Exploring State Transition Uncertainty in Variational Reinforcement Learning

Jen-Tzung Chien
Department of Elect. & Comp. Eng.
National Chiao Tung University
Hsinchu, Taiwan

Wei-Lin Liao
Department of Elect. & Comp. Eng.
National Chiao Tung University
Hsinchu, Taiwan

Issam El Naqa
Department of Radiation Oncology
University of Michigan
Ann Arbor, USA

***Abstract*—Model-free agent in reinforcement learning (RL) generally performs well but inefficient in training process with sparse data. A practical solution is to incorporate a model-based module in model-free agent. State transition can be learned to make desirable prediction of next state based on current state and action at each time step. This paper presents a new learning representation for variational RL by introducing the so-called transition uncertainty critic based on the variational encoder-decoder network where the uncertainty of structured state transition is encoded in a model-based agent. In particular, an action-gating mechanism is carried out to learn and decode the trajectory of actions and state transitions in latent variable space. The transition uncertainty maximizing exploration (TUME) is performed according to the entropy search by using the intrinsic reward based on the uncertainty measure corresponding to different states and actions. A dedicate latent variable model with a penalty using the bias of state-action value is developed. Experiments on Cart Pole and dialogue system show that the proposed TUME considerably performs better than the other exploration methods for reinforcement learning.**

***Index Terms*—machine learning, reward optimization**

I. Introduction

Reinforcement learning (RL) aims to develop an agent to sequentially decide an action a_t based on current state s_t which contains the observation about environment at each time step t. A reward r_t and next state s_{t+1} are then received from environment. Such an agent is basically learned and updated by maximizing the future reward at each time. According to the updating procedure and the way of making decisions, the agents in RL can be model-based or model-free. There are policy-based, value-based and actor-critic agents in model-free RL. Model-based agents learn the state transition from environment under a Markov decision process (MDP) while model-free methods learn directly from reward without MDP. An interesting issue in RL is to integrate the tradeoff between model-based and model-free methods. When learning the state transitions of environment, it is crucial to represent the relation among current state s_t, action a_t and next state s_{t+1} [1], [2], [3], [4]. The representation in latent variable space is attractive and powerful because the environment dynamics can be effectively learned by using generative model based on variational autoencoder (VAE) [5]. This paper proposes a new variational neural network to implement a transition uncertainty critic (TUC) in a variational RL procedure where the environment is explored via entropy search with a latent variable model driven by the action-dependent Markov chains. An action-gating scheme is incorporated to carry out this TUC which does not only characterize the state transitions but also the state-action values in latent variable space. In addition to the maximum entropy reduction, this variational RL is further improved by balancing between exploitation and exploration through bias minimization in state-action value.

II. Background Survey

In general, the actor-critic agent in RL consists of an actor and a critic which are updated by policy gradients given by the policy network $\pi(a_t|s_t)$ with parameter θ_π and the critic network which are used to learn the state transition. Here, a variational recurrent neural network [6], inspired by VAE, is developed to represent the latent structure of state transitions. The idea of entropy search, driven by Bayesian optimization [7], is implemented to search for the optimal decision based on the model-based critic in latent space. Action-dependent Markov chains are geared by an action-gating mechanism. A variational bound of total reward function is maximized.

A. Variational Autoencoder

Variational autoencoder is known as a popular latent variable model for data generation where the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ of hidden variable $\mathbf{z}$ is learned to reconstruct original data $\mathbf{x}$. This distribution characterizes the randomness of hidden units which provides a vehicle to reconstruct different realizations of output signals $\widehat{\mathbf{x}}$ rather than a point estimate of outputs in traditional auto-encoder. The encoder in VAE aims to represent the latent variable $\mathbf{z}$ using a variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ with parameter ϕ. Latent variables $\mathbf{z}$ are then sampled to reconstruct original signal $\widehat{\mathbf{x}}$ based on the decoder using the likelihood function $p_\theta(\mathbf{x}|\mathbf{z})$ with parameter θ. Variational parameter ϕ and model parameter θ are estimated by maximizing the *variational lower bound* of log likelihood obtained by

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] \quad (1)$$

where D_{KL} denotes the Kullback-Leibler (KL) divergence and the prior density $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is used. Stochastic gradient variational Bayes is implemented for variational learning [8]. This study presents a variational neural network and constructs a transition uncertainty critic where the latent variable $\mathbf{z}$ is encoded from s_t and a_t and used to decode s_{t+1} and measure the state-action value $Q(a_t, s_t)$ for variational RL.

B. Variational Information Maximizing Exploration

It is important to balance the tradeoff between exploration and exploitation in reinforcement learning. In [9], the entropy search based on variational information maximizing exploration (VIME) was proposed to encourage an agent to explore efficiently. VIME maximized the entropy reduction of surrogate function based on the Bayes-by-backprop network [10]. State transition of the environment was learned. Weights in Bayesian neural network (BNN) were characterized. BNN was robust to mode collapse. Entropy reduction was viewed as the intrinsic reward to guide the agent to explore with maximum entropy reduction. The environment dynamics were measured by $p_\theta(s_{t+1}|s_t, a_t)$ with parameter θ. The trajectory of RL until time step t is denoted as $h_t = \{s_1, a_1, s_2, a_2, \dots, s_{t-1}, a_{t-1}, s_t\}$. The entropy reduction due to a new action a_t and next state s_{t+1} is calculated by $\sum_t[\mathbb{H}(\theta|h_t) - \mathbb{H}(\theta|h_t, a_t, s_{t+1})]$ where $\mathbb{H}(\cdot)$ denotes an entropy function and the individual terms in summation are seen as the *information gains*. Such an entropy reduction can be also expressed in terms of KL divergence as

$$\mathbb{E}_{p(s_{t+1}|h_t)}\big[D_{\text{KL}}[p(\theta|h_t, a_t, s_{t+1})\|p(\theta|h_t)]\big]. \quad (2)$$

However, the posterior of training data $\mathbf{x}$ or trajectory h_t or $\{h_t, a_t, s_{t+1}\}$ with hidden variable θ, i.e. $p(\theta|\mathbf{x})$, $p(\theta|h_t)$ or $p(\theta|h_t, a_t, s_{t+1})$, is intractable. Similar to VAE in Eq. (1), an alternative distribution $q(\theta; \phi)$ or variational parameter ϕ is estimated by maximizing the variational lower bound

$$\mathcal{L}[q(\theta;\phi), \mathbf{x}] = \mathbb{E}_{q(\theta;\phi)}[\log p(\mathbf{x}|\theta)] - D_{\text{KL}}[q(\theta;\phi)\|p(\theta)]. \quad (3)$$

The parameter updating for variational distribution $q(\theta;\phi)$ using parameter from ϕ_t to and ϕ_{t+1} is performed by maximizing Eq. (3) and then used to approximate the entropy reduction in Eq. (2). Intrinsic reward is therefore defined by

$$r_t^i = D_{\text{KL}}\big[q(\theta;\phi_{t+1})\|q(\theta;\phi_t)\big] \quad (4)$$

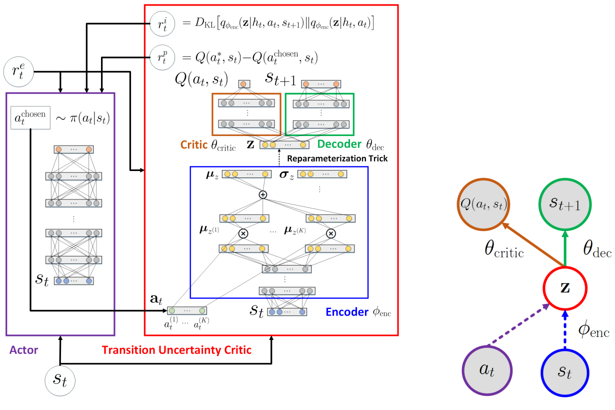
The agent is then run according to total reward r_t containing the intrinsic reward r_t^i given by Eq. (4) and the extrinsic reward r_t^e given by environment, i.e. $r_t = r_t^e + r_t^i$. The exploration based on r_t^i and the exploitation based on r_t^e are implemented.

III. Exploration for Transition Uncertainty

This paper presents a new reinforcement learning which potentially deals with two issues. First, how to learn the latent variables of state transitions caused by different actions, and second, how to utilize the latent information of state transitions to facilitate exploration and exploitation for agent. To tackle these issues, we propose a latent variable model to carry out the action-gating scheme and conduct the uncertainty modeling for state transitions as well as state-action values in variational reinforcement learning for dialogue system.

A. System Architecture & Action-Gating Mechanism

Figure 1(a) illustrates an overview of neural reinforcement learning based on an engine consisting of an actor with a policy network given by parameter θ_π for finding policy $\pi(a_t|s_t)$ and choosing action a_t^{chosen} and a transition uncertainty critic (TUC) with encoder parameter ϕ_{enc}, critic parameter θ_{critic} and decoder parameter θ_{dec} for providing the state-action value $Q(a_t, s_t)$ as well as the state transition from s_t to s_{t+1} in latent space $\mathbf{z}$. Figure 1(b) shows the graphical representation. Action-gating scheme driven by a one-hot action vector $\mathbf{a_t} = [a_t^{(1)}, \dots, a_t^{(K)}]^\top$ is implemented with K discrete actions. The learning procedure for policy network and critic network is driven by a penalized reward function r_t which is composed of environment reward r_t^e, intrinsic reward r_t^i and penalty r_t^p in a form of $r_t = r_t^e + r_t^i - r_t^p$ as detailed in what follows.


r_t^i $= D_{\text{KL}}[q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t, s_{t+1})\|q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t)]$
r_t^p $= Q(a_t^*, s_t) - Q(a_t^{\text{chosen}}, s_t)$
r_t^e
a_t^{chosen} $\sim \pi(a_t|s_t)$
$Q(a_t, s_t)$
s_{t+1}
Critic θ_{critic}
$\mathbf{z}$
Decoder θ_{dec}
Reparameterization Trick
$\boldsymbol{\mu}_z$
$\boldsymbol{\sigma}_z$
$\boldsymbol{\mu}_{z^{(1)}}$
$\boldsymbol{\mu}_{z^{(K)}}$
s_t
$\mathbf{a}_t$
$a_t^{(1)}$ ⋯ $a_t^{(K)}$
Encoder ϕ_{enc}
Actor
Transition Uncertainty Critic
s_t
$Q(a_t, s_t)$
s_{t+1}
θ_{critic}
θ_{dec}
$\mathbf{z}$
ϕ_{enc}
a_t
s_t

(a) (b)

Fig. 1: (a) An actor-TUC agent for variational reinforcement learning. (b) Graphical model for encoder-decoder network.

Basically, the probability that an agent moves into the next state s_{t+1} depends on the action a_t chosen by the agent. Next state s_{t+1} is decided by the current state s_t and the chosen action $\mathbf{a}_t$ or a_t^{chosen}. It is meaningful to build learning representation with the action-dependent MDP where Markov chains corresponding to individual actions are modeled. An action-gating mechanism is merged to handle this issue based on a latent variable model where $\mathbf{z}^{(k)}$ denotes the continuous latent variable for the pair of state s_t and action $a_t^{(k)}$. The overall latent variable $\mathbf{z}$ is obtained from switching over latent variables corresponding to different actions $\mathbf{z} = \sum_{k=1}^K a_t^{(k)}\ \mathbf{z}^{(k)}$ where $a_t^{(k)} \in \{0, 1\}$. In the implementation, the stochastic backpropagation is performed by using the *samples* of latent variables $\mathbf{z}^{(k)}$. The reparameterization trick [5] is applied to find samples based on $\mathbf{z}^{(k)} = \boldsymbol{\mu}_{z^{(k)}} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_{z^{(k)}}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\odot$ denotes the element-based product, $\boldsymbol{\mu}_{z^{(k)}}$ and $\boldsymbol{\sigma}_{z^{(k)}}$ denote the vectors of mean and standard deviation of Gaussian distribution which are calculated in the output nodes of encoder, respectively. The red box in Figure 1 shows how the action-gating mechanism is incorporated in a variational neural network for state transition from s_t to s_{t+1} driven by sampling shown in dashed arrow. The inputs of action $\mathbf{a}_t$ and state s_t are used to predict next state s_{t+1} and the state-action value $Q(a_t, s_t)$ based on an *encoder-decoder critic network*. The action-dependent state transitions are implemented via a latent variable model. In practice, such an action-gating scheme can be also derived by implementing an embedding vector $\mathbf{a}_t$ from continuous or high-dimensional actions.

B. Transition Uncertainty in Latent Space

Exploration in RL was guided according to curiosity [11], [12], [13], information gain or entropy reduction [9]. In Section II-B, VIME learned the state transitions directly over parameter space θ neither considering the latent variable space $\mathbf{z}$ over individual influences from different actions nor producing the state-action value $Q(a_t, s_t)$ for reinforcement learning. This study presents a transition uncertainty critic based on a variational neural network which maps the current state s_t and action a_t to next state s_{t+1} and state-action value $Q(a_t, s_t)$. In this mapping, the latent variable $\mathbf{z}$ is sampled and driven by an action-gating scheme. Different from VIME finding the variational distribution for parameter θ without a critic and without a decoder, we construct an encoder-decoder network based on latent variable $\mathbf{z}$, which is capable of reconstructing or mapping from $\{s_t, a_t\}$ to $\{s_{t+1}, Q(a_t, s_t)\}$. Policy network and critic network are jointly trained. The curiosity-driven intrinsic reward is calculated in latent space $\mathbf{z}$ in a form of

$$\begin{aligned} r_t^i &= D_{\text{KL}}[p(\mathbf{z}|h_t, a_t, s_{t+1}) \| p(\mathbf{z}|h_t, a_t)] \\ &\approx D_{\text{KL}}[q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t, s_{t+1}) \| q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t)] \end{aligned} \tag{5}$$

where the entropy reduction due to intractable posterior distributions from $p(\mathbf{z}|h_t, a_t)$ to $p(\mathbf{z}|h_t, a_t, s_{t+1})$ is approximated by using the tractable variational distributions $q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t)$ and $q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t, s_{t+1})$ based on encoder parameter ϕ_{enc}. Such a curiosity-driven exploration strategy guides an actor to select the actions that lead to the *informative states* to truly reflect surrounding environment. To calculate Eq. (5), we need to estimate the encoder ϕ_{enc} and decoder parameters θ_{dec}. These parameters are updated by maximizing the variational lower bound of log likelihood of new state s_{t+1} as formulated by

$$\begin{aligned} \log\, p(s_{t+1}|h_t, a_t) \geq\ & \mathbb{E}_{q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t)}[\log\, p_{\theta_{\text{dec}}}(s_{t+1}|h_t, a_t, \mathbf{z})] \\ & - D_{\text{KL}}(q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t) \| p(\mathbf{z})). \end{aligned} \tag{6}$$

C. State-Action Value in Latent Space

This study typically estimates a latent variable model which conveys the uncertainty about trajectory h_t or state transition $s_t \to s_{t+1}$ for exploration. The return $R_t = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}$ with $\gamma \in (0, 1]$ can be improved at each time step t. This is because that latent variable $\mathbf{z}$ contains future information for prediction of state value $V(s_t)$ or state-action value $Q(a_t, s_t)$. The estimation of state transition, controlled by actor, is facilitated to decide next state s_{t+1} as well as action a_t. In case of a well-trained TUC, we may choose the *best* action a_t^* which achieves the *highest* state-action value $Q(a_t, s_t)$. At the same time, the actor also selects an action a_t^{chosen} based on the policy network $\pi(a_t|s_t)$, i.e. $a_t^{\text{chosen}} \sim \pi(a_t|s_t)$. The goodness of selection can be reflected by the difference of state-action values between TUC and actor. The smaller the difference, the better the actor is choosing. Accordingly, in addition to maximizing the extrinsic (or environment) reward r_t^e and the implicit reward r_t^i for exploitation and exploration, we further introduce the bias of state-action value as a penalty term

$$r_t^p = Q(a_t^*, s_t) - Q(a_t^{\text{chosen}}, s_t) \geq 0. \tag{7}$$

This penalty is neglected $r_t^p = 0$ when the chosen action is the same as the best action $a_t^{\text{chosen}} = a_t^*$. Attractively, the state-action value $Q(a_t, s_t)$ is provided by TUC via the encoder-decoder network with an encoded latent code $\mathbf{z} = f_{\phi_{\text{enc}}}(a_t, s_t)$. This value can be expressed by a hybrid function $Q(a_t, s_t) = f_{\theta_{\text{critic}}}(f_{\phi_{\text{enc}}}(a_t, s_t))$ using the encoder parameter ϕ_{enc} and critic parameter θ_{critic}. We aim to compensate this bias at each time step t by penalizing the agent with the chosen action a_t^{chosen} which differs from the best action a_t^*. In case of the practical reward, $r_t = r_t^e - r_t^p$, RL is to maximize the total reward

$$\max_{\theta_\pi} R_t = \max_{\theta_\pi} \left(\sum\nolimits_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}^e - \sum\nolimits_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}^p \right) \tag{8}$$

where the first term denotes the extrinsic total reward R_t^e and the second term is always nonnegative. Minimizing the second term encourages the agent to choose the action which is close to that of critic, namely $\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}^p \simeq 0$ or $Q(a_t^*, s_t) \simeq Q(a_t^{\text{chosen}}, s_t)$. Basically, $Q(a_t^*, s_t)$ is determined as the best transition value among different trajectories in different episodes. When $Q(a_t^*, s_t)$ is close to $Q(a^{\text{chosen}}, s_t)$, this RL carries out the behavior of actor which is close to that of critic with the best policy from different trajectories.

Algorithm 1: Learning and exploring in variational RL

```
Initialize parameters $\{\phi_{\text{enc}}, \theta_{\text{dec}}, \theta_\pi, \theta_{\text{critic}}\}$
Assume environment for $\{s_t, a_t, s_{t+1}\}$ is Markovian
for each episode $i = 1, \ldots, M$ do
    for each time step $t = 0, 1, 2, \ldots, T-1$ do
        sample $a_t^{\text{chosen}}$ from policy $\pi(\cdot|s_t)$
        sample $s_{t+1}$ and $r_t^e$ from environment
        sample $a_{t+1}^{\text{chosen}}$ from policy $\pi(\cdot|s_{t+1})$
        compute Gaussian vectors in $q_{\phi_{\text{enc}}}(\mathbf{z}|s_t, a_t^{\text{chosen}})$
        update $\theta_{\text{dec}}$ by using $\nabla_{\theta_{\text{dec}}}\mathcal{L}_{\theta_{\text{dec}}}$
        update $\phi_{\text{enc}}$ by using $\nabla_{\phi_{\text{enc}}}\mathcal{L}_{\phi_{\text{enc}}}$
        compute Gaussian vectors in $q_{\phi_{\text{enc}}}(\mathbf{z}|s_{t+1}, a_{t+1}^{\text{chosen}})$
        compute
          $r_t^i = D_{\text{KL}}[q_{\phi_{\text{enc}}}(\mathbf{z}|s_{t+1}, a_{t+1}^{\text{chosen}}) \| q_{\phi_{\text{enc}}}(\mathbf{z}|s_t, a_t^{\text{chosen}})]$
        compute $r_t^p = Q(a_t^*, s_t) - Q(a_t^{\text{chosen}}, s_t)$
        compute practical reward $r_t = r_t^e + r_t^i - r_t^p$
    end
    compute $R_t^e$ and $R_t = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau}$
    update $\theta_{\text{critic}}$ by using $\nabla_{\theta_{\text{critic}}}\mathcal{L}_{\theta_{\text{critic}}}$
    update $\theta_\pi$ by using $\nabla_{\theta_\pi}\mathcal{L}_{\theta_\pi}$
end
```

D. Variational Reinforcement Learning

Algorithm 1 addresses the variational RL based on the proposed transition uncertainty critic and action-gating mechanism. The policy network and critic network are jointly learned according to four terms in learning objective $\mathcal{L}_{\phi_{\text{enc}}}$, $\mathcal{L}_{\theta_{\text{dec}}}$, $\mathcal{L}_{\theta_{\text{critic}}}$ and $\mathcal{L}_{\theta_\pi}$ which are used to update the parameters encoder, decoder, critic and actor, respectively. In this learning procedure, the uncertainty of state transitions in RL is first explored via a variational encoder-decoder network with parameters

$\{\theta_{\text{dec}}, \phi_{\text{enc}}\}$ at each time step t. We minimize the negative of variational lower bound in Eq. (6) consisting of

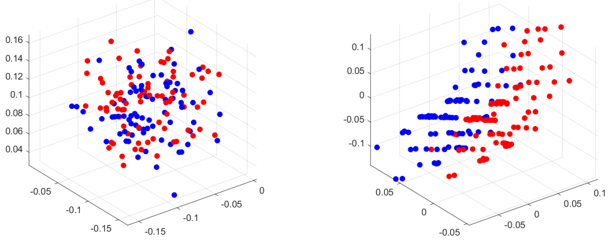
$$\mathcal{L}_{\theta_{\text{dec}}} = -\mathbb{E}_{q_{\phi_{\text{enc}}}(\mathbf{z}|h_t,a_t)}[\log p_{\theta_{\text{dec}}}(s_{t+1}|h_t, a_t, \mathbf{z})] \quad (9)$$

$$\mathcal{L}_{\phi_{\text{enc}}} = D_{\text{KL}}[q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t)||p(\mathbf{z})]. \quad (10)$$

These terms are calculated by using the variational distribution $q_{\phi_{\text{enc}}}(\mathbf{z}|s_t, a_t^{\text{chosen}})$ or $q_{\phi_{\text{enc}}}(\mathbf{z}|h_t, a_t^{\text{chosen}})$ with next state s_{t+1} and extrinsic reward r_t^e sampled from environment, and current action a_t^{chosen} and next action a_{t+1}^{chosen} sampled by policy networks $\pi(\cdot|s_t)$ and $\pi(\cdot|s_{t+1})$ at each state transition at time t, respectively. The intrinsic reward r_t^i and penalty r_t^p are then determined by Eqs. (5) and (7), respectively. At the end of each episode, the critic parameter θ_{critic} and actor parameter θ_π are updated according to the regression error for state-action value $\mathcal{L}_{\theta_{\text{critic}}} = \mathbb{E}_{q_{\phi_{\text{enc}}}(\mathbf{z}|h_t,a_t)}[\|Q(a_t, s_t) - R_t^e\|^2]$ and the policy gradient for actor $\nabla_{\theta_\pi}\mathcal{L}_{\theta_\pi} = -\mathbb{E}_{q_{\phi_{\text{enc}}}(\mathbf{z}|h_t,a_t)}[R_t \nabla_{\theta_\pi} \log\pi(a_t|s_t)]$, respectively. The actor-TUC agent is implemented to fulfill the so-called *transition uncertainty maximizing exploration* for variational RL as illustrated in the experiments.

IV. Experiments

We evaluated the performance of the proposed transition uncertainty maximizing exploration (TUME) for reinforcement learning based on two tasks: Cart Pole and dialogue system. For comparative study, the VIME as addressed in Section II-B and the curiosity maximizing exploration (CME) [13] were implemented as different intrinsic rewards r_t^i. Baseline result with only extrinsic reward r_t^e was included. Different methods were carried out via policy gradient using Adam optimization [14]. The annealing hyperparameters were selected to tune the weights for intrinsic reward and penalty in variational RL.

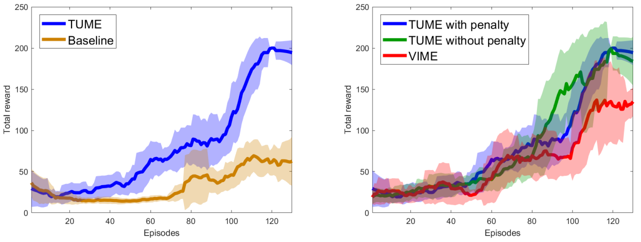


(a) without action-gating (b) with action-gating

Fig. 2: Latent features $\mathbf{z}$ of TUME in $\mathbb{R}^3$. Blue and red dots show the samples corresponding to two actions of Cart Pole.

A. Evaluation on Cart Pole

The agent of Cart Pole [15] aims to balance the stick by pushing (action 0) or pulling (action 1) the cart. The same reward 1 was repeatedly received by the environment at each time step until the termination of episode. State $s_t \in \mathbb{R}^4$ and latent variable $\mathbf{z} \in \mathbb{R}^3$ were represented. Single hidden layer was configured in the neural networks of actor, encoder, decoder and critic. $\gamma = 0.95$ and activation function tanh were used. To evaluate the effect of latent variable model, Figures 2(a) and 2(b) display the samples of latent variable $\mathbf{z}$ of two actions without and with action-gating method, respectively. TUME without action-gating scheme was implemented by simply using an augmented vector consisting of state s_t and action a_t as the inputs to encoder of TUC. Obviously, the action-gating scheme produces separate latent codes for individual actions while the latent codes of different actions are mixed without action-gating method. Figures 3(a) and 3(b) illustrate the results of total reward versus episode by comparing TUME with baseline and VIME, respectively. TUMEs without and with penalty term r_t^p are also evaluated. As we can see, baseline system, VIME and TUME roughly converge at 110 episodes. After running 130 episodes, total reward is increased from 62 using baseline to 129 using VIME and 193 using TUME. If TUME is performed without penalty r_t^p, the performance is still significantly better than VIME. Total reward is decreased to 188. Intrinsic reward r_t^i and penalty r_t^p are both important to TUME.



(a) (b)

Fig. 3: Total rewards versus episodes in Cart Pole using (a) baseline and TUME, and (b) VIME, TUME without and with penalty r_t^p.

B. Evaluation on Dialogue Management

PyDial [16], [17] is an open-source end-to-end evaluation system for task-oriented dialogue where the benchmark environments with different dialogue modules are provided. The dialogue management module based on variational RL using deep Q network (DQN) was investigated. Different exploration methods were evaluated by 18 dialogue tasks which were built by 6 environments (with different semantic error rate (0%,15% or 30%), action masking (on or off) and user model (standard or unfriendly)) and 3 domains (Cambridge (CR) and San Francisco (SFR) restaurants, and laptops (LAP) with number of actions (25 or 40)). Action masking was to test the learning capability of the algorithms. Unfriendly user meant that users provided less extra information. This paper adopted the default setting of hyperparameters in DQN provided by PyDial. DQN used ε-greedy exploration with a linear schedule starting from $\varepsilon = 0.3$ and then annealed to 0. The encoder, decoder, critic and actor networks were modeled by two to three fully-connected layers with the activation functions provided by Pydial. $\mathbf{z} \in \mathbb{R}^8$ was used. The replay buffer was set to be 6000. The maximum number of turns in dialogue was 25. $\gamma = 0.99$ was used. Every model was trained over ten different random seeds. After each 1000 training dialogues, the models were evaluated over 500 test dialogues. Dialogue performance was assessed by using the metrics of success rate and reward for policy model. Success rate was defined as the percentage

TABLE I: Success rates and rewards by using different explorations in DQN over 18 benchmarking tasks. The best numbers are bold.

		Baseline		VIME		CME		TUME w/o AG		TUME w/o r_t^p		TUME w AG r_t^p	
Task		Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.	Suc.	Rew.
Env. 1	CR	92.5%	12.2	91.6%	12.2	94.4%	12.6	94.0%	12.5	94.6%	**12.8**	**95.1%**	12.7
	SFR	74.1%	7.6	81.5%	9.0	83.0%	9.3	82.7%	9.1	83.7%	9.4	**84.1%**	**9.5**
	LAP	73.0%	7.5	74.0%	7.5	78.3%	8.2	77.8%	7.9	78.1%	**8.3**	**78.4%**	8.1
Env. 2	CR	90.7%	11.7	94.8%	12.6	95.1%	12.2	95.3%	12.5	95.1%	**12.7**	**95.5%**	12.3
	SFR	90.1%	10.7	83.0%	8.9	87.4%	10.0	86.7%	9.7	87.5%	10.4	**91.8%**	**11.0**
	LAP	84.9%	9.1	79.7%	8.2	79.4%	7.7	81.1%	8.2	86.1%	**9.8**	**86.9%**	9.3
Env. 3	CR	93.6%	12.0	94.2%	12.0	94.5%	12.0	94.3%	12.1	**95.1%**	12.2	94.9%	**12.3**
	SFR	73.3%	6.1	71.7%	5.9	75.9%	6.8	75.0%	6.6	76.9%	7.0	**78.8%**	**7.4**
	LAP	69.0%	5.6	66.1%	5.0	69.0%	5.7	68.8%	5.9	71.0%	**6.2**	**72.3%**	5.9
Env. 4	CR	86.4%	9.7	91.1%	10.9	92.9%	11.3	92.2%	10.9	90.2%	11.0	**93.2%**	**11.6**
	SFR	80.5%	8.5	78.9%	8.0	79.0%	7.8	80.2%	8.0	83.5%	8.5	**85.0%**	**9.1**
	LAP	78.6%	7.5	75.9%	7.0	83.2%	8.0	83.0%	7.8	80.5%	**8.3**	**83.3%**	7.6
Env. 5	CR	90.2%	10.0	91.2%	10.3	93.5%	10.6	93.1%	10.5	**94.3%**	10.6	94.1%	**10.9**
	SFR	71.6%	4.3	74.7%	4.8	73.3%	4.7	73.9%	4.9	79.1%	5.5	**81.2%**	**5.8**
	LAP	51.8%	0.8	57.7%	1.6	53.3%	1.0	56.4%	1.3	58.2%	**1.8**	**59.8%**	1.7
Env. 6	CR	89.9%	10.2	89.3%	10.1	89.9%	**10.3**	89.5%	10.1	89.8%	10.2	**90.0%**	10.2
	SFR	64.0%	3.3	63.9%	3.2	63.5%	3.1	64.0%	3.4	65.6%	3.6	**67.3%**	**4.1**
	LAP	56.2%	2.1	59.8%	2.6	54.3%	2.1	58.3%	2.6	59.3%	2.7	**60.5%**	**3.1**

of dialogues which were completed successfully. Reward was defined as $20 \cdot D - T$, where D was the success indicator and T was the number of dialogue turns. Table I compares the results of different methods. TUME without action-gating (AG) and penalty r_t^p (TUME w/o AG), TUME with AG but without r_t^p (TUME w/o r_t^p) and TUME with AG and r_t^p (TUME w AG r_t^p) are compared. TUME w/o AG obtains slightly worse results than CME. However, the full realization of TUME consistently performs better than the other methods in most environments and domains. The results of TUME are degraded by disregarding the penalized reward and/or action-gating. Full TUME achieves the desirable results in this comparison.

V. Conclusions

This paper has presented a variational reinforcement learning for exploring state transition uncertainty based on a critic constructed by a variational encoder-decoder network. The action-gating scheme to action-driven latent variable model was implemented. Exploration and exploitation based on the entropy reduction and the bias penalty were investigated, respectively. The discrete control tasks on Cart Pole and dialogue management were evaluated. The results of this new actor-critic model based on latent variable representation was evaluated to be consistently better than those of the newest systems based on the variational information maximizing exploration and the curiosity maximizing exploration. This general solution to exploration can be extended to other scenarios in signal processing and medical diagnosis [18].

References

[1] J. Munk, J. Kober, and R. Babuska, "Learning state representation for deep actor-critic control," in *Proc. of IEEE Conference on Decision and Control*, 2016, pp. 4667–4673.

[2] A. Venkatraman, N. Rhinehart, W. Sun, L. Pinto, M. Hebert, B. Boots, K. Kitani, and J. Bagnell, "Predictive-state decoders: Encoding the future into recurrent networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 1172–1183.

[3] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[4] J.-T. Chien, "Association pattern language modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1719–1728, 2006.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations*, 2013.

[6] J.-T. Chien and K.-T. Kuo, "Variational recurrent neural networks for speech separation," in *Proc. of Annual Conference of International Speech Communication Association*, 2017, pp. 1193–1197.

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[8] J.-T. Chien, "Deep Bayesian learning and understanding," in *Proc. of COLING: Tutorial Abstracts*, 2018, pp. 13–18.

[9] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "VIME: Variational information maximizing exploration," in *Neural Information Processing Systems*, 2016, pp. 1109–1117.

[10] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. of International Conference on Machine Learning*, 2015.

[11] L. Itti and P. F. Baldi, "Bayesian surprise attracts human attention," in *Neural Information Processing Systems*, 2006, pp. 547–554.

[12] J. Schmidhuber, "Curious model-building control systems," in *Proc. of IEEE International Joint Conference on Neural Networks*, 1991, pp. 1458–1463.

[13] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. of International Conference on Machine Learning*, 2017, pp. 2778–2787.

[14] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[16] S. Ultes, L. M. R. Barahona, P.-H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.-H. Wen, M. Gasic, et al., "Pydial: A multi-domain statistical dialogue system toolkit," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 73–78.

[17] J.-T. Chien and W. X. Lieow, "Meta learning for hyperparameter optimization in dialogue system," in *Proc. of Annual Conference of International Speech Communication Association*, 2019, pp. 839–843.

[18] H.-H. Tseng, Y. Luo, S. Cui, J.-T. Chien, R. K. Ten Haken, and I. El Naqa, "Deep reinforcement learning for automated radiation adaptation in lung cancer," *Medical physics*, vol. 44, no. 12, pp. 6690–6705, 2017.