Stochastic Curiosity Maximizing Exploration

Jen-Tzung Chien
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

Po-Chien Hsu
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

***Abstract*—Deep reinforcement learning (RL) is known as an emerging research trend in machine learning for autonomous systems. In real-world scenarios, the extrinsic rewards, acquired from the environment for learning an agent, are usually missing or extremely sparse. Such an issue of sparse reward constrains the learning capability of agent because the agent only updates the policy when the goal state is successfully attained. It is always challenging to implement an efficient exploration in RL algorithms. To tackle the sparse reward and inefficient exploration, the agent needs other helpful information to update its policy even when there is no interaction with the environment. This paper proposes the stochastic curiosity maximizing exploration (SCME), a learning strategy explored to allow the agent to act as human. We cope with the sparse reward problem by encouraging the agent to explore future diversity. To do so, a latent dynamic system is developed to acquire the latent states and latent actions to predict the variations in future conditions. The mutual information and the prediction error in the predicted states and actions are calculated as the *intrinsic rewards*. The agent based on SCME is therefore learned by maximizing these rewards to improve sample efficiency for exploration. The experiments on PyDial and Super Mario Bros show the benefits of the proposed SCME in dialogue system and computer game, respectively.**

***Index Terms*—deep reinforcement learning, sparse reward, intrinsic reward, exploration, dialogue system**

I. Introduction

Recent rapid development in deep neural networks has brought a great success in deep reinforcement learning (RL) [1] for numerous complicated applications in the domains of natural language processing [2], [3], computer vision, game playing, robotic control and autonomous driving. A typical natural language application based on RL is the task-oriented dialogue system. Deep RL is feasible to carry out a deep policy network which implements a successful dialogue to meet a specific task. In general, a reinforcement learning system aims to train an agent or learn a policy to interact with environment via sequential decision making. During the learning procedure, such an agent obtains the observations from environment, decides an action according to the policy, and then receives the feedback which is known as the reward. Agent is learned to choose an action through many trials and errors. The goal of RL is to build an agent by maximizing total rewards provided by the environment.

In general, RL methods are categorized into model-based RL and model-free RL according to the updating procedure and the way of making decisions. Model-based RL aims to learn the state transition of the environment based on Markov decision process (MDP) by using the trajectories which are stored while interacting with environment. An accurate model of the MDP is hard to build in complicated environments. It is accordingly popular to construct the model-free agent for deep RL which learns the value function for states while the prediction of next state is avoided. The agents in model-free RL can be further divided into three types including value-based, policy-based and actor-critic agents. First, the value-based agent decides a random action or an optimal action with the highest probability. This agent continuously updates the state value estimator. Second, the policy-based agent directly updates the parameterized policy through the policy gradient [4]. Third, the actor-critic agent is seen as a hybrid agent in sequential learning which is not only trained by updating the parameterized policy but also adjusting the state value estimator. As a human being, we can easily understand the environment and predict the future when facing the unseen scenarios. But, such a task is challenging for a reinforcement learner who would like to build the model-based agent to deal with sequential decision problem. This paper addresses how MDP is implemented to carry out the model-based agent where the total reward is maximized through understanding the state transition of environments. We pursue an agent who is similar to the role of humans in terms of predicting the future and planning the direction. An agent is constructed to act for self learning through exploration of useful states based on a latent dynamic system.

This study builds the latent dynamic system to carry out the proposed stochastic curiosity maximizing exploration (SCME) for deep RL. The latent dynamics of states and actions are represented by means of the variational autoencoder (VAE) [5], [6]. Basically, VAE is a powerful generative model which comprises an inference model as an encoder and a generative model as a decoder. The encoder compresses the states and actions into latent representations while the decoder generates the synthesized samples from latent state and action space. Given the latent states and actions, the second step is to predict the variational future. If the future latent state differs from the actual latent state, it means that the agent still has some region which has not been explored. Finally, the agent measures the mutual information between the predicted latent state and latent action. Agent is designed to explore those predicted states with large mutual information. This information is treated as intrinsic reward to be maximized to encourage exploration. In the experiments, we compare the performance of SCME with the exploration strategy based on deep Q network [7],

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

[8]. An open-source and end-to-end statistical spoken dialogue system toolkit based on PyDial [9], [10] is investigated. PyDial module is evaluated with 18 environments based on different conditions. We examine the benefits of SCME in different environments. The objective function using SCME consists of a number of loss functions. Each function is demonstrated to be considerable with individual function. The influence of each intrinsic reward is analyzed. The performance of SCME in presence of the other exploration strategy based on the asynchronous advantage actor-critic [11] in Super Mario Bros with dense and sparse rewards is also evaluated.

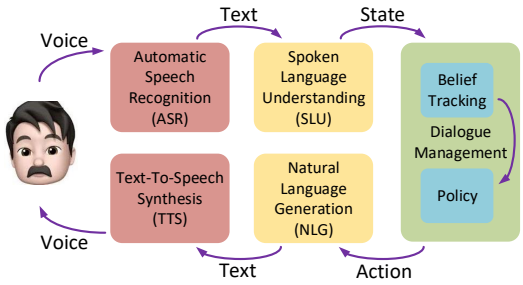

Fig. 1. Spoken dialogue is constructed as reinforcement learning system.

II. Background Survey

This section addresses the statistical spoken dialogue procedure which is seen as an autonomous system where deep reinforcement learning is performed to explore for task-oriented goal. Finding an informative intrinsic reward is crucial. A couple of learning strategies for exploration are introduced.

A. Statistical Spoken Dialogue System

A statistical spoken dialogue system (SDS) is composed of different modules where each module plays a specific role. SDS is basically a very complicated system with various uncertainties where several issues should be handled simultaneously. Figure 1 shows how a dialogue system is constructed and simulated as a RL scenario. A user interacts with dialogue system using speech input via automatic speech recognition (ASR), and then receives a spoken response via text-to-speech (TTS) synthesis. ASR module calculates the posterior probabilities of words in text sequence corresponding to an utterance. Instead of retaining the most probable hypothesis only, the N-best list of sentence hypotheses with the corresponding probabilities is often retrieved by ASR. The spoken language understanding (SLU) module is then used to understand the semantics of sentence hypothesis given the outputs of ASR. SLU module produces a type of meaning for input utterance based on a slot-value pair which is seen as the state. The dialogue management (DM) uses this observation for sequential belief tracking and decision making according to the learned policy. DM is the core module for taking actions in dialogue system. Such a module sufficiently reflects the behavior of SDS. Basically, DM is comparable as human brain, belief tracking captures the memory trajectory of a human, and policy acts for choosing actions. The behavior of policy is similar to the agent in RL. The natural language generation (NLG) module generates the sentence based on the actions from DM. TTS module receives the text from NLG, and then synthesizes an agent voice to interact with user. To facilitate the interaction between agent and user, deep RL is performed to build an efficient policy to accomplish the dialogue task.

B. Deep Reinforcement Learning

Deep Q network (DQN) [7], [8], [12] is one of the most successful algorithms for deep RL. DQN builds a value-based agent where a deep neural network (DNN) is incorporated to calculate the state-action value functions $Q_\theta(s_t, a_t)$ in network outputs at each time t for individual actions a_t where state s_t is used as input. This calculation is to estimate the expected return $Q(s_t, a_t) = \mathbb{E}[R_t|s_t, a_t]$ where $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ as the state-action value function. Here, r_t is the extrinsic reward from environment and $\gamma \in (0, 1]$ is a discount factor. DNN parameters θ in value network $Q_\theta(s_t, a_t)$ is updated by minimizing a square error loss function

$$J(\theta) = \Big(r_t + \gamma \max_a Q_\theta(s_{t+1}, a) - Q_\theta(s_t, a_t)\Big)^2 \quad (1)$$

DQN aims to predict the Q value $Q_\theta(s_t, a_t)$ which gets close to the temporal-difference (TD) [13], [14] target $r_t + \gamma \max_a Q(s_{t+1}, a)$. TD error is minimized. In addition to this value network, DQN implements Q learning by using replay buffer as well as target network. Replay buffer memories the transitions $\{s_t, a_t, r_t, s_{t+1}\}$ which are sampled in minibatch to calculate the gradient of $J(\theta)$ with respect to θ where the slow convergence in Q learning due to too close consecutive states is improved. Besides, there is no correct target in Q learning. An additional target network is merged to calculate $r_t + \gamma \max_a \widehat{Q}_{\hat{\theta}}(s_{t+1}, a)$ as TD target. It is not suitable to use the same network to calculate target value $\widehat{Q}_{\hat{\theta}}(s_{t+1}, a_t)$ and Q value $Q_\theta(s_t, a_t)$. Parameter of target network $\hat{\theta}$ is reset by that of value network θ periodically every a number of time steps. Exploration based on ϵ-greedy algorithm is applied.

Mnih et al. [11] proposed an asynchronous approach to deep RL based on an actor-critic agent. Rather than communicating with single environment in DQN, this agent communicates with several environments at the same time. The asynchronous advantage actor-critic (A3C) agent consists of a global network and multiple workers with individual parameters in presence of multiple environments. Each worker interacts with its own copy of environment The benefit of using A3C is that replay buffer is not required since there is no correlation between different environments. This A3C works better because the overall experiences from multiple workers provide more diversity in training procedure. A3C uses the actor-critic architecture which maintains a policy $\pi_\theta(a_t|s_t)$ with parameter θ as actor and estimates the state value function $V_{\theta_v}(s_t)$ with parameter θ_v by critic where $V(s_t) = \mathbb{E}_a[Q(s_t, a)]$. Policy and state value function are updated every t_{max} time steps or at the terminal state. An advantage of action a_t in state s_t is estimated by $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$. According to A3C

algorithm, the updating of θ in policy network $\pi_\theta(a_t|s_t)$ is based on promoting the advantage over increasing the reward by using the gradient $\nabla_{\theta'} \log \pi_{\theta'}(a_t|s_t) A_{\theta,\theta'_v}(s_t, a_t)$ where an estimate of advantage function is yielded by

$$A_{\theta,\theta_v}(s_t, a_t) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V_{\theta_v}(s_{t+k}) - V_{\theta_v}(s_t). \quad (2)$$

The updating of θ_v in value function $V_{\theta_v}(s_t)$ is also performed by using the gradient $\nabla_{\theta'_v}(A_{\theta',\theta'_v}(s_t, a_t))^2$. Exploration in A3C is based on maximizing the diversity over actor-learners. Basically, the exploration in DQN and A3C is insufficient. The convergence in learning procedure is likely degraded.

C. Efficient Exploration for Environment Dynamics

In reinforcement learning, the agent typically receives a positive reward when a specific goal is achieved. A penalty with negative reward is obtained if the agent fails to meet the requirement. However, in many scenarios, the agent neither succeeds nor fails to achieve final goal. Accordingly, there is no reward received from environment. The sparse reward problem happens. This problem is especially serious when the state space is large. In addition, the random exploration using the ϵ-greedy is difficult to reach many rare states in complicated environment. The efficiency for exploration in DQN is limited. How to deal with the sparse reward and carry out an efficient exploration is crucial for deep RL. In what follows, two previous approaches to improve exploration are surveyed. The intrinsic rewards are calculated.

It is important to balance the trade-off between exploration and exploitation for RL. To improve the learning efficiency for scalable RL, the exploration strategy based on variational information maximizing exploration (VIME) [15] was proposed. VIME used the entropy search, a popular Bayesian optimization method, to encourage agent to explore efficiently. The maximization of information gain was implemented to realize the agent's belief in environment dynamics. Correspondingly, the sum of entropy reduction due to new state s_{t+1} along a trajectory $\sum_t (H(\theta|\xi_t, a_t) - H(\theta|s_{t+1}, \xi_t, a_t))$ given with an experience of history is calculated. $H(\cdot)$ is the entropy function. The *intrinsic reward* at each time t is expressed as

$$r_t^i = \mathrm{KL}(p(\theta|s_{t+1}, \xi_t, a_t) \| p(\theta|\xi_t)) \quad (3)$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence, θ denotes the parameter of transition model $p(s_{t+1}|s_t, a_t, \theta)$ and $\xi_t = \{s_1, a_1, \ldots, s_t\}$ denotes the history of an agent who experiences until time step t. In practice, the KL divergence in Eq. (3) is implemented by $\mathrm{KL}(q(\theta|\phi_{t+1}) \| q(\theta|\phi_t))$ using variational distribution $q(\theta|\phi_t)$ with parameter ϕ_t updated by variational inference at each time t. Agent is trained by maximizing extrinsic reward r_t^e as well as intrinsic reward r_t^i, i.e. $r_t = r_t^e + r_t^i$. VIME follows the Bayesian perspective by maximizing the information gain or the uncertainty reduction for an agent who explores the environment dynamics through state transitions. Houthooft et al. [15] applied the Bayes-by-backprop network (BBN) [16] to learn the state transition of an environment. Weights of BBN were sampled from Gaussian distribution which implemented a robust network against the mode collapse. BBN calculated the entropy reduction or KL divergence between the posteriors of weight distributions at consecutive time steps, and used this measure as the intrinsic reward to encourage agent to explore. An efficient exploration was performed by maximizing the expected sum of reduction of uncertainty in environment dynamics. The agent explored the environment with maximum entropy reduction.

In [17], the curiosity maximizing exploration (CME) was carried out as a model-based module which was called the intrinsic curiosity module (ICM). ICM module extracted the transition information from input tuple $\{s_t, a_t, s_{t+1}\}$ and used it as the intrinsic reward r_t^i for agent learning. CME trained a forward dynamics model $f(\cdot)$ with parameter θ_f that predicted the feature representation of next state $\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t, \theta_f)$ where $\phi(\cdot)$ was a feature extractor. The difference between the features of predicted state $\hat{\phi}(s_{t+1})$ and actual state $\phi(s_{t+1})$ was measured as the prediction error or a negative intrinsic reward which was minimized to encourage the curiosity and pursue the best state prediction during exploration. The value of intrinsic reward in CME with a scaling factor $\eta > 0$ was computed as

$$r_t^i = \frac{\eta}{2} \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2. \quad (4)$$

A self-supervised prediction was performed. In addition, an inverse neural network model with parameter θ_g was merged to predict an action $\hat{a}_t = g(s_t, s_{t+1}, \theta_g)$ where the discrepancy between the predicted $\hat{a}_t$ and actual actions a_t was minimized. Using CME, the tuples $\{s_t, a_t, s_{t+1}\}$ were used to jointly train the policy network $\pi(\cdot)$, the feature encoder $\phi(\cdot)$, the forward model $f(\cdot)$ and the inverse model $g(\cdot)$.

III. Stochastic Curiosity Maximizing Exploration

VIME promotes the exploration by maximizing the reduction of uncertainty using Bayesian neural network while CME performs the self-supervised exploration by maximizing the curiosity [18] or minimizing the prediction error of the compressed states. This paper boosts the strengths of VIME and CME and proposes the stochastic curiosity maximizing exploration (SCME) where the information-theoretic curiosity is maximized for reinforcement learning. Figure 2 shows the diagram of RL based on SCME. In addition to policy network $\pi_\theta(a_t|s_t, r_t)$, there are three components in SCME which include encoder network, curiosity network and the information network as detailed in what follows.

A. Latent Dynamic System

RL aims to train an agent by using the trajectories of state s_t, action a_t and reward r_t at different time steps t [19], [20]. It is crucial to learn the state transitions or build a dynamic system for environment dynamics which characterizes the relations among current state s_t, action a_t and next state s_{t+1}. Usually, states and actions are high-dimensional especially in continuous control tasks. To facilitate stochastic modeling in RL, this study develops a latent dynamic system

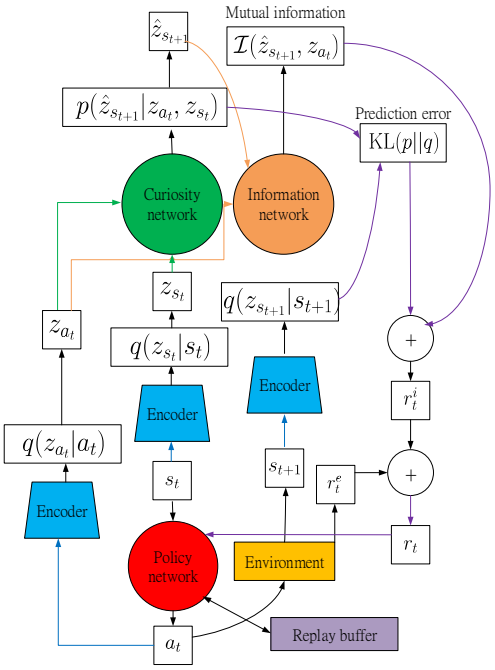

Fig. 2. Diagram of stochastic curiosity maximizing exploration for reinforcement learning which consists of the policy network (red), the encoder networks (blue), the curiosity network (green), and the information network (orange).

to simulate the unknown behavior of environment where the uncertainties of state s_t and action a_t are represented for information-theoretic exploration. In addition to uncertainty modeling, this latent dynamic system also promotes learning efficiency for aspects of environment where the redundancy or irrelevance in information extraction can be reduced during exploration from high-dimensional state space. In [5], variational autoencoder (VAE) was proposed as a general extractor or encoder for random features from inputs. This paper builds a latent dynamic system where low-dimensional latent states and actions are extracted by the learned encoders for abstract and compressed representations of high-dimensional states and actions, respectively. Variational inference is performed for curiosity-driven exploration. A variational RL is developed by maximizing three objectives including

1) evidence lower bounds (ELBOs) of log likelihoods for states and actions
2) intrinsic reward based on curiosity in terms of the prediction error of latent state
3) mutual information between new state s_{t+1} and current action a_t in latent space.

In the implementation, low-dimensional random state z_{s_t} and action z_{a_t} are calculated by using two encoder networks where high-dimensional state s_t and action a_t are used as inputs, respectively. It is meaningful to use the same encoder for current state s_t and next state s_{t+1}. Importantly, stochastic modeling is embedded in a latent dynamic system based on z_{s_t}, z_{a_t} and $z_{s_{t+1}}$, which are learned by maximizing the ELBOs for state and action encoders as given by [5], [21]

$$\begin{aligned}\mathcal{L}(s;\theta_s,\phi_s) &= \mathcal{L}_{\theta_s} + \mathcal{L}_{\phi_s} \\ &= \mathbb{E}_{q_{\phi_s}(z_s|s)}[\log p_{\theta_s}(s|z_s)] - \mathrm{KL}(q_{\phi_s}(z_s|s)\|p(z_s))\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}(a;\theta_d,\phi_d) &= \mathcal{L}_{\theta_a} + \mathcal{L}_{\phi_a} \\ &= \mathbb{E}_{q_{\phi_a}(z_a|a)}[\log p_{\theta_a}(a|z_a)] - \mathrm{KL}(q_{\phi_a}(z_a|a)\|p(z_a)).\end{aligned} \tag{6}$$

There are two terms in ELBOs. One is the log likelihoods $\{\log p_{\theta_s}(s|z_s), \log p_{\theta_a}(a|z_a)\}$ with latent variables $\{z_s, z_a\}$ sampled by variational distributions $\{q_{\phi_s}(z_s|s), q_{\phi_a}(z_a|a)\}$. The other is the KL terms for regularizing those variational distributions to get close to their priors $\{p(z_s), p(z_a)\}$.

B. Stochastic Curiosity Learning

This latent dynamic system is constructed with twofold considerations. One is to faithfully reflect stochastic features from heterogeneous observations of state and action. The other is to facilitate the stochastic curiosity learning which deals with sparse reward as well as model uncertainty for exploration in RL. The agent learns under latent dynamic space based on the intrinsic reward calculated from current random variables z_{s_t} and z_{a_t} according to curiosity network with output probability $p(\hat{z}_{s_{t+1}}|z_{a_t}, z_{s_t})$. This network provides high-level abstraction to predict future state $\hat{z}_{s_{t+1}}$. The agent of SCME learns to explore those unseen regions in environment which are found by optimizing the stochastic curiosity. This curiosity-driven exploration is performed by maximizing the difference between the predictive information of next state $\hat{z}_{s_{t+1}}$ and the information measured by the distribution of next state $z_{s_{t+1}}$ acquired in real environment. The stochastic curiosity is measured by KL divergence between the distribution of predicted state $p(\hat{z}_{s_{t+1}}|z_{s_t}, z_{a_t})$ via curiosity network and the variational distribution of actual state $q(z_{s_{t+1}}|s_t)$ via state encoder. The objective is to explore future in next state by

$$\mathcal{L}_{\theta_{\text{cur}}} = \mathrm{KL}(p(\hat{z}_{s_{t+1}}|z_{s_t}, z_{a_t})\|q(z_{s_{t+1}}|s_{t+1})). \tag{7}$$

In the implementation, we sample the latent variables of the predicted state $\hat{z}_{s_{t+1}}$ from predictive distribution as well as the actual state $z_{s_{t+1}}$ from variational distribution. The intrinsic reward is correspondingly calculated by $r_t^i = ||\hat{z}_{s_{t+1}} - z_{s_{t+1}}||_2^2$. This study further strengthens the information-theoretic learning in SCME by incorporating a mutual information term as regularization in curiosity-based intrinsic reward.

C. Information-Theoretical Learning

Information theory provides meaningful criterion to estimate the informative distributions based on knowledge or constraint [21]–[24]. This study learns an agent by interacting with environment which can enhance the mutual information between latent variables of the predicted state $\hat{z}_{s_{t+1}}$ and the selected action z_{a_t} from policy network. An information network is merged to compute mutual information at each time t by

$$I(\hat{z}_{s_{t+1}}, z_{a_t}) = \mathbb{E}_{p(\hat{z}_{s_{t+1}}, z_{a_t})}\left[\log \frac{p(\hat{z}_{s_{t+1}}, z_{a_t})}{p(\hat{z}_{s_{t+1}})p(z_{a_t})}\right] \tag{8}$$

which is maximized to encourage the dependencies between the outputs of curiosity network $\hat{z}_{s_{t+1}}$ and action encoder

z_{a_t}. The selected action can sufficiently reflect the exploration based on stochastic curiosity. Intrinsic reward is extended as

$$\tilde{r}_t^i = ||\hat{z}_{s_{t+1}} - z_{s_{t+1}}||_2^2 + I(\hat{z}_{s_{t+1}}, z_{a_t}) \quad (9)$$

where the mutual information acts as a regularization which brings in a regularized latent variable $\hat{z}_{s_{t+1}}$ via information-theoretic learning for curiosity-driven exploration. However, it is challenging to construct a neural estimation for mutual information so that an analytical neural network solution to deep RL based on SCME can be implemented. Accordingly, we refer to [25] and derive the variational lower bound of mutual information, parameterized by neural network, as a tractable and scalable objective for implementation of SCME.

Basically, the problem is to select a family of functions $M_\theta : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ which are parametrized by a neural network model with parameters $\theta \in \Theta$. The lower bound of mutual information is expressed in $\mathcal{I}(A, B) \geq \mathcal{I}_\Theta(A, B)$ where the bound $I_\Theta(A, B)$ is derived as a neural information [26]

$$\mathcal{I}_\Theta(A, B) = \sup_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}_{AB}}[M_\theta] - \log(\mathbb{E}_{\mathbb{P}_A \otimes \mathbb{P}_B}[e^{M_\theta}]) \quad (10)$$

where $\mathbb{P}_{AB}$ and $\mathbb{P}_A \otimes \mathbb{P}_B$ denote the joint and the product of marginals in probability space, respectively. In implementation of SCME, the parameter of information network θ_{inf} is incorporated in the lower bound of mutual information between latent state $\hat{z}_{s_{t+1}}$ and latent action z_{a_t}. The neural network parameter θ_{inf} is used to calculate the function $M_{\theta_{\text{inf}}}$ for the objective of mutual information (MI) $\mathcal{L}_{\theta_{\text{inf}}}$. The bound $\mathcal{L}_{\theta_{\text{inf}}}$ is calculated by the joint distribution $p(\hat{z}_{s_{t+1}}, z_{a_t})$ and the marginal distributions $p(\hat{z}_{s_{t+1}})$ and $p(z_{a_t})$ by using N minibatch samples. Joint distribution is calculated from the transitions in replay buffer. Marginals are calculated by shuffling the individual samples of $\hat{z}_{s_{t+1}}$ or z_{a_t} in the transitions. The Donsker-Varadhan representation of MI is obtained by

$$\begin{aligned}
\mathcal{I}(z_s, z_a) &= \sum_{z_s \in Z_S, z_a \in Z_A} p(z_s, z_a) \log \frac{p_{(z_s, z_a)}(z_s, z_a)}{p_{z_s}(z_s) p_{z_a}(z_a)} \\
&= H(z_a) - H(z_a|z_s) = \text{KL}(p_{z_s, z_a} \| p_{z_s} p_{z_a}) \\
&\geq \sup_{\theta_{\text{inf}} \in \Theta} \mathbb{E}_{p_{(z_s, z_a)}}[M_{\theta_{\text{inf}}}] - \log(\mathbb{E}_{p(z_s)p(z_a)}[e^{M_{\theta_{\text{inf}}}}]) \\
&= \frac{1}{N} \sum_{n=1}^N M_{\theta_{\text{inf}}}(z_s^{(n)}, z_a^{(n)}) - \log(\frac{1}{N} \sum_{n=1}^N e^{M_{\theta_{\text{inf}}}(z_s^{(n)}, z_a^{(n)})}) \\
&\triangleq \mathcal{L}_{\theta_{\text{inf}}}.
\end{aligned} \quad (11)$$

D. Implementation Procedure

Deep RL using SCME is then implemented by training the state encoder (Eq. (5)), action encoder (Eq. (6)), curiosity network (Eq. (7)), information network (Eq. (11)) and policy network $\pi_\theta(a_t|s_t)$ by jointly optimizing four objectives with two regularization parameters λ_c and λ_c

$$\mathcal{L} = \mathcal{L}(s; \theta_s, \phi_s) + \mathcal{L}(a; \theta_a, \phi_a) + \lambda_c \mathcal{L}_{\theta_{\text{cur}}} + \lambda_i \mathcal{L}_{\theta_{\text{inf}}}. \quad (12)$$

Fig. 2 shows the details of system diagram and optimization procedure using SCME where the sequence of optimization steps is addressed. The agent is trained by implementing the following nine steps where this learner

1) interacts with the environment, and saves the transitions $\{s_t, a_t, s_{t+1}, r_t^e\}$ in the replay buffer
2) samples a minibatch of transitions from replay buffer
3) produces the belief probabilities for each transition based on the state and action encoders for latent spaces of current state $q(z_{s_t}|s_t)$, next state $q(z_{s_{t+1}}|s_{t+1})$ and current action $q(z_{a_t}|a_t)$
4) applies the reparameterization trick (by referring to VAE [5]) to sample latent states z_{s_t}, z_{st+1} and action z_{a_t}
5) uses the samples z_{a_t}, z_{s_t} and the curiosity network to predict the next latent state $\hat{z}_{s_{t+1}}$ via $p(\hat{z}_{s_{t+1}}|z_{a_t}, z_{s_t})$
6) measures the KL divergence between the predicted $p(\hat{z}_{s_{t+1}}|z_{a_t}, z_{s_t})$ and the actual latent state distributions $q(z_{s_{t+1}}|s_{t+1})$
7) estimates the mutual information $\mathcal{I}(\hat{z}_{s_{t+1}}, z_{a_t})$ from the samples $\hat{z}_{s_{t+1}}$ and z_{a_t} by using information network
8) updates the state encoder ϕ_s, action encoder ϕ_a, curiosity network θ_{cur} and information network θ_{inf}
9) calculates the intrinsic reward $\tilde{r}_t^i$, explores the environment and updates the policy network $\pi_\theta(a_t|s_t)$

IV. Experiments

SCME was implemented for deep RL and evaluated for the spoken dialogue system based on PyDial toolkit. OpenAI Gym was used to evaluate deep RL under different conditions.

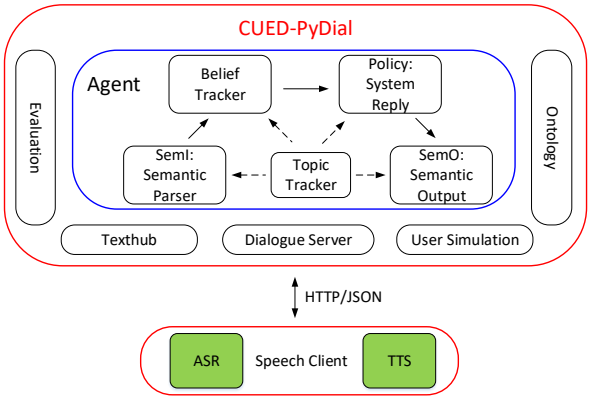

Fig. 3. Statistical spoken dialogue system module using PyDial

A. Experiments on PyDial

PyDial [9], [10], [27] is an open-source end-to-end evaluation system for task-oriented dialogue where the benchmark environments with different dialogue modules are provided as shown in Figure 3. The dialogue management module based on deep RL using DQN and other algorithms could be 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)). Action masking was to test the learning

TABLE I
COMPARISON OF SUCCESS RATES AND REWARDS BY USING DIFFERENT EXPLORATIONS IN DQN UNDER 18 BENCHMARKING TASKS.

Task		DQN		VIME		CME		SCME		SCME-MI	
		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	94.8%	12.7	**95.6%**	**13.0**
	SFR	74.1%	7.6	81.5%	9.0	83.0%	9.3	**84.7%**	9.6	84.6%	**9.7**
	LAP	73.0%	7.5	74.0%	7.5	78.3%	8.2	76.8%	8.0	**78.5%**	**8.3**
Env. 2	CR	90.7%	11.7	94.8%	12.6	95.1%	12.2	**95.7%**	**12.8**	95.1%	12.4
	SFR	90.1%	10.7	83.0%	8.9	87.4%	10.0	87.5%	10.5	**92.5%**	**11.3**
	LAP	84.9%	9.1	79.7%	8.2	79.4%	7.7	87.2%	9.4	**89.4%**	**10.5**
Env. 3	CR	93.6%	12.0	94.2%	12.0	94.5%	12.0	95.2%	12.3	**96.1%**	**12.4**
	SFR	73.3%	6.1	71.7%	5.9	75.9%	6.8	77.3%	7.0	**82.7%**	**8.0**
	LAP	69.0%	5.6	66.1%	5.0	69.0%	5.7	73.0%	6.1	**73.8%**	**6.4**
Env. 4	CR	86.4%	9.7	91.1%	10.9	92.9%	11.3	90.2%	10.8	**93.6%**	**11.4**
	SFR	80.5%	8.5	78.9%	8.0	79.0%	7.8	84.7%	8.8	**87.1%**	**9.5**
	LAP	78.6%	7.5	75.9%	7.0	83.2%	8.0	**83.3%**	**8.5**	80.2%	7.8
Env. 5	CR	90.2%	10.0	91.2%	10.3	93.5%	10.6	93.5%	10.7	**94.8%**	**11.2**
	SFR	71.6%	4.3	74.7%	4.8	73.3%	4.7	80.4%	6.2	**82.2%**	**6.5**
	LAP	51.8%	0.8	57.7%	1.6	53.3%	1.0	59.4%	1.9	**61.1%**	**2.2**
Env. 6	CR	89.9%	10.2	89.3%	10.1	89.9%	10.2	**90.2%**	**10.3**	89.7%	10.2
	SFR	64.0%	3.3	63.9%	3.2	63.5%	3.1	66.6%	3.7	**70.4%**	**4.6**
	LAP	56.2%	2.1	59.8%	2.6	54.3%	2.1	58.3%	2.7	**61.9%**	**3.3**

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. Regularization parameters $\lambda_c = 0.2$ and $\lambda_i = 1.0$ were specified. The encoder, curiosity, information and policy networks were modeled by two to three fully-connected layers with the setting of activation functions provided by Pydial. The Adam optimizer [28] with initial learning rate 0.001 was used. The replay buffer was set to be 6000. The maximum number of turns in dialogue was 25. The discount factor was 0.99. 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 using different explorations. Success rate was defined as the percentage 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.

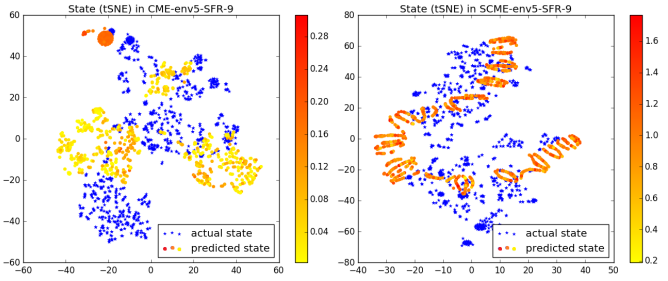

Fig. 4. Predicted state and actual state after training with 9000 dialogues in Env. 5 of SFR domain. Color bar indicates the values of intrinsic rewards. **Left:** CME model. **Right:** SCME model.

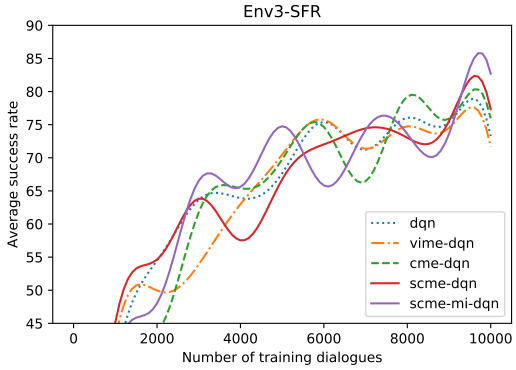

Fig. 5. Comparison of success rates in Env. 3 of SFR domain.

Fig. 4 compares the latent variables of predicted states $\hat{z}_{s_{t+1}}$ and actual states $z_{s_{t+1}}$ by using CME and SCME. t-SNE [29] was applied to show two-dimensional samples. SCME and SCME-MI denotes the proposed SCME without and with mutual information included in intrinsic reward for exploration, respectively, The values of intrinsic rewards in the prediction are shown. The DQNs with ϵ-greedy exploration (denoted by DQN) and other explorations based on VIME, CME, SCME and SCME-MI were implemented for comparison. We can see that CME does encourage the agent to explore the unknown environment, but not all the predicted regions have high intrinsic reward. The proposed SCME also encourages the agent to explore the unknown regions where the state distribution sufficiently reflects the environment. The intrinsic rewards in the predicted states are high. SCME does maximize the exploration. In addition, Fig. 5 compares the success rates of different methods in learning procedure. SCME-MI is more complicated than other methods. SCME-MI learns slowly in the beginning, but grows quickly by

increasing number of training dialogues. SCME-MI still grows up after 10000 training dialogues, but other methods converge gradually. SCME-MI performs better than the other methods. SCMEs obtains higher success rates than VIME and CME.

Table I summarizes the rewards and success rates with 10000 training dialogues. In general, the average performances of all methods are not improved significantly in CR domain which is an easy domain in PyDial where the sophisticated exploration is not required. In the SFR and LAP domains, SCMEs perform better than the others because the exploration is improved in complicated environments with sparse reward. In particular, SFR domain is a high-dimensional state task where SCMEs work quite well. This is because SCMEs built the latent dynamic space in deep RL which can capture informative features about high dimension states. More importantly, SCME with mutual information works better than the method without mutual information. This implies that mutual information can help the agent to explore the informative states. We can see that SCME-MI has considerable improvement in environment 5 where an unfriendly user setting was considered. SCME-MI provides large curiosity for maximizing the exploration for unseen states in future. The proposed SCME can handle the heterogeneous environment with unfriendly user problem.

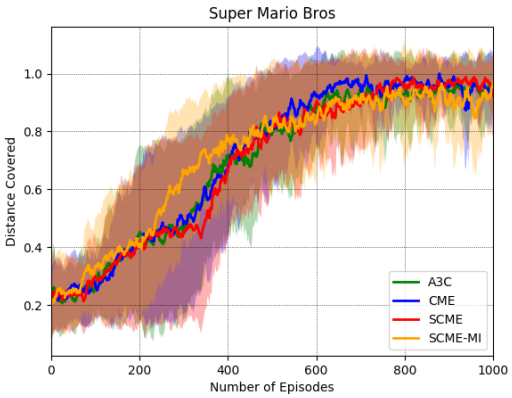

Fig. 6. Performance evaluation for different methods under the dense reward setting.

B. Experiments on OpenAI Gym

Super Mario Bros is a game playing task in OpenAI Gym [30]. Action space consisted of 14 discrete actions ranging across 7 actions including left, right, up, down, run, jump and no action. Different exploration methods were evaluated under the first level within the first world in the environment of Super Mario Bros. The episodes were terminated when the agent died, finished the game or ran out of time. The scenarios of dense reward and sparse reward were implemented and compared. In dense reward setting, the agent received reward at each time step. In sparse reward setting, the agent only received positive reward when experiencing each 5% of the game or reaching the goal, and received negative reward when

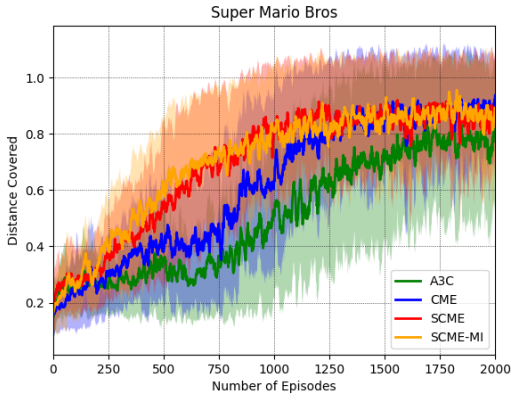

Fig. 7. Performance evaluation for different methods under the sparse reward setting.

the agent died or ran out of time. A3C algorithm was carried out and the explorations based on CME, SCME and SCME-MI were compared. The default setting in CME [17] was adopted. The settings of SCME and SCME-MI in OpenAI Gym were the same as those in PyDial task. System performance is evaluated by the metric of the distance that agent covered in the game. Under the dense reward setting, Figure 6 shows that different methods obtain comparable performance. This implies that the explorations based on CME and SCME are not so required for deep RL in dense reward setting. On the other hand, under the sparse reward setting, Figure 7 illustrates that the explorations based on CME and SCME can help A3C learning. SCME and SCME-MI performs better than CME. The baseline A3C agent gets worse in 200 episodes under dense reward setting and 500 episodes under sparse reward setting. The exploration based on SCME avoids getting stuck.

V. Conclusions

This paper presented the stochastic curiosity maximizing exploration which is a general solution to model-based reinforcement learning. The proposed approach dealt with the sparse reward task and maximized the exploration via information-theoretic learning. The variational inference was introduced to learn the latent dynamics for environment where high-dimensional state space could be compactly represented. The curiosity network was trained to predict the latent future with diversity. The information network was learned to measure the mutual information as a regularized exploration for future. By using this embedding information as intrinsic reward, the agent learned by itself and explored for useful future. Experiments on dialogue system demonstrate the effectiveness of the proposed method in different environments.The reward was increased when the variational inference was run for curiosity maximizing exploration. The results on OpenAI Gym showed the benefit of the proposed exploration in sparse reward setting. This study develops a new exploration scheme which is general and could be extended to different reinforcement learning algorithms and tasks for different applications.

REFERENCES

[1] N. Chentanez, A. G. Barto, and S. P. Singh, "Intrinsically motivated reinforcement learning," in *Advances in Neural Information Processing Systems*, 2005, pp. 1281–1288.

[2] J.-T. Chien, "Deep Bayesian learning and understanding," in *Proc. of International Conference on Computational Linguistics: Tutorial Abstracts*, 2018, pp. 13–18.

[3] J.-T. Chien, "Deep Bayesian natural language processing," in *Proc. of Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2019, pp. 25–30.

[4] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[6] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[8] 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.

[9] 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.

[10] I. Casanueva, P. Budzianowski, P.-H. Su, N. Mrkšić, T.-H. Wen, S. Ultes, L. Rojas-Barahona, S. Young, and M. Gašić, "A benchmarking environment for reinforcement learning based task oriented dialogue management," *arXiv preprint arXiv:1711.11023*, 2017.

[11] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and . Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of International Conference on Machine Learning*, 2016, pp. 1928–1937.

[12] 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.

[13] R. S. Sutton, "Temporal credit assignment in reinforcement learning," 1985.

[14] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[15] 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.

[16] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. of International Conference on Machine Learning*, 2015, pp. 1613–1622.

[17] 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.

[18] T. Hester and P. Stone, "Intrinsically motivated model learning for developing curious robots," *Artificial Intelligence*, vol. 247, pp. 170 – 186, 2017.

[19] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *Advances in Neural information Processing Systems*, 2017, pp. 4055–4065.

[20] S. Mahadevan, "Proto-value functions: Developmental reinforcement learning," in *Proc. of International Conference on Machine Learning*, 2005, pp. 553–560.

[21] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[22] J.-T. Chien and H.-L. Hsieh, "Convex divergence ica for blind source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 302–313, 2012.

[23] Y. Tu, M.-W. Mak, and J.-T. Chien, "Information maximized variational domain adversarial learning for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6449–6453.

[24] J.-T. Chien, *Source Separation and Machine Learning*, Academic Press, 2018.

[25] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "MINE: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[26] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain Markov process expectations for large time, II," *Communications on Pure and Applied Mathematics*, vol. 28, no. 2, pp. 279–301, 1975.

[27] 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.

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

[29] L. van der Maaten and G. E. Hinton, "Visualizing data using *t*-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

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