (22):

$$\pi^{k+1}(x) = \arg \min_{\pi} \left\{ \mathbb{E}_{\hat{\xi}_k} \{l(x', \pi(x)) + \gamma V^k(x')\} \right\} \quad (22)$$

where $\pi^{k+1}(x)$ is the new policy. The use of estimated $\hat{\xi}_k$ naturally embeds both analytical model and state-action data into RL, which is able to improve the accuracy of the additive stochastic uncertainty ξ and x' and achieve high convergence speed. The mixed RL algorithm is summarized in Algorithm 1.

Algorithm 1 Mixed RL algorithm

Initialize IBE parameters $\hat{\mu}_0 = \mu_M$ and $\hat{\mathcal{K}}_0 = \mathcal{K}_M$
Initialize state $x_0 \in \mathcal{X}$, $k = 0$
repeat
 update distribution of $\hat{\xi}_k$ and mixed model with ξ_k^D by IBE (20)
 PEV with mixed model:
 $V^k(x) = \mathbb{E}_{\hat{\xi}_k} \{l(x', \pi^k(x)) + \gamma V^k(x')\}$
 PIM with mixed model:
 $\pi^{k+1}(x) = \arg \min_{\pi} \left\{ \mathbb{E}_{\hat{\xi}_k} \{l(x', \pi(x)) + \gamma V^k(x')\} \right\}$
 $k = k + 1$
until $|V^{k+1} - V^k| \leq \epsilon$ and $|\pi^{k+1} - \pi^k| \leq \epsilon$

B. Recursive Stability and Convergence Under Fixed $\hat{\xi}$

In this section, we prove the recursive stability and convergence under fixed additive uncertainty $\hat{\xi}$.

1) *Recursive stability*: Recursive stability means π_k can stabilize the plant so long as π_{k-1} can. We call the closed-loop stochastic system is stable in probability, if for any $\epsilon > 0$, the following equality holds,

$$\lim_{x_0 \rightarrow 0} P \left(\sup_{i \geq 0} \|x_i\| \geq \epsilon \right) = 0 \quad (23)$$

Lemma 1 (Lyapunov stability criterion [20]): If there exists a positive definite Lyapunov sequence $\{g(x_t)\}$ on $N \times \mathbb{R}^n$, which satisfies

$$\mathbb{E} \{g(x_{t+1})\} \leq g(x_t) \quad (24)$$

then the stochastic system is stable in probability, where $g(\cdot)$ is a continuous function, and $g(x_0) < \infty$.

Next, we prove the recursive stability criterion for mixed RL algorithm under fixed $\hat{\xi}$ using Lemma 1.

Theorem 1 (Recursive stability theorem): For any step k in mixed RL, π^{k+1} is stable in probability if π^k is stable in probability and the discount factor γ is selected appropriately under the mixed model.

Proof: Since u^{k+1} is optimal for “weak” Bellman equation, and u^k is non-optimal for $k+1$ step value, we have:

$$\mathbb{E}_{\hat{\xi}} \{l(x', u^{k+1}) + \gamma V^k(x')\} \leq V^k(x) \quad (25)$$

where x' is the next state with (x, u^{k+1}) under the mixed model, and x'_{u^k} is the next state with (x, u^k) . Therefore,

$$V^k(x) \geq \mathbb{E}_{\hat{\xi}} \{V^k(x') + l(x', u^{k+1}) - (1 - \gamma)V^k(x')\} \quad (26)$$

Since π^k is stable in probability, $V^k(x)$ is bounded, thus, $V^k(x')$ is bounded. Considering the fact that $l(\cdot, \cdot)$ and $V^k(\cdot)$ is positive definite function, $1 - \frac{l(x', u^{k+1})}{V^k(x')} < 1$ holds, except for $(x', u^{k+1}) = (0, 0)$ which is stable in probability naturally.

We choose a proper γ to satisfy:

$$\begin{aligned} l(x', u^{k+1}) - (1 - \gamma)V^k(x') &\geq 0 \\ 1 - \frac{l(x', u^{k+1})}{V^k(x')} &\leq \gamma < 1 \end{aligned} \quad (27)$$

Therefore, $V^k(x)$ is monotonically decreasing w.r.t time t with approximate γ , i.e.,

$$V^k(x) \geq \mathbb{E}_{\hat{\xi}} \{V^k(x')\} \quad (28)$$

In short, π^{k+1} is stable in probability. ■

2) *Convergence of mixed RL:* The convergence property describes whether the generated policy, π^k , can converge to the optimum π^* under the mixed RL. Here, we prove the convergence of mixed RL algorithm under fixed $\hat{\xi}$.

Theorem 2 (State value decreasing theorem): For any $\forall x \in \mathcal{X}$ under the additive stochastic uncertainty $\hat{\xi}$, $V^k(x)$ is monotonically decreasing with respect to k , i.e.,

$$V^k(x) \geq V^{k+1}(x), \forall x \in \mathcal{X} \quad (29)$$

Proof: The key is to examine (except for $(0, 0)$)

$$V^k = V_0^{k+1} \geq V_1^{k+1} \geq V_2^{k+1} \geq \dots \geq V_\infty^{k+1} = V^{k+1} \quad (30)$$

At each RL iteration, we initialize $k+1$ step value function by $V_0^{k+1}(x) = V^k(x)$. The first PEV iteration for π^k is

$$\begin{aligned} V_1^{k+1}(x) &= \mathbb{E}_{\hat{\xi}} \{l(x', u^{k+1}) + \gamma V_0^{k+1}(x')\} \\ &= \mathbb{E}_{\hat{\xi}} \{l(x', u^{k+1}) + \gamma V^k(x')\} \end{aligned} \quad (31)$$

With respect to (25), we know

$$V^k(x) \geq V_1^{k+1}(x), \forall x \in X \quad (32)$$

For following PEV iterations, we need to reuse the inequality (32):

$$\begin{aligned} V_1^{k+1}(x) &= \mathbb{E}_{\hat{\xi}} \{l(x', u^{k+1}) + \gamma V^k(f(x, u^{k+1}))\} \\ &\geq \mathbb{E}_{\hat{\xi}} \{l(x', u^{k+1}) + \gamma V_1^{k+1}(f(x, u^{k+1}))\} \\ &= V_2^{k+1}(x) \end{aligned} \quad (33)$$

Similarly, $V_2^{k+1}(x) \geq V_3^{k+1}(x) \geq \dots \geq V_\infty^{k+1}(x)$. Therefore, $\{V_j^{k+1}(x), j = 0, 1, 2, \dots\}$ is a monotonically decreasing sequence and bounded by 0 for $V_j^{k+1}(x) \geq 0$ always holds. Finally, $V_j^{k+1}(x)$ will converge

$$\lim_{j \rightarrow \infty} V_j^{k+1}(x) = V_\infty^{k+1}(x) = V^{k+1}(x) \quad (34)$$

So we have $V^k(x) \geq V^{k+1}(x)$. ■

C. Recursive Stability and Convergence Under Varying $\hat{\xi}_k$

In this section, we discuss the recursive stability and convergence under varying additive uncertainty $\hat{\xi}$, and propose the sufficient condition by designing an upper bound for the differences between $\hat{\xi}_k$ and $\hat{\xi}_{k+1}$.

Under $\hat{\xi}_k$, the self-consistency condition is

$$V^k(x) = \mathbb{E}_{\hat{\xi}_k} \{l(x', u^k) + \gamma V^k(x')\} \quad (35)$$

Since u_{k+1} is the optimal action with respect to $V^k(x)$ of in the k -th iteration, we have

$$\mathbb{E}_{\hat{\xi}_k} \{l(x', u^{k+1}) + \gamma V^k(x')\} \leq V^k(x) \quad (36)$$

which is the key inequality in the proof in section IV-B.

However, when $\hat{\xi}$ is updated from $\hat{\xi}_k$ to $\hat{\xi}_{k+1}$, the variation of $\hat{\xi}$ should be bounded in the interest of stability and convergence. Here, we give the sufficient condition of recursive stability and convergence under varying $\hat{\xi}$, that is, the maximum variation condition (MVC) of the additive stochastic uncertainty (38).

Define $h(x, u^k, \xi)$ as the expected cumulative cost under the additive stochastic uncertainty ξ ,

$$h(x, u, \xi) = \mathbb{E}_{\xi} \{l(x', u) + \gamma V^k(x')\} \quad (37)$$

Theorem 3 (Sufficient condition for recursive stability and convergence): For any step k in mixed RL, π^{k+1} is recursive stable and $V^k(x)$ is monotonically decreasing with respect to k , if the following MVC is satisfied

$$h(x, u^{k+1}, \hat{\xi}_{k+1}) - h(x, u^{k+1}, \hat{\xi}_k) \leq e_k(x) \quad (38)$$

where $e_k(x)$ is the decrease of cumulative cost after PIM,

$$e_k(x) = h(x, u^k, \hat{\xi}_k) - h(x, u^{k+1}, \hat{\xi}_k) \geq 0 \quad (39)$$

The MVC requires that the change of $\hat{\xi}_k$ have less impact on the cumulative cost calculation than PIM in the last iteration.

Proof: Since $h(x, u^{k+1}, \hat{\xi}_k) \geq 0$, when MVC is satisfied, $h(x, u^{k+1}, \hat{\xi}_{k+1}) \leq h(x, u^k, \hat{\xi}_k)$, thus, we have

$$\begin{aligned} &\mathbb{E}_{\hat{\xi}_{k+1}} \{l(x', u^{k+1}) + \gamma V^k(x')\} \\ &\leq \mathbb{E}_{\hat{\xi}_k} \{l(x', u^k) + \gamma V^k(x')\} = V^k(x) \end{aligned} \quad (40)$$

Subsequently, recursive stability theorem (41) and state value decreasing theorem (42) under varying $\hat{\xi}$ can be proved similarly to section IV-B

$$V^k(x) \geq \mathbb{E}_{\hat{\xi}_{k+1}} \{V^k(x')\} \quad (41)$$

$$V^k(x) \geq V^{k+1}(x) \quad (42)$$

Next, we first present Lemma 2 that will be used for the convergence analysis of IBE, then we prove the MVC is satisfied with probability one.

Lemma 2 (Convergence criterion of Bayesian estimation [21]): In Bayesian estimation, if the empirical data \mathcal{D} and the parameter's prior distribution obey Gauss distribution and the covariance matrix of prior distribution is full rank, then

the estimation result $\hat{\mu}$ and $\hat{\mathcal{K}}$ will converge to the sample's mean and covariance asymptotically.

Theorem 4 (MVC is satisfied with probability one criteria): The MVC is satisfied with probability one after sufficient iterations, with the assumption that the IBE converges faster than PIM and PEV.

Proof: Using Kolmogorov strong law of large numbers [22], we have

$$\lim_{k \rightarrow \infty} P \left\{ \frac{1}{k} \sum_{j=1}^k \xi_j^D = \mu \right\} = 1$$

$$\lim_{k \rightarrow \infty} P \left\{ \frac{1}{k} \sum_{j=1}^k (\xi_j^D - \mu) (\xi_j^D - \mu)^T = \mathcal{K} \right\} = 1 \quad (43)$$

where ε and δ are arbitrary small positive constants, μ and \mathcal{K} are the true mean and covariance of ξ . Thus, using Lemma 2 and (43), we know that, when $k \rightarrow \infty$, $\hat{\mu}_k$ and $\hat{\mathcal{K}}_k$ converges to μ and \mathcal{K} in probability one [21], i.e.,

$$\hat{\mu}_k \xrightarrow{1} \mu, \quad \hat{\mathcal{K}}_k \xrightarrow{1} \mathcal{K} \quad (44)$$

Since $\hat{\mu}_k \xrightarrow{1} \hat{\mu}_{k+1} \xrightarrow{1} \mu$, $\hat{\mathcal{K}}_k \xrightarrow{1} \hat{\mathcal{K}}_{k+1} \xrightarrow{1} \mathcal{K}$, and both $p(\hat{\xi}_k)$ and $p(\hat{\xi}_{k+1})$ obey Gaussian distribution, the KL-divergence between $p(\hat{\xi}_k)$ and $p(\hat{\xi}_{k+1})$ converge to 0 with probability one [23], i.e.,

$$\lim_{k \rightarrow \infty} P \left\{ D_{KL} \left(p(\hat{\xi}_k), p(\hat{\xi}_{k+1}) \right) = 0 \right\} = 1$$

$$\Leftrightarrow \mathbb{E} \left\{ p(\hat{\xi}_k) \log \frac{p(\hat{\xi}_k)}{p(\hat{\xi}_{k+1})} \right\} \xrightarrow{1} 0 \Leftrightarrow \frac{p(\hat{\xi}_k)}{p(\hat{\xi}_{k+1})} \xrightarrow{1} 1 \quad (45)$$

Thus, we have

$$\int \left(p(\hat{\xi}_k) - p(\hat{\xi}_{k+1}) \right) \{ l(x', u) + \gamma V^k(x') \} d\xi \xrightarrow{1} 0$$

$$h(x, u, \hat{\xi}_{k+1}) - h(x, u, \hat{\xi}_k) \xrightarrow{1} 0 \quad (46)$$

Since $e_k(x) \geq 0$, when $k \rightarrow \infty$, the MVC (38) holds with probability one, i.e.,

$$\lim_{k \rightarrow \infty} P \left\{ h(x, u^{k+1}, \hat{\xi}_{k+1}) - h(x, u^{k+1}, \hat{\xi}_k) \leq e_k(x) \right\} = 1 \quad (47)$$

In general, MVC indicates that the excessive difference between $\hat{\xi}_{k+1}$ and $\hat{\xi}_k$ should be avoided. In mixed RL, we update the distribution of the additive stochastic uncertainty by Bayesian estimation. As shown in Fig. 3, if a single data batch has large deviation from the total data, the Bayesian estimator can reduce the deviation between the posterior distribution and the total data distribution by introducing appropriate prior distribution of parameters.

V. MIXED RL WITH PARAMETERIZED FUNCTIONS

For large state spaces, both value function and policy are parameterized in mixed RL, as shown in (48). The parameterized value function with known parameter w is called the

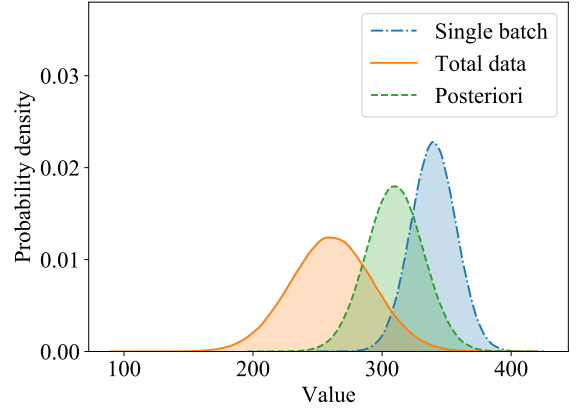


Fig. 3. IBE is effective to prevent excessive difference between $\hat{\xi}_{k+1}$ and $\hat{\xi}_k$ due to the use of appropriate prior distribution of parameters.

“critic”, and the parameterized policy with known parameter θ is called the “actor” [24].

$$V(x) \cong V(x; w)$$

$$u \cong \pi(x; \theta) \quad (48)$$

The parameterized critic is to minimize the average square error (49) in PEV, i.e.,

$$J_{\text{critic}} = \mathbb{E}_{\xi} \left\{ \frac{1}{2} \left(l(x', u_{\theta}) + \gamma V^k(x'; w) - V^k(x; w) \right)^2 \right\} \quad (49)$$

The semi-gradient of the critic is

$$\frac{\partial J_{\text{critic}}}{\partial w} = \int p(x') (V^k(x; w) - V_{\text{target}}) \frac{\partial V^k(x; w)}{\partial w} dx' \quad (50)$$

where $V_{\text{target}} = l(x', u_{\theta}) + \gamma V^k(x')$ and $x' = f(x, u_{\theta}) + \hat{\xi}$.

The parameterized actor is to minimize the “weak” Bellman condition, i.e., to minimize the following objective function,

$$J_{\text{Actor}} = \mathbb{E}_{\xi} \{ l(x', u_{\theta}) + \gamma V^k(x') \}$$

$$= \int [l(x', u_{\theta}) + \gamma V^k(x')] p(x'; u_{\theta}) dx' \quad (51)$$

$$p(x'; u_{\theta}) \sim N \left(f(x, u_{\theta}) + \hat{\mu}, \hat{\mathcal{K}} \right)$$

where $\hat{\mu}$ and $\hat{\mathcal{K}}$ are the mean and covariance of $\hat{\xi}$. The gradient of J_{Actor} is calculated as follows,

$$\frac{\partial J_{\text{Actor}}}{\partial \theta} = \int \left\{ [l(x', u_{\theta}) + \gamma V^k(x')] \frac{\partial p(x'; u_{\theta})}{\partial \theta} + \frac{\partial l(x', u_{\theta})}{\partial \theta} p(x'; u_{\theta}) \right\} dx' \quad (52)$$

In essence, the parameterized method is called generalized policy iteration (GPI). Different from the traditional policy iteration, PEV and PIM each has only one step in GPI, which greatly improves the computational efficiency when RL is combined with neural network.

Since in each GPI cycle, the gradient descent of PIM is only carried out once, the maximum variation condition (MVC) may not be satisfied. We propose a Adaptive GPI (AGPI)

method to solve this problem. In every iteration, we check whether the PIM results satisfy MVC. If not, the algorithm will continue the gradient descent steps in PIM until the MVC is satisfied or when the maximum internal circulation step is reached. Subsequently, the mixed RL algorithm with parameterized Adaptive GPI (AGPI) is summarized in Algorithm 2.

Algorithm 2 Mixed RL with parameterized value and policy

Initialize IBE parameters $\hat{\mu}_0 = \mu_M$ and $\hat{K}_0 = K_M$
Initialize network weights θ_0 and w_0 , choose appropriate learning rates α and β , $k = 0$
repeat
 update distribution of $\hat{\xi}_k$ and mixed model with ξ_k^D by IBE (20)
 update Critic with $\hat{\xi}_k$:
 $w^{k+1} = w^k - \alpha \frac{\partial J_{Critic}}{\partial w}$
 update Actor with $\hat{\xi}_k$:
 $\theta^{k+1} = \theta^k - \beta \frac{\partial J_{Actor}}{\partial \theta}$
 $j = 0$,
 repeat
 update policy net, $j = j + 1$
 until MVC (38) is satisfied or $j = j_{max}$
 $k = k + 1$
until $|V^{k+1} - V^k| \leq \epsilon$ and $|\pi^{k+1} - \pi^k| \leq \epsilon$

VI. NUMERICAL EXPERIMENTS

We consider a typical optimal control problem of stochastic non-affine nonlinear systems, i.e., the combined lateral and longitudinal control of an automated vehicle with stochastic disturbance (i.e., the influence of small road slope and road bumps). The vehicle is subjected to random longitudinal interference force F_{dis} in the tracking process and the vehicle dynamics is shown in (53) [25].

$$\dot{x} = \begin{bmatrix} \frac{F_{yf} \cos \delta + F_{yr}}{a F_{yf} \cos \delta - b F_{yr}} - v_x r \\ a_x + v_y r - \frac{I_z F_{yf} \sin \delta}{m} + \frac{F_{dis}}{m} \\ v_x \sin \phi + v_y \cos \phi \end{bmatrix} \quad (53)$$

where the state $x = [v_y \ r \ v_x \ \phi \ y]^T$, v_y is the lateral velocity, r is yaw rate, v_x is the difference between longitudinal velocity and desired velocity, ϕ is the yaw angle, and y is the distance between vehicle's centroid and the target trajectory. For the control input $u = [\delta \ a_x]^T$, where δ is the front wheel angle and a_x is the longitudinal acceleration. The F_{yf} and F_{yr} are the lateral tire forces of the front and rear tires respectively, which are calculated by the Fiala tire model [26]. In the tire model, the tire-road friction coefficient μ is set as 1.0. The front wheel cornering stiffness and rear wheel cornering stiffness are set as 88000 N/rad and 94000 N/rad respectively. The mass m is set as 1500 kg, the a and b are the distances from centroid to front axle and rear axle, and set as 1.14 m and 1.40 m respectively. The polar moment of inertia I_z at centroid is set as 2420 N/rad. The random longitudinal

interference force $F_{dis} \sim N(261, 32)$ and the desired velocity is set as 12 m/s [27].

For comparison purpose, a double-lane change task was simulated respectively with three different RL algorithms. The task is to track the desired trajectory in the lateral direction while maintaining the desired longitudinal velocity under the longitudinal interference F_{dis} . Hence, the optimal control problem with discretized stochastic system equation is given by

$$\min_u \sum_{t=0}^{\infty} \gamma^t \left(45(v_x - 12)^2 + 60y^2 + u^\top \begin{bmatrix} 800 & 0 \\ 0 & 1 \end{bmatrix} u \right) dt$$

$$s.t. \quad x_{t+1} = f(x_t, u_t) + \xi_t, \quad \xi_t = F_{dis}T/m \quad (54)$$

where $\gamma = 0.99$ is the discounting factor, $f(\cdot, \cdot)$ is the deterministic part of the discretized system equation of (47), ξ_t is the additive stochastic uncertainty and the simulation time interval T is set as 1/200(s). In this simulated task, we compared the performance of mixed RL with both model-driven RL and data-driven RL. The data-driven RL computes the control policy only by using the state-action data with a typical data-driven algorithm (i.e., DDPG) [3]. The model-driven RL computes the policy by GPI [28] directly using the given empirical model

$$x_{t+1} = f(x_t, u_t) + \xi_t^M, \quad \xi_t^M \sim N(\mu_M, K_M) \quad (55)$$

where the prior distribution is set as $\mu_M = [150T/m, 0, 0, 0, 0]^T$ and K_M is a diagonal matrix, whose diagonal elements are $[(4T/m)^2, 10^{-20}, 10^{-20}, 10^{-20}, 10^{-20}]$.

The convergence performance of these three algorithms are compared in Fig. 4. The mixed RL and model-driven RL can converge in 1×10^4 iterations, while the data-driven RL needs 4×10^4 iterations to converge under the same hyper-parameter.

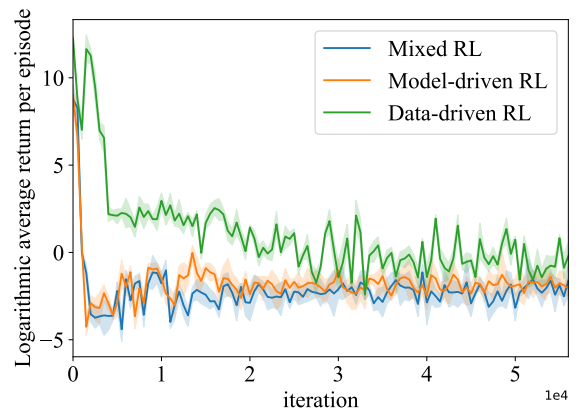


Fig. 4. Convergence rate comparison between mixed RL, model-driven RL, and data-driven RL.

For control performance, we test the policies calculated by three methods in the double lane change task. As shown in Fig. 5, all three policies stably tracked the target trajectory, but with different control error. In fact, as shown in Fig. 6, the mixed RL has the minimum longitudinal speed error, since

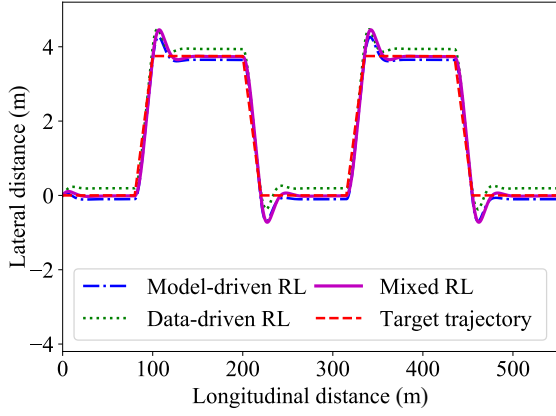


Fig. 5. Tracking performance comparison.

it enables the vehicle to decelerate rapidly at sharp turns and adjust back appropriately after passing the turns. In contrast, due to the model error, the model-driven RL has higher speed error and its deceleration when making turns is insufficient. Due to the slow convergence, the data-driven RL generates a poor solution and has the largest speed error. The mixed RL

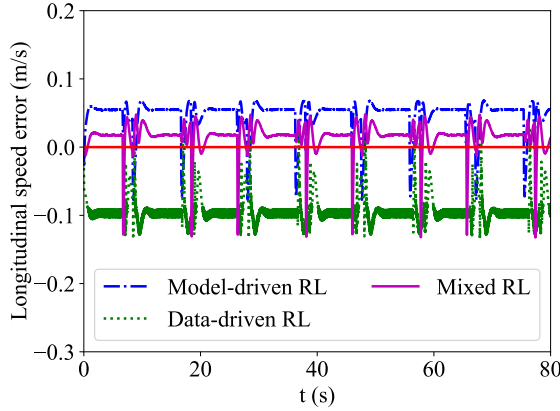


Fig. 6. Longitudinal speed error.

also outperforms the other two benchmark methods in terms of the lateral position error. As shown in Fig. 7, the mixed RL has the minimum steady-state lateral position error, while data-driven RL has the largest lateral position error and frequent speed fluctuation.

The mean absolute errors of three methods are compared in Table I. The longitudinal speed error of mixed RL is 77.41% less, and the lateral position error is 33.77% less than the data-driven RL. Besides, the longitudinal speed error of mixed RL is 58.82% less, and the lateral position error is 15.64% less than the model-driven RL.

In summary, mixed RL exhibits the fastest convergence speed during the training process and the greatest control performance in double lane change task. The model-driven RL has similar convergence speed as the mixed RL, but has higher control error due to the model error. The data-driven RL has the slowest convergence rate and the largest control

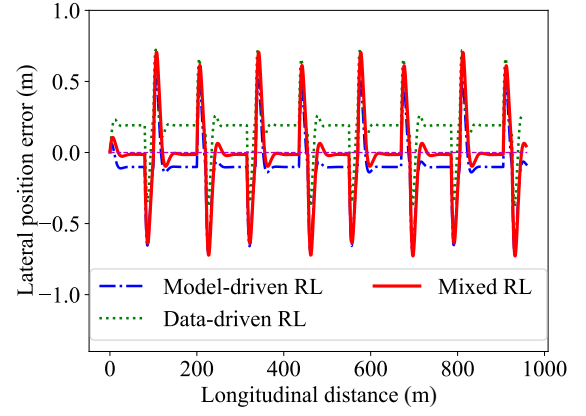


Fig. 7. Lateral position error.

TABLE I
PERFORMANCE COMPARISON OF THREE METHODS

Method	Position error [m]	Speed error [m/s]
Mixed RL	0.151	0.021
Data-driven RL	0.228	0.093
Model-driven RL	0.179	0.051

error, due to the difficulties in finding the optimal policy only by state-action data.

VII. CONCLUSION

This paper proposes a mixed reinforcement learning approach with better performances on convergence speed and policy accuracy for non-linear systems with additive Gaussian uncertainty. The mixed RL utilizes an iterative Bayesian estimator to accurately model the environmental dynamics by integrating the designer's knowledge with the measured state transition data. The convergence and recursive stability of learned policy were proved via Bellman's principle of optimality and Lyapunov analysis. It is observed that mixed RL achieves faster convergence rate and more stable training process than the data-driven counterpart. Meanwhile, mixed RL has lower policy error than model-driven counterpart since the environmental model is refined iteratively by Bayesian estimation. The benefits of mixed RL are demonstrated by a double-lane change task with an automated vehicle. The potential of mixed RL in more general environmental dynamics and non-Gauss uncertainties will be investigated in the future.

REFERENCES

- [1] E. Gibney, "Google ai algorithm masters ancient game of go," *Nature News*, vol. 529, no. 7587, p. 445, 2016.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [4] J. Duan, Y. Guan, Y. Ren, S. E. Li, and B. Cheng, "Addressing value estimation errors in reinforcement learning with a state-action return distribution function," *arXiv preprint arXiv:2001.02811*, 2020.

- [5] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624–2632, 2014.
- [6] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.
- [7] J. Duan, S. E. Li, Z. Liu, M. Bujarbaruah, and B. Cheng, "Generalized policy iteration for optimal control in continuous time," *arXiv preprint arXiv:1909.05402*, 2019.
- [8] J. Duan, Z. Liu, S. E. Li, Q. Sun, Z. Jia, and B. Cheng, "Deep adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints," *arXiv preprint arXiv:1911.11397*, 2019.
- [9] F. L. Lewis and D. Liu, *Reinforcement learning and approximate dynamic programming for feedback control*. John Wiley & Sons, 2013, vol. 17.
- [10] D. P. Bertsekas, "Approximate policy iteration: a survey and some new methods," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 310–335, 2011.
- [11] Y. Guan, S. E. Li, J. Duan, J. Li, Y. Ren, and B. Cheng, "Direct and indirect reinforcement learning," *arXiv preprint arXiv:1912.10600*, 2019.
- [12] S. Bansal, R. Calandra, K. Chua, S. Levine, and C. Tomlin, "Mbmf: Model-based priors for model-free reinforcement learning," *arXiv preprint arXiv:1709.03153*, 2017.
- [13] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.
- [14] V. Feinberg, A. Wan, I. Stoica, M. Jordan, J. Gonzalez, and S. Levine, "Model-based value expansion for efficient model-free reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018.
- [15] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014, pp. 1071–1079.
- [16] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.
- [17] R. Lioutikov, A. Paraschos, J. Peters, and G. Neumann, "Sample-based informationl-theoretic stochastic optimal control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3896–3902.
- [18] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel, "Model-ensemble trust-region policy optimization," *arXiv preprint arXiv:1802.10592*, 2018.
- [19] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," in *Advances in Neural Information Processing Systems*, 2018, pp. 8224–8234.
- [20] H. Deng, M. Krstic, and R. J. Williams, "Stabilization of stochastic nonlinear systems driven by noise of unknown covariance," *IEEE Transactions on automatic control*, vol. 46, no. 8, pp. 1237–1253, 2001.
- [21] P. Diaconis and D. Freedman, "On the consistency of bayes estimates," *The Annals of Statistics*, pp. 1–26, 1986.
- [22] K. Chung, "The strong law of large numbers," *Selected Works of Kai Lai Chung*, pp. 145–156, 2008.
- [23] J. M. H. Lobato, "Expectation propagation for approximate bayesian inference," 2007.
- [24] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.
- [25] S. E. Li, H. Chen, R. Li, Z. Liu, Z. Wang, and Z. Xin, "Predictive lateral control to stabilise highly automated vehicles at tire-road friction limits," *Vehicle System Dynamics*, pp. 1–19, 2020.
- [26] Y.-H. J. Hsu, S. M. Laws, and J. C. Gerdes, "Estimation of tire slip angle and friction limits using steering torque," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 896–907, 2009.
- [27] S. Xu, S. E. Li, B. Cheng, and K. Li, "Instantaneous feedback control for a fuel-prioritized vehicle cruising system on highways with a varying slope," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1210–1220, 2016.
- [28] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.



Yao Mu received the B.S. degree in vehicle engineering from Harbin institute of technology, China, in 2018. He is currently a Master student at the State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University. His active research interests include intelligent vehicles, automatic driving technology, optimal control and reinforcement learning algorithms.



Shengbo Eben Li received the M.S. and Ph.D. degrees from Tsinghua University in 2006 and 2009. Before joining Tsinghua University, he has worked at Stanford University, University of Michigan, and UC Berkeley. He is now leading Intelligent Driving Lab (iDLab) at Tsinghua University. His active research interests include intelligent vehicles and driver assistance, reinforcement learning and optimal control, distributed control and estimation, etc. He is the author of over 100 peer-reviewed journal/conference papers, and the co-inventor of over 30

patents. Dr. Li was the recipient of Best Paper Award in 2014 IEEE ITS, Best Paper Award in 14th Asian ITS, National Award for Technological Invention of China (2013), Excellent Young Scholar of NSF China (2016), Young Professorship of Changjiang Scholar Program (2016), Tsinghua University Excellent Professorship Award (2017), National Award for Progress in Science and Technology of China (2018), Distinguished Young Scholar of Beijing NSF (2018), etc. He also serves as Board of Governor of IEEE ITS Society, AEs of IEEE ITSM, IEEE Trans ITS, etc.



Chang Liu (S'15, M'17) is a Postdoctoral Associate in the Sibley School of Mechanical and Aerospace Engineering at Cornell University, where he works on the decentralized perception and planning of multi-agent systems. He received the B.S. degrees in Electrical Engineering and in Applied Mathematics (double major) in 2011 from Peking University, China. He received the M.S. degrees in Mechanical Engineering and in Computer Science in 2014 and 2016 from the University of California, Berkeley. He received his Ph.D. in Mechanical Engineering from

the University of California, Berkeley in 2017. His research interests include planning and decision making of robots, multi-agent systems, state estimation and prediction, computer vision, and human-robot collaboration.



Qi Sun received his Ph.D. degree in Automotive Engineering from Ecole Centrale de Lille, France, in 2017. He did scientific research and completed his Ph.D. dissertation in CRISTAL Research Center at Ecole Centrale de Lille, France, between 2013 and 2016. He is currently a Postdoctor at the State Key Laboratory of Automotive Safety and Energy and at the Department of Automotive Engineering, Tsinghua University, Beijing, China. His active research interests include intelligent vehicles, automatic driving technology, distributed control and

optimal control.



Bo Cheng received the B.S. and M.S. degrees in automotive engineering from Tsinghua University, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree in mechanical engineering from the University of Tokyo, Tokyo, Japan, in 1998. He is currently a Professor with School of Vehicle and Mobility, Tsinghua University, and the Dean of Tsinghua University-Suzhou Automotive Research Institute. He is the author of more than 100 peer-reviewed journal/conference papers and the co-inventor of 40 patents. His active research interests

include autonomous vehicles, driver-assistance systems, active safety, and vehicular ergonomics, among others. Dr. Cheng is also the Chairman of the Academic Board of SAE-Beijing, a member of the Council of the Chinese Ergonomics Society, and a Committee Member of National 863 Plan, among others.



Baiyu Peng Baiyu Peng received the B.S. degree in vehicle engineering from Tsinghua University, Beijing, China, in 2019, where he is currently pursuing the Master degree in vehicle engineering. He is currently with the State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University. His current research interests include decision-making and control of automated vehicles, and reinforcement learning algorithms.



Bingbing Nie is an Associate Professor at School of Vehicle and Mobility, Tsinghua University, China. She received her BSc (2007) from Tsinghua University, MSc (2009) from RWTH-Aachen, Germany and PhD (2013) from Tsinghua University. Prior to joining Tsinghua University in 2016, she worked as a Visiting Scholar in General Motors R&D (2012-2013), and as a Research Associate at University of Virginia (2013-2016). Her research areas include human-vehicle interaction, vehicle safety, applied biomechanics. She has authored/co-authored more than 40 technical papers and 3 patents. She has also served as Session Organizer of Pedestrian and Cyclist Safety of SAE World Congress, IRCOBI Scientific Review Committee Member, AAAM Scientific Program Committee Member and China-Sweden CTS Scientific Committee Member.