A Dirichlet Process Mixture of Robust Task Models for Scalable Lifelong Reinforcement Learning

Zhi Wang, *Member, IEEE*, Chunlin Chen, *Senior Member, IEEE*, and Daoyi Dong, *Senior Member, IEEE*

***Abstract*—While reinforcement learning (RL) algorithms are achieving state-of-the-art performance in various challenging tasks, they can easily encounter catastrophic forgetting or interference when faced with lifelong streaming information. In this article, we propose a scalable lifelong RL method that dynamically expands the network capacity to accommodate new knowledge while preventing past memories from being perturbed. We use a Dirichlet process mixture to model the nonstationary task distribution, which captures task relatedness by estimating the likelihood of task-to-cluster assignments and clusters the task models in a latent space. We formulate the prior distribution of the mixture as a Chinese restaurant process (CRP) that instantiates new mixture components as needed. The update and expansion of the mixture are governed by the Bayesian nonparametric framework with an expectation maximization (EM) procedure, which dynamically adapts the model complexity without explicit task boundaries or heuristics. Moreover, we use the domain randomization technique to train robust prior parameters for the initialization of each task model in the mixture; thus, the resulting model can better generalize and adapt to unseen tasks. With extensive experiments conducted on robot navigation and locomotion domains, we show that our method successfully facilitates scalable lifelong RL and outperforms relevant existing methods.**

***Index Terms*—Chinese restaurant process (CRP), Dirichlet process mixture, domain randomization, expectation maximization (EM), lifelong reinforcement learning (RL).**

I. Introduction

LIFELONG learning, also referred to as continual learning, corresponds to the capability of continually accommodating new information throughout the lifespan without forgetting previous knowledge, which is crucial for artificially intelligent agents performing in real-world scenarios and proceeding multiple tasks in sequence [1] An effective lifelong learning system must satisfy two potentially conflicting goals of stably maintaining old skills and rapidly acquiring a new skill. These simultaneous constraints represent the long-standing challenge as *stability-plasticity dilemma* [2]. While reinforcement learning (RL) algorithms [3] have achieved state-of-the-art performance in various challenging tasks [4]–[10], they typically exhibit poor sample efficiency and generalization ability when trained on sequential tasks, since continual acquiring new information from a nonstationary task distribution can easily result in catastrophic forgetting or interference [11]. Learning systems are trained to keep outputs consistent with inputs using explicit or implicit parametric function approximation. Training them toward a new objective will change the data distribution and lead to abrupt erasure of previously acquired knowledge, resulting in high plasticity but little stability.

Previous attempts to alleviate catastrophic forgetting in lifelong RL settings can generally be classified into three categories: 1) replay based [11], [12]; 2) regularization based [13], [14]; and 3) expansion based [15], [16]. Replay-based approaches use a replay buffer to store old samples, which are reproduced for rehearsal and interleaving online updates when learning a new task. They require large working memory to store and replay old samples, which might not be viable in real-world situations [17]. Regularization-based approaches retain old knowledge by adding regularization terms that impose constraints on the update of network weights and prevent large changes on significant weights. With a limited amount of neural resources, comprising additional loss terms can result in a tradeoff on the accomplishment of the old and new tasks [1]. In contrast, expansion-based approaches differ from the others in that they dynamically expand the model architecture, for example, a policy/option library [18], [19] or the network capacity [20], upon the arrival of each task to accommodate new knowledge. Therefore, they can mitigate catastrophic forgetting by avoiding the perturbation on past memories from the new information [21]. However, previous expansion-based approaches typically suffer from the lack of scalability due to two critical limitations: 1) they heavily rely on explicit task boundaries and hand-designed heuristics for incorporating new resources and 2) the network size may scale quadratically in the number of encountered tasks [15].

Humans can continually accommodate new information and expand cognitive capabilities while preventing past memories from being perturbed. For the purpose of artificial general intelligence (AGI), RL algorithms ought to continually build on their experiences to develop increasingly complex skills and adapt quickly to new tasks throughout their lifetime [22],

Manuscript received January 12, 2022; accepted April 22, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62006111 and Grant 62073160; in part by the Australian Research Council's Discovery Projects funding scheme under Project DP190101566; and in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20200330. This article was recommended by Associate Editor H. Liu. *(Corresponding author: Chunlin Chen.)*

Zhi Wang and Chunlin Chen are with the School of Management and Engineering, Nanjing University, Nanjing 210093, China (e-mail: zhiwang@nju.edu.cn; clchen@nju.edu.cn).

Daoyi Dong is with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (e-mail: daoyidong@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3170485.

Digital Object Identifier 10.1109/TCYB.2022.3170485

without forgetting what has already been learned. In this article, we aim at a novel expansion-based method for scalable lifelong RL, with the assumption that task boundaries are not provided explicitly. We develop and maintain a Dirichlet process mixture of task models to tackle the nonstationary task distribution, which captures task relatedness by estimating the likelihood of assigning each task to mixture components and clusters the task models in a latent space. We formulate the prior distribution of the mixture as a Chinese restaurant process (CRP) that assigns some probability of instantiating a new task model as needed. During lifelong learning, the mixture model is updated via an expectation maximization (EM) procedure, where the E-step calculates the posterior inference of task-to-cluster probabilities and the M-step updates all model parameters for future learning. Furthermore, we adopt the domain randomization technique to train *robust* prior parameters for the initialization of each task model in the mixture, thus the resulting model can better generalize and adapt to unseen tasks.

Our primary contribution is a scalable lifelong RL method that uses an EM procedure to learn a Dirichlet process mixture of robust task models with a flexible memory system, where the prior distribution of the mixture is formulated as a CRP. With explicitly estimating task relatedness, our method has the potential to enhance the stability of past memories by modulating transferability across similar tasks, and to promote plasticity by recognizing outlier tasks that require a more significant degree of adaptation. The mixture is updated and expanded under the Bayesian nonparametric framework that dynamically adapts the model complexity over the agent's lifetime, instead of fixing the model complexity beforehand in parametric approaches. By treating the task-to-cluster assignments as latent variables, our method is capable of adapting to the nonstationary task distribution without task boundaries or hand-designed heuristics for incorporating new resources. Our method is evaluated in the context of deep deterministic policy gradient (DDPG) algorithm on robot navigation and MuJoCo [23] locomotion domains in lifelong learning settings. Experimental results verify that our method facilitates efficient lifelong RL and outperforms several baseline methods.

The remainder of this article is structured as follows. Section II introduces preliminaries of RL algorithms. Section III presents the problem statement and our method in detail. Section IV shows the experimentation on the robot navigation and MuJoCo locomotion domains. Section V reviews related work regarding lifelong RL, and Section VI presents concluding remarks and future work.

II. Preliminaries

A. Reinforcement Learning

The standard paradigm of an RL agent interacting with an environment is formalized as a Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, \infty)$ defines the probability density function of transitioning to state $s_{t+1} \in \mathcal{S}$ conditioned on the agent taking action $a_t \in \mathcal{A}$ in state $s_t \in \mathcal{S}$, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function that maps each transition (s_t, a_t) to a scalar, and $\gamma \in [0, 1)$ is the discounting factor. RL aims to learn a policy, a probability density function over available actions given a state $\pi(a_t|s_t)$, that maximizes the expected return as

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{1}$$

where $r_t \sim r(s_t, a_t)$ denotes the received reward after taking action a_t in state s_t.

Model-free methods directly interact with an initially unknown environment to learn optimal policies, releasing the dependency on an explicit model or any prior knowledge of the environment. Off-policy methods decouple the behavior and target policies, enabling an agent to learn using samples collected by arbitrary policies or from replay buffers. In our method, we utilize the DDPG [24] algorithm, a popular variant of the model-free off-policy Q-learning [25] algorithm for continuous control.

B. Q-Learning

The expected return is related to the optimal action-value function as

$$Q^*(s, a) = \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} | s_t = s, a_t = a\right] \tag{2}$$

which is the maximal sum of rewards $r_{t'} \sim r(s_{t'}, a_{t'})$ multiplied by the discount factor γ at each step t', after executing action a in state s. The optimal Q-function obeys a significant identity, that is, the Bellman equation [26]

$$Q^*(s, a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*\left(s', a'\right) | s, a\right] \tag{3}$$

where $r \sim r(s, a)$. In tabular cases, the widely used Q-learning algorithm [25] updates the action-value function as

$$Q(s, a) \leftarrow Q(s, a) + \alpha\left(r + \gamma \max_{a'} Q\left(s', a'\right) - Q(s, a)\right) \tag{4}$$

where $\alpha \in (0, 1]$ is the learning rate.

C. Deep Deterministic Policy Gradient

For generalization in continuous state spaces, we usually use a function approximator to approximate the action-value function, $Q_{\boldsymbol{\varphi}}(s, a) \approx Q^*(s, a)$, where $\boldsymbol{\varphi}$ denote the learning parameters. In deep RL, a deep neural network (DNN) is utilized to approximate the Q-function, known as the famous deep Q-network (DQN) [27]. The parameters $\boldsymbol{\varphi}$ are adjusted at each iteration to minimize the mean-squared error (MSE) in the Bellman equation, that is, Bellman residual. This induces a loss function $\mathcal{L}(\boldsymbol{\varphi})$ that varies at every learning iteration

$$\mathcal{L}(\boldsymbol{\varphi}) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q_{\boldsymbol{\varphi}}\left(s', a'\right) - Q_{\boldsymbol{\varphi}}(s, a)\right)^2\right]. \tag{5}$$

To be applicable to continuous action spaces, DDPG [24] learns a deterministic neural network policy $\mu_{\boldsymbol{\phi}}$ (i.e., the actor) along with the action-value function $Q_{\boldsymbol{\varphi}}$ (i.e., the critic)

by performing gradient updates on parameter sets $\boldsymbol{\varphi}$ and $\boldsymbol{\phi}$. Analogous to the classical Q-learning, the critic is trained to minimize the Bellman residual over all sampled transitions as

$$\mathcal{L}(\boldsymbol{\varphi}) = \mathbb{E}_{s,a,r,s'}\Big[\big(r + \gamma Q_{\boldsymbol{\varphi}}\big(s', \mu_{\boldsymbol{\phi}}\big(s'\big)\big) - Q_{\boldsymbol{\varphi}}(s,a)\big)^2\Big]. \quad (6)$$

The actor is then trained to yield actions that maximize the Q-values estimated by the critic, equivalent to minimizing the loss function as

$$\mathcal{L}(\boldsymbol{\phi}) = -\mathbb{E}_{s,a,r,s'}\big[Q_{\boldsymbol{\varphi}}(s, \mu_{\boldsymbol{\phi}}(s))\big]. \quad (7)$$

While DDPG trains a deterministic policy, its behavior policy used to collect transitions during training is usually augmented with a Gaussian (or Ornstein-Uhlenbeck) noise. Therefore, actions are collected as $a \sim \mathcal{N}(\mu_{\boldsymbol{\phi}}(s), \varsigma^2)$ for fixed standard deviation ς.

III. Method

In this section, we first formulate the problem statement of lifelong RL. Then, we explain the idea of modeling the latent task structure with a mixture model to deal with the nonstationary task distribution. Next, we present the nonparametric Bayesian inference framework that formulates the prior distribution over the mixture of task clusters as a CRP and updates the mixture using the EM algorithm. Finally, we introduce the domain randomization approach that trains the robust prior parameters for each task model.

A. Problem Statement

Let $\boldsymbol{\varphi}$ and $\boldsymbol{\phi}$ denote the weights of the DQN (critic) and the policy network (actor), respectively, and $\boldsymbol{\theta} = (\boldsymbol{\varphi}, \boldsymbol{\phi})$. The model receives the state–action pair (s, a) as its input x and produces an action-value function prediction $Q_{\boldsymbol{\varphi}}(s, a)$ as its predicted output $\hat{y}$. The target value $r + \gamma Q_{\boldsymbol{\varphi}}(s', \mu_{\boldsymbol{\phi}}(s'))$ can be considered as the ground-truth label y to mimic a supervised learning setting, where r and s' are the received reward and the next state when taking action a in state s. The lifelong learning scenario deals with an infinite sequence of tasks $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \ldots,]$ where each task $\mathcal{D}_t$ is associated with a batch of transitions $\mathcal{T}_t = \sum_i (s_i, a_i, r_i, s_i')$. The tasks change over the lifetime, leading to a nonstationary task distribution, and the current task identity at each time period t is unknown to the learner. The learner has to perform all tasks in the sequence. The full objective is thereby given as to minimize the unbiased sum of losses among all tasks as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\phi}) &= \mathbb{E}_{\mathcal{D}_t \sim \mathcal{D}}\Big[\mathbb{E}_{x,y\sim\mathcal{D}_t}\Big[\big(\hat{y} - y\big)^2\Big]\Big] \\ &= \mathbb{E}_{\mathcal{D}_t \sim \mathcal{D}}\Big[\mathbb{E}_{s,a,r,s'\sim\mathcal{D}_t}\Big[\big(r + \gamma Q_{\boldsymbol{\varphi}}\big(s', \mu_{\boldsymbol{\phi}}\big(s'\big)\big) - Q_{\boldsymbol{\varphi}}(s,a)\big)^2\Big]\Big]\end{aligned} \quad (8)$$

which is equivalent to minimizing the Bellman residual over the given transitions. While being trained for the task at time period t, the learner is fed with samples only from task $\mathcal{D}_t$.

In real-world applications, tasks might correspond to customized requirements, user preferences, unknown dynamics of the system, or other unexpected perturbations. This problem formulation involves a wide variety of RL challenges requiring lifelong adaptation to sequential tasks and balance between plasticity and stability. Throughout the lifetime, the learner needs to continually build upon previously learned knowledge to facilitate optimizing the policy of the current task at hand, in conjunction with accommodating the acquired new information for future learning.

B. Modeling Latent Task Structure

Changing circumstances and unpredictable perturbations are quite common in real-world scenarios, resulting in heterogeneous task distributions. Assuming a single model for lifelong RL is not suitable because it is unlikely to adequately adapt the learner to various tasks using only a few gradient steps. Expansion-based lifelong learning approaches follow the idea that if we learn a new task with new parameters and keep previous parameters unchanged, we can well preserve the knowledge of previous tasks. A straightforward way is to train and store separate parameters for each task [15], while it is rather restricted to ideal settings with explicit task boundaries. Moreover, it quickly suffers from the lack of scalability as the number of encountered tasks increases. In contrast, capturing task relatedness is promising to enhance the stability of past memories by modulating both positive and negative transfer, and to promote plasticity by recognizing outlier tasks that require a more significant degree of adaptation [28], [29]. Therefore, we begin with a more rational idea that clusters previous tasks into a mixture of Bayesian models with an appropriate notion of task relatedness, reducing redundancy within past memories.

For handling the task variability, we assume that the parameters of each task model are drawn from a Dirichlet process mixture of base distributions $\{\boldsymbol{\theta}^{(l)}\}_{l=1}^{L}$, where $\boldsymbol{\theta}^{(l)}$ denote the model parameters corresponding to the lth cluster. Then, we estimate the relatedness across tasks by calculating the likelihood of task-to-cluster assignments, which equals to clustering the task models in a latent space. Each task cluster is initialized by some prior parameters $\boldsymbol{\theta}^{-}$, which is learned by the domain randomization technique and will be introduced in more detail in Section III-D. The prior distribution $P(\boldsymbol{\theta})$ over the mixture of task models is formulated as a CRP that allows for some probability of instantiating a new cluster as needed. Without knowing the number of task clusters, we start with a single mixture component and initialize this task model from $\boldsymbol{\theta}^{-}$. From here, we continually maintain and update the mixture model to handle the lifelong task distribution, and instantiate new mixture components as required using the CRP.

Existing approaches usually rely on awareness of explicit task identities, which is unlikely to hold in real-world applications. Instead, we use the mixture model to estimate the prior and posterior distributions over task clusters, which are utilized to predict task identifies and update parameters of all task models. This results in a scalable lifelong RL method that jointly learns task-to-cluster assignments and model parameters, which can efficiently modulate the task transferability by clustering task models in a latent space. The main idea of our method is illustrated as in Fig. 1.

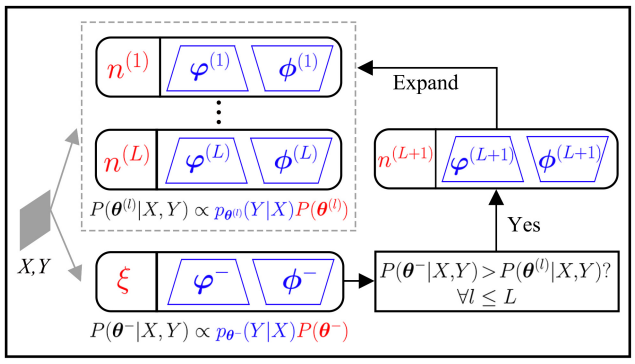


Fig. 1. Overview of our method. $n^{(l)}$ is the assigned task count per cluster, and ξ is a constant that regulates the instantiation of new clusters.

C. Dirichlet Process Mixture for Modeling Task Distribution

Let the categorical latent variable z denote the cluster assignment of task model parameters $\boldsymbol{\theta}$. Since z is unknown, we ought to infer the posterior task-to-cluster assignments $P(z|X, Y)$, where $(X, Y) = \sum_i (x_i, y_i)$ is the dataset constructed from a batch of transitions at a given time period. Moreover, we cannot know the total amount of task clusters in advance. Hence, we propose to employ a Bayesian nonparametric framework, specifically the Dirichlet process mixture model (DPMM), to formulate the nonstationary task distribution into a flexible structure where task clusters are dynamically established and expanded throughout lifelong learning. The instantiation of DPMM is depicted by the CRP that is well suitable for lifelong learning. The CRP is a discrete-time stochastic process analogous to seating an infinite sequence of customers at tables in a Chinese restaurant, where each table represents a distinct cluster. Each customer chooses a preexisting table with a probability proportional to the count of customers already seated there, or sits down alone at a new empty table with a probability proportional to a preset concentration parameter.

For an infinite sequence of tasks $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \ldots,]$ the first task is allocated to the nominal cluster since the number of task clusters is unknown. At time period t, assume that the accumulated knowledge from previous time periods $1, 2, \ldots, t-1$ is accommodated as a mixture of L task clusters $\{\boldsymbol{\theta}_t^{(l)}\}_{l=1}^{L}$. Then, the prior distribution of cluster assignments for the current task is given by

$$P\left(\boldsymbol{\theta}_t^{(l)}\right) = P(z_t = l) = \begin{cases} \frac{n^{(l)}}{t-1+\xi}, & l \leq L \\ \frac{\xi}{t-1+\xi}, & l = L+1 \end{cases} \tag{9}$$

where $n^{(l)}$ is the expected number of tackled tasks that have occupied the lth cluster, and ξ is a constant positive concentration parameter for regulating the new cluster instantiation. $l \leq L$ indicates assigning the current task to an existing cluster, and $l = L + 1$ implies the potential spawning of a new task cluster into the mixture model. Taking all history periods into account, the prior probabilities over task clusters become

$$P\left(\boldsymbol{\theta}_t^{(l)}|\boldsymbol{\theta}_{1:t-1}, \xi\right) = \begin{cases} \frac{\sum_{t'=1}^{t-1} P\left(\boldsymbol{\theta}_{t'}^{(l)}\right)}{t-1+\xi}, & l \leq L \\ \frac{\xi}{t-1+\xi}, & l = L+1. \end{cases} \tag{10}$$

This nonparametric formulation fits the mixture distribution without a constant number of components, allowing the mixture to dynamically adapt its cluster structure to the increased complexity of the lifelong learning process.

It may become an intractable combinatorial optimization problem to directly maximize the expected likelihood of the mixture model. We need a scalable approximation that can represent the conditional distribution of the latent variable with maximum *a posteriori* (MAP) estimation. Hence, we employ an EM procedure to update the mixture of task-specific parameters in an online manner, without access to samples from previous tasks. Here, the E-step in EM computes the posterior expectation of task-to-cluster assignments, that is, estimating the conditional mode of task-specific parameters, and the M-step involves updating parameters of all task models for future learning.

Let $p_{\boldsymbol{\theta}}(Y|X)$ denote the predictive-likelihood function regarding the task model $\boldsymbol{\theta}$ on a batch of samples (X, Y), that is, $P(Y|X, \boldsymbol{\theta})$. The predictive function treats each sample as an independent Gaussian $\mathcal{N}(y_i; \hat{y}_{\boldsymbol{\theta}}(x_i), \sigma^2)$ as

$$\begin{aligned} p_{\boldsymbol{\theta}}(Y|X) &= \Pi_i \mathcal{N}\left(y_i; \hat{y}_i, \sigma^2\right) \\ &= \Pi_i \mathcal{N}\left(r_i + \gamma Q_{\boldsymbol{\varphi}}\left(s_i', \mu_{\boldsymbol{\phi}}\left(s_i'\right)\right); Q_{\boldsymbol{\varphi}}(s_i, a_i), \sigma^2\right) \end{aligned} \tag{11}$$

where $r_i \sim r(s_i, a_i)$ and σ^2 is a constant. First, we estimate the expectation over preexisting task clusters and the potential new one. The posterior probability of task-to-cluster assignment $P(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t)$ is calculated by the Bayes rule as

$$P\left(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t\right) = \frac{P\left(Y_t|X_t\boldsymbol{\theta}_t^{(l)}\right)P\left(X_t|\boldsymbol{\theta}_t^{(l)}\right)P\left(\boldsymbol{\theta}_t^{(l)}\right)}{P(X_t, Y_t)}. \tag{12}$$

We assume that the input marginal likelihood $P(X_t|\boldsymbol{\theta}_t^{(l)})$ is approximately invariant across tasks and can be neglected. Then, the posterior can be roughly approximated by

$$P\left(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t\right) \propto p_{\boldsymbol{\theta}_t^{(l)}}(Y_t|X_t)P\left(\boldsymbol{\theta}_t^{(l)}\right). \tag{13}$$

Combining the predictive likelihood in (11) and the CRP prior distribution in (10), we perform the E-step to compute the posterior probabilities of task-to-cluster assignments as

$$P\left(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t\right) \propto \begin{cases} p_{\boldsymbol{\theta}_t^{(l)}}(Y_t|X_t) \sum_{t'=1}^{t-1} P\left(\boldsymbol{\theta}_{t'}^{(l)}\right), & l \leq L \\ p_{\boldsymbol{\theta}_t^{(l)}}(Y_t|X_t)\xi, & l = L+1. \end{cases} \tag{14}$$

With the estimated posterior task-to-cluster assignments, we turn to the M-step to maximize the expected log likelihood of the mixture model as

$$\mathcal{L}(\boldsymbol{\theta}_t) = \mathbb{E}_{\boldsymbol{\theta}_t \sim P(\boldsymbol{\theta}_t|X_t, Y_t)}\left[\log p_{\boldsymbol{\theta}_t}(Y_t|X_t)\right]. \tag{15}$$

Supposing that each task model begins with some prior parameters $\boldsymbol{\theta}^{-}$, the value of $\boldsymbol{\theta}_t$ after taking all history gradient updates is calculated as

$$\begin{aligned} \boldsymbol{\theta}_{t+1}^{(l)} &= \boldsymbol{\theta}_1^{(l)} \\ &+ \alpha \sum_{t'=1}^{t} P\left(\boldsymbol{\theta}_{t'}^{(l)}|X_{t'}, Y_{t'}\right) \nabla_{\boldsymbol{\theta}_{t'}^{(l)}} \log p_{\boldsymbol{\theta}_{t'}^{(l)}}(Y_{t'}|X_{t'}) \ \forall l \end{aligned} \tag{16}$$

Algorithm 1: Scalable Lifelong RL With A Dirichlet Process Mixture

Input: Task sequence $\mathcal{D} = [\mathcal{D}_1, ..., \mathcal{D}_{t-1}, \mathcal{D}_t, ...]$, robust prior parameters $\boldsymbol{\theta}^-$

Output: Optimal model parameters $\boldsymbol{\theta}_t^*$

1 Initialize $L = 1, t = 1, l^* = 1$, and $\boldsymbol{\theta}_1^{(1)} \leftarrow \boldsymbol{\theta}^-$
2 **for** *each time period t* **do**
3 Initialize $\boldsymbol{\theta}_t^{(L+1)} \leftarrow \boldsymbol{\theta}^-$
4 Receive a batch of transitions $\mathcal{T}_t = \sum_i (s_i, a_i, r_i, s_i')$
5 Construct (X_t, Y_t) from $\mathcal{T}_t$
6 Calculate $p_{\boldsymbol{\theta}_t^{(l)}}(Y_t|X_t)$ using (11), $\forall l \le L+1$
7 Infer $P(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t)$ using (14), $\forall l \le L+1$
8 **if** $P(\boldsymbol{\theta}_t^{(L+1)}|X_t, Y_t) > P(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t), \forall l \le L$ **then**
9 Add $\boldsymbol{\theta}_t^{(L+1)}$ to $\boldsymbol{\theta}_t$ thereafter
10 $L \leftarrow L+1$
11 **end**
12 **while** *not terminated* **do**
13 E-step, re-calculate $P(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t)$ using (13) with updated $\boldsymbol{\theta}_t^{(l)}$, $\forall l \le L$
14 M-step, adapt $\boldsymbol{\theta}_t^{(l)}$ using (17) with updated $P(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t)$, $\forall l \le L$
15 **end**
16 $\boldsymbol{\theta}_{t+1}^{(l)} \leftarrow \boldsymbol{\theta}_t^{(l)}$, $\forall l \le L$
17 $l^* = \arg\max_{l \le L} p_{\boldsymbol{\theta}_{t+1}^{(l)}}(Y_t|X_t)$
18 **end**

where α is the learning rate. In the lifelong learning setting, all model parameters are updated at each time period. Hence, we can perform the M-step in (16) by simply updating model parameters at the previous time period $\boldsymbol{\theta}_{t-1}$ on the newest samples as

$$\boldsymbol{\theta}_{t+1}^{(l)} = \boldsymbol{\theta}_t^{(l)} + \alpha P\left(\boldsymbol{\theta}_t^{(l)}|X_t, Y_t\right) \nabla_{\boldsymbol{\theta}_t^{(l)}} \log p_{\boldsymbol{\theta}_t^{(l)}}(Y_t|X_t) \quad \forall l. \tag{17}$$

This formation removes the requirement for memorizing samples of previous tasks, yielding a practical lifelong RL algorithm that tackles a continual stream of data. In addition, the E- and M-steps are iteratively alternated to converge to fully implement the EM algorithm.

We summarize the proposed lifelong RL algorithm and outline it in Algorithm 1. At the nominal time period, we initialize the mixture model that contains one entry $\boldsymbol{\theta}_1^{(1)} \leftarrow \boldsymbol{\theta}^-$ in line 1. From there, at each time step t, we first initialize an empty task model from the prior parameters $\boldsymbol{\theta}^-$ in line 3. Then, we collect a batch of transitions in line 4 and construct the input–output samples in line 5. Next, we calculate the predictive likelihood of all task models in line 6, and infer the posterior task-to-cluster assignments in line 7. The CRP prior allows some probability for instantiating a new cluster to the mixture distribution, while the posterior task-to-cluster assignments determine the expansion of the new task cluster into the mixture model. In lines 8–11, the new potential cluster is added to the mixture if its posterior probability is larger than those of the L preexisting clusters. Then, we keep alternating the E- and M-steps until the learning is terminated in lines 12–15,[1] and obtain the updated model parameters in line 16. Using updated model parameters $\boldsymbol{\theta}_{t+1}^{(l)}$, the next batch of transitions is predicted according to the most likely task l^* in line 17.

D. Robust Prior via Domain Randomization

We formulate a mixture of task models for performing lifelong learning adaptation in the face of an infinite stream of incoming data. New task models are instantiated as needed under the Bayesian inference framework, where parameters of each new task model are initialized from $\boldsymbol{\theta}^-$. However, modern parametric models, for example, DNNs, are usually hard to train in such a lifelong learning setting. They typically require numerous iterations with plenty of training samples to learn a sensible solution, which can be infeasible when faced with lifelong streaming information. Therefore, we employ the *domain randomization* approach [30], [31] to train the prior parameters $\boldsymbol{\theta}^-$ for each task model. Domain randomization is originally proposed to learn control policies *robust* to the transfer from simulation to reality, that is, "sim-to-real," by randomizing various aspects of the simulated environment at training time. We adopt this technique to learn robust model initialization that can generalize well to nonstationary task distributions during lifelong learning.

In contrast to learning a policy for one particular task, we train a model $\boldsymbol{\theta}^-$ that is capable of tackling a diversity of tasks. In the lifelong learning setting, we collect samples from a finite number of tasks $\mathcal{D}^- = [\mathcal{D}_1, \ldots, \mathcal{D}_m]$, and use all these samples to train a robust model prior. The objective is then modified to minimize the expected loss, that is, the Bellman residual, across a distribution of tasks as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}^-) &= \mathbb{E}_{\mathcal{D}_i \sim \mathcal{D}^-}\left[\mathbb{E}_{x,y \sim \mathcal{D}_i}\left[\left(\hat{y} - y\right)^2\right]\right] \\ &= \mathbb{E}_{\mathcal{D}_i \sim \mathcal{D}^-}\left[\mathbb{E}_{s,a,r,s' \sim \mathcal{D}_i}\left[\left(r + \gamma Q_{\boldsymbol{\varphi}^-}\left(s', \mu_{\boldsymbol{\phi}^-}(s')\right) - Q_{\boldsymbol{\varphi}^-}(s,a)\right)^2\right]\right].\end{aligned} \tag{18}$$

By training the model to adapt to variability in the nonstationary task distribution, the resulting model is supposed to better generalize to unseen tasks. After a new task model is instantiated from the prior parameters $\boldsymbol{\theta}^-$, it might then better adapt to any task using only a few gradient steps.

IV. Experiments

Experiments are conducted on a suite of continuous control tasks to show the applicability and scalability of our method in lifelong learning settings. Using agents in these tasks, we create a variety of representative RL problems with nonstationary task distributions, where scalable lifelong learning is crucial. The following two sections show empirical results and corresponding insights on the experimentation. We compare our method to several baseline methods.

1) *Fine-Tune:* As a representative dynamic evaluation baseline in [32] and [33], it continually trains a single base model as the steaming data enters.

[1]Empirically, the learning is terminated when the change of the neural network weights $\boldsymbol{\theta}$ is smaller than a preset tiny threshold.

2) *Reservoir:* Simple experience replay with reservoir sampling can be a powerful tool in lifelong learning [11], [34]. It uniformly selects a batch of samples from an infinite data stream, which is well suited to manage the replay memory without explicit task boundaries.
3) *Consolidation:* The policy consolidation [14] method uses a cascade of hidden networks to simultaneously remember policies at a range of timescales and regularize the current policy by its own history [35]. Since it does not require knowledge of task boundaries, we evaluate this regularization-based method for comparison.
4) *Progressive:* We evaluate the progressive neural network [15] as a classical expansion-based baseline for lifelong learning, which freezes the previous network and allocates new subnetworks to accommodate new information [20]. Note that it requires explicit task boundaries and labels.

We use the DDPG algorithm to handle continuous control tasks, where the actor maps a given state to an estimated optimal action and the critic approximates the action-value function. Both the actor and critic are represented by a neural network containing two 512-node hidden layers with ReLU activation, and their parameters are optimized by gradient descent. To promote good stability [24], we utilize the soft updating strategy to update weights of target networks. For Reservoir, the model is fed with two minibatches of the same size at each learning iteration: one from the current data stream and the other from the long-term replay memory. For Consolidation, the model consists of four sets of hidden networks in addition to the visible one that is amenable for the current policy. The hyperparameters are set as: learning rate $\alpha = 0.001$, discounting factor $\gamma = 0.99$, and batch size for network updating $n = 64$.

We define two performance metrics for each evaluation unit, that is, a given tested approach running on a given task. One is the return of one learning episode that is associated with the learned policy, defined as $\sum_{i=1}^{H} r(s_i, \mu_{\boldsymbol{\phi}}(s_i))$, where H is the time horizon of the learning episode. The other is the average return over all learning episodes, defined as $\frac{1}{J}\sum_{j=1}^{J}\sum_{i=1}^{H} r(s_i^j, \mu_{\boldsymbol{\phi}}(s_i^j))$, where J is the number of learning episodes. The former will be plotted in figures and the latter will be presented in tables. We continually change the learning task at random for $T = 50$ times to create a lifelong learning process with a nonstationary task distribution $\mathcal{D} = [\mathcal{D}_1, \ldots, \mathcal{D}_T]$. We report the performance of all tested approaches for each task instance $\mathcal{D}_t (1 \le t \le T)$, and record the statistical outcome over all encountered tasks to demonstrate the capability of lifelong learning. Our code is available online.[2]

A. Simple 2-D Navigation

As an explanatory experiment, we implement a simple 2-D navigation task in a continuous state–action space to serve as a proof of principle and test if our method achieves both plasticity and stability in lifelong learning scenarios. As shown in Fig. 2, the task is to move a point agent to a goal position within a unit square. The state is the agent's current position in the 2-D coordination system. The action is a 2-D velocity vector and is clipped to the range of $[-0.1, 0.1]$. The reward function is set as the negative Euclidean distance to the goal position minus a minor control cost proportional to the action magnitude. Each learning episode begins with a fixed initial state, and terminates when the point agent reaches the region within 0.01 of the goal position or the time step comes to the horizon of $H = 100$. During lifelong learning, the goal position may change over time within the unit square at random, resulting in a nonstationary task distribution.

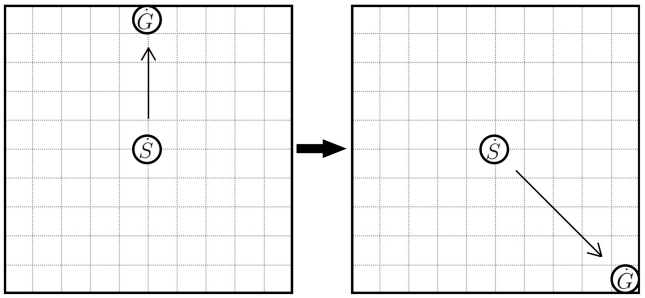


Fig. 2. Simple 2-D navigation task in a lifelong learning setting where the goal may change over time. S is the starting point and G is the goal.

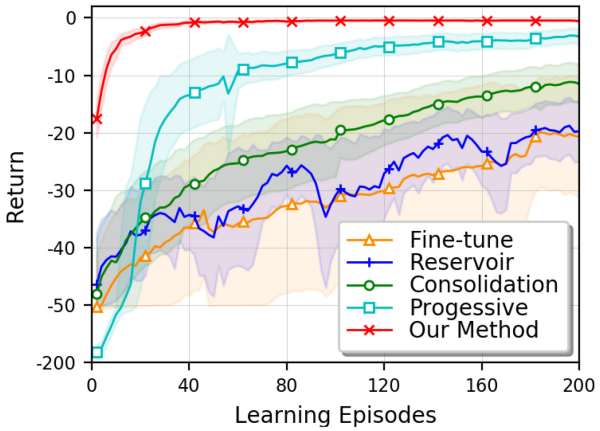


Fig. 3. Return per learning episode of baselines and our method in the 2-D navigation task. Here and in similar figures below, the bold line depicts the mean of received return per episode over $T = 50$ sequential tasks, and the shaded plots 95% bootstrapped confidence intervals of the mean.

We first show main results of our method and baseline methods implemented on the 2-D navigation task. For our method, $L = 4$ task clusters are instantiated totally. Fig. 3 presents the received return per learning episode, and Table I reports the numerical average return over 200 learning episodes. Fine-tune obtains the worst lifelong learning performance under a nonstationary task distribution, since it adopts the simplest learning adaptation mechanism. Reservoir achieves slightly higher average return than fine-tune, while its received return per learning episode tends to oscillate more during lifelong learning. We conjecture that replaying samples from other tasks can impose some interference on the online updates when learning a new task. Consolidation obtains better performance than Reservoir, indicating that regularizing the current policy

[2]https://github.com/HeyuanMingong/sllrl

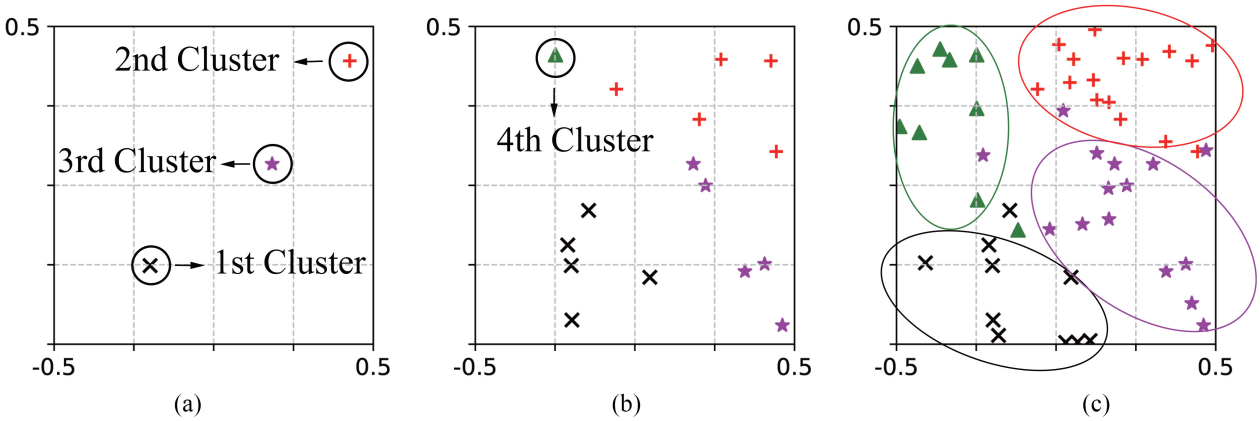


Fig. 4. Visualization of the Bayesian mixture during lifelong learning. (a) Initial three clusters are instantiated at the initial three time steps. (b) Fourth cluster is instantiated at time step $t = 16$. (c) All the $T = 50$ tasks are clustered into four mixture components effectively.

TABLE I
NUMERICAL AVERAGE RETURN OVER 200 EPISODES OF BASELINES AND OUR METHOD ON THE 2-D NAVIGATION TASK. HERE AND IN SIMILAR TABLES BELOW, WE PRESENT THE MEAN OVER $T = 50$ SEQUENTIAL TASKS AND CORRESPONDING STANDARD ERRORS. WE MARK THE BEST PERFORMANCE IN BOLDFACE

Methods	Return
Fine-tune	-31.73 ± 0.51
Reservoir	-28.78 ± 0.48
Consolidation	-22.01 ± 0.64
Progressive	-15.47 ± 2.07
Our Method	$\mathbf{-1.23 \pm 0.17}$

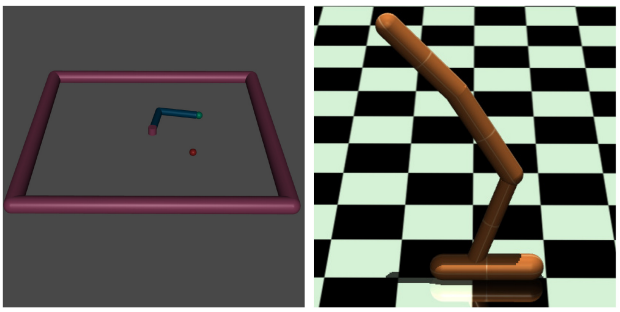
by its own history to force it less overfitted to the task at hand can alleviate catastrophic interference to some extent. Progressive performs the best among the baseline methods, which is supposed to benefit from maintaining stability by blocking changes to the previous network and promoting plasticity by allocating new subnetworks to accommodate new knowledge.

In contrast, it can be observed from Fig. 3 that our method achieves much faster learning adaptation to the nonstationary task distribution compared to all baseline methods. Our method takes only 20 learning episodes to obtain near-optimal asymptotic performance for a given task during lifelong learning, while it takes far more than 100 episodes for all baseline methods to achieve comparable performance. Table I illustrates that our method obtains remarkably greater average return over all learning episodes than all baselines. Based on explicitly estimating task relatedness, our method is capable of enhancing stability by modulating transferability across tasks and promoting plasticity by recognizing outlier tasks that require a more significant degree of adaptation. Moreover, statistical results show that our method achieves narrower confidence intervals and smaller standard errors than all baselines, indicating that our method enables more stable lifelong learning adaptation to a changing distribution of tasks.

Furthermore, to test whether our method correctly estimates task relatedness and clusters encountered tasks in a latent space, we gain an intuition of the Bayesian mixture via visualization to observe and comprehend the lifelong learning process. Each task is characterized by the reward function associated with its goal position. Tasks with adjacent goals reveal higher similarity and are likely to be assigned to the same mixture component. Therefore, we employ the goal position in the 2-D coordinate system as a visualization to measure relatedness between tasks. As illustrated in Fig. 4, each data point within the unit square represents a goal position associated with a particular task $\mathcal{D}_t (1 \leq t \leq T)$. Tasks belonging to different clusters in the mixture are depicted by data points with different colors and shapes. We can observe that the four task clusters are expanded into the mixture model at time steps $t = 1, 2, 3, 16$ incrementally. During the entire lifelong learning process, the tasks under a nonstationary distribution over $T = 50$ time steps are consecutively clustered as four mixture components in a latent space, as visualized in Fig. 4(c). It successfully verifies that our method is capable of clustering tasks from a nonstationary distribution in a latent space where similar tasks are closely spaced and tend to be assigned to the same cluster. This is crucial for a scalable lifelong RL algorithm since correctly estimating task relatedness is the prerequisite for modulating transferability across tasks. At each time step, the task at hand is allocated to a preexisting cluster or expanded as a new component in the mixture according to the nonparametric Bayesian framework. This nonparametric formulation fits the mixture distribution without *a priori* fixed number of components and without any external information to signal task boundaries in advance, which is critical for scalable lifelong learning in real world.

B. MuJoCo Locomotion

The results in the simple 2-D navigation domain demonstrate that our method facilitates scalable lifelong RL with good balance between stability and plasticity. Next, we test whether similar benefits can be obtained for lifelong learning when our method is applied to more sophisticated deep RL problems at the scale of DNNs. In the next set of experiments, we investigate two kinds of high-dimensional continuous control problems based on the MuJoCo physics engine [23].



(a) (b)

Fig. 5. Two kinds of MuJoCo locomotion tasks. (a) Reacher. (b) Hopper.

TABLE II
NUMERICAL AVERAGE RETURN OVER 200 LEARNING EPISODES OF BASELINES AND OUR METHOD IMPLEMENTED ON THE TWO KINDS OF MUJOCO LOCOMOTION DOMAINS

Methods	Reacher	Hopper
Fine-tune	-8.58 ± 0.20	-4.56 ± 0.08
Reservoir	-4.22 ± 0.07	-4.71 ± 0.02
Consolidation	-9.18 ± 0.19	-4.36 ± 0.08
Progressive	-4.30 ± 0.18	-2.62 ± 0.15
Our Method	$\mathbf{-2.63 \pm 0.11}$	$\mathbf{0.05 \pm 0.11}$

As shown in Fig. 5(a), one is the Reacher domain that aims to move a two-joint torque-controlled robot arm to a particular target point. The reward function is set as the negative Euclidean distance between the fingertip and the target location, minus a minor control penalty proportional to the scale of action. Each learning episode begins with a fixed initial state, and terminates when the fingertip reaches the region within 0.001 of the target location or the time step comes to the horizon of $H = 100$. The lifelong task distribution is created by changing the target point within the reachable circle at random. As shown in Fig. 5(b), the other is the Hopper domain that requires a one-legged hopper robot to run forward at a given velocity along the x-axis. The reward function is set as the negative absolute difference between the current velocity of the robot and a goal one, plus an alive bonus. Each learning episode terminates when the robot falls down or the time step comes to the horizon of $H = 100$. We consecutively change the goal velocity randomly within the range of [0.0, 1.0], resulting in a nonstationary task distribution.

We show primary results of baselines and our method implemented on the two kinds of locomotion domains. Fig. 6 illustrates the received return per episode, and Table II presents corresponding numerical average return over 200 learning episodes. Fine-tune and Consolidation obtain similar performance in both domains. Reservoir can receive high returns at the early learning stage, while it tends to achieve suboptimal performance later on since replaying old samples of previous tasks may interfere with learning the new task. Progressive also performs better than other baselines, demonstrating the effectiveness and superiority of expansion-based approaches for lifelong learning. Our method usually achieves significantly more rapid and stable lifelong learning adaptation compared to baseline approaches. By comparison, it consumes

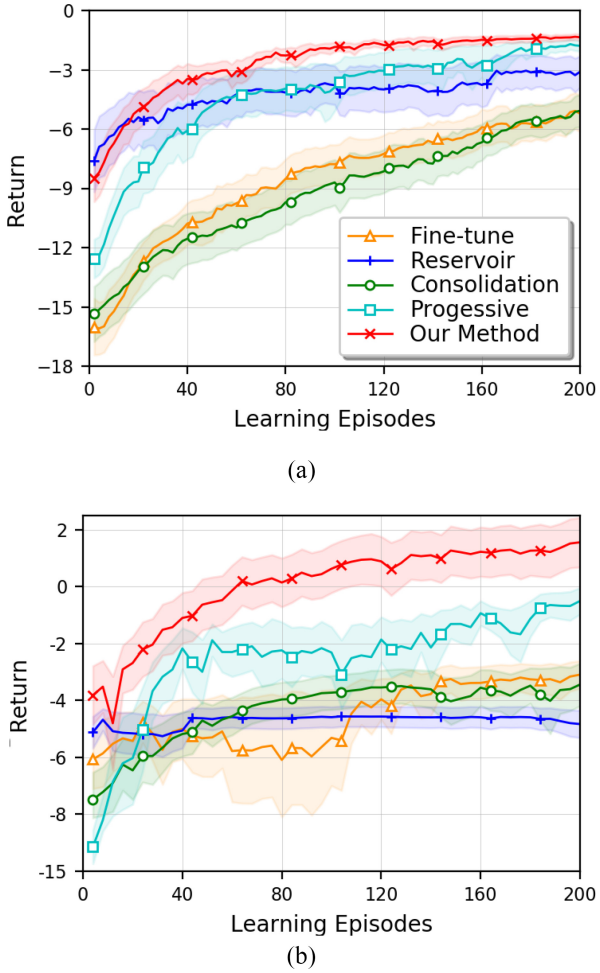


Fig. 6. Received return per learning episode of baselines and our method in the two kinds of MuJoCo locomotion domains. (a) Reacher. (b) Hopper.

significantly more computation cost for baseline approaches to achieve comparable performance to our method. For example, in the Hopper domain, our method only needs approximately 40 learning episodes to achieve a near-optimal return, while all baselines can cost far more than 200 episodes.

The results reveal that our method effectively builds on previously learned knowledge to improve learning adaptation to new tasks throughout the lifetime. Governed by the Bayesian nonparametric framework, the task identity at each time period is automatically detected by MAP estimation. Subsequently, our method retrieves the most similar experience from the mixture of robust task models (including the potential new cluster), which is supposed to benefit the new task at hand most. By modulating transferability across tasks, our method only requires to "fine-tune" the selected prior experience a little bit using a small quantity of computational efforts, being significantly more efficient for lifelong learning adaptation to nonstationary task distributions.

C. Ablation Study

To identify the respective contribution of the two components to the overall performance, that is: 1) DPMM and 2) domain randomization, we conduct an ablation study to

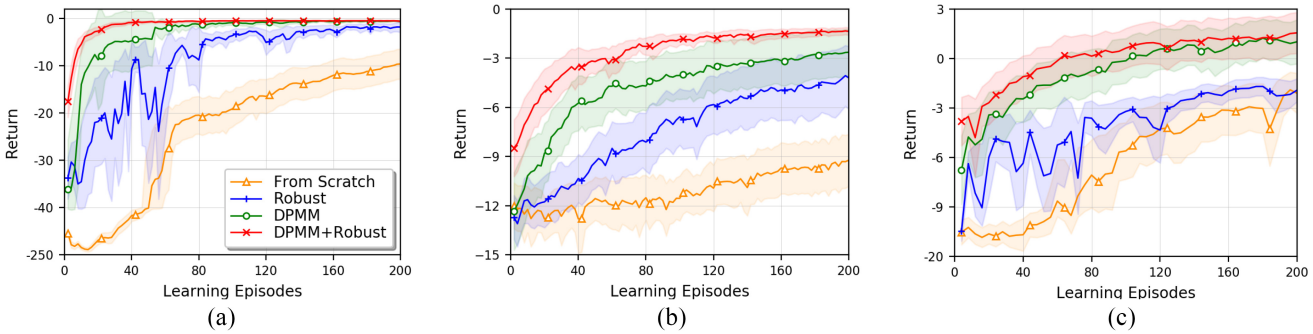


Fig. 7. Received return per learning episode of the ablation study on all domains. (a) Navigation. (b) Reacher. (c) Hopper.

TABLE III
NUMERICAL AVERAGE RETURN OVER 200 LEARNING EPISODES OF THE ABLATION STUDY ON ALL DOMAINS

Methods	Navigation	Reacher	Hopper
From Scratch	-49.79 ± 4.61	-11.12 ± 0.08	-7.50 ± 0.33
Robust	-9.08 ± 0.66	-7.45 ± 0.18	-4.14 ± 0.18
DPMM	-3.37 ± 0.46	-4.78 ± 0.16	-0.75 ± 0.14
DPMM+Robust	$\mathbf{-1.23 \pm 0.17}$	$\mathbf{-2.63 \pm 0.11}$	$\mathbf{0.05 \pm 0.11}$

separate the two components apart for observation on both the Navigation and MuJoCo domains. During lifelong learning, we implement four variants of our method as follows.

1) *From Scratch:* Without component used, it learns each task from scratch, providing a lower bound to show the benefits of lifelong transfer in general.
2) *Robust:* We employ domain randomization to train a robust prior, and learn each task using DDPG with parameters initialized from that prior.
3) *DPMM:* The domain randomization component is ablated from our method.
4) *DPMM+Robust:* Both components are used.

Fig. 7 shows the received return per episode, and Table III presents numerical average return over 200 learning episodes.

First, we identify how DPMM affects the lifelong learning performance by comparing From Scratch with DPMM, and by comparing Robust with DPMM+Robust. It is observed that DPMM and DPMM+Robust can largely improve the performance of From Scratch and Robust, respectively, which verifies the significant effectiveness of our DPMM component. With formulating the nonstationary task distribution with an increasing number of clusters, our DPMM component provides a flexible structure for modulating transferability across tasks and accommodating new knowledge as needed.

Next, we test the capability of the adopted domain randomization technique. Comparing From Scratch with Robust, we can observe that domain randomization is capable of improving learning performance to a large extent. It validates the promising efficiency of domain randomization for learning robust model initialization that can generalize well to nonstationary task distributions during lifelong learning. Comparing DPMM with DPMM+Robust, it is observed that DPMM can achieve a moderate performance improvement with the help of domain randomization, as the promotion only occurs when a new task cluster is expanded into the mixture.

Finally, we compare all the four variants. DPMM is the crucial component of our method, in that removing this component can cause a large drop in learning performance. DPMM better facilitates learning performance than domain randomization, and combining the two components jointly results in the best lifelong learning adaptation to the nonstationary task distribution.

V. RELATED WORK

Lifelong learning considers learning multiple tasks in sequence, which needs to retain previously learned knowledge and leverage that knowledge to facilitate learning new skills [36]. Various configurations in the literature are related to lifelong RL. Multitask RL [37] aims to optimize the overall performance of all tasks, which needs a reservoir of persistent training samples for all tasks. Transfer RL [38], [39] assumes the simultaneous availability of multiple source tasks and concentrates on facilitating the performance of a particular target task. Meta-RL [33], [40], also called as few-shot RL, learns a base model (i.e., the meta) that can quickly adapt to new tasks, while not considering the alleviation of catastrophic forgetting or interference.

A variety of approaches has been investigated to tackle catastrophic forgetting or interference in the machine learning community. These can be classified into three major categories according to how the knowledge of previous tasks is memorized and leveraged: 1) replay based; 2) regularization based; and 3) expansion based.

Replay-based approaches use the idea of episodic memory, where examples from prior tasks are stored to recall experiences encountered in the past. While storing past examples for rehearsal can date back to 1990s [41], it yields decent results against catastrophic forgetting in practical problems. Rolnick *et al.* [11] leveraged off-policy learning from replay experiences to enhance stability, and used behavior cloning to keep the policy distribution close to historical data. Isele and Cosgun [42] proposed a rank-based method for the online collection and preservation of training experiences in a long-term memory to reduce the effects of forgetting. Instead of storing training examples, Lopez-Paz and Ranzato [36] stored gradients of previous tasks, such that at any time the

gradients of all tasks except the current one can be used to form a trust region that prevents forgetting. An inherent drawback is the constraint on the memory capacity as the number of encountered tasks grows, which could limit its application to large-scale problems. To avoid storing past examples, Shin *et al.* [12] sampled synthetic data from a generative model, shifting the problem to the training of this generative model. However, the generative model used to mimic older parts of the data distribution can also suffer from catastrophic forgetting [14]. Furthermore, the policy trained using experiences from an enormous range of domains may learn a conservative strategy or fail to learn the task [43].

Regularization-based approaches are typically inspired by theoretical neuroscience suggesting that synapses with different levels of plasticity can protect consolidated knowledge from forgetting [44]. From a computational perspective, additional regularization terms are imposed on the learning objective, aiming to identify the important weights of previous tasks and penalize large updates on those weights when learning a new task. Elastic weight consolidation (EWC) [13] slowed down the learning for weights relevant to the knowledge of previous tasks by adding a quadratic penalty on the difference between parameters of the old and new tasks weighted by the Fisher information matrix. Similar to EWC, [45] maintained an online estimate of the synapse's importance regarding past tasks and penalized changes to the most relevant synapses, such that new tasks are trained with minimal forgetting. Schwarz *et al.* [17] used a modified version of EWC to mitigate forgetting when distilling the newly learned behavior into the knowledge base. Kaplanis *et al.* [14], [35] proposed a cascade of hidden networks that simultaneously remember policies at a range of timescales and regularized the current policy by its own history, thereby improving its ability to learn without forgetting. In general, with limited neural resources, comprising additional regularization terms may lead to a tradeoff on the accomplishment of old and new tasks [1].

On the other hand, expansion-based approaches incrementally expand new architectural resources, for example, a policy/option library or the network capacity, in response to new information. Conceptually, such a direction has two superiorities compared with the above two: 1) catastrophic forgetting is mitigated by protecting past memories from being perturbed by the new information and 2) the model capacity is determined adaptively throughout the lifetime. The family of policy reuse algorithms [16], [18], [46], [47] improved its exploration in a new task by probabilistic exploitation of similar policies from a built policy library. Analogously, option reuse approaches [19], [48] summarized prior experience through temporally extended actions (i.e., subpolicies or options) and leveraged only reusable parts of the policy for future learning. Another way is to expand the neural network capacity in the context of deep learning. The simplest example is to freeze early layers and fine-tune later layers when learning the new task [49]. Rusu *et al.* [15], [50] blocked any changes to the network trained on previous tasks and allocated a new subnetwork with fixed capacity to process the new information. Similarly, dynamically expanding network [20] increased the amount of trainable parameters to accommodate new tasks incrementally and used group sparse regularization to decide how many neurons to add at each layer. Parisi *et al.* [51] used self-organizing networks to update connectivity patterns and allocate neural resources dynamically for lifelong learning of human action sequences.

Nevertheless, existing expansion-based approaches usually suffer from the lack of scalability due to two critical limitations: 1) most of them are studied in a rather restricted setting that requires explicit task boundaries and hand-designed heuristics for incorporating new resources and 2) the network size may scale quadratically in the number of encountered tasks. In contrast, we use a Dirichlet process mixture to handle the nonstationary task distribution and automatically infer task identities under the Bayesian nonparametric framework, thereby achieving scalable lifelong RL. The proposed method is an extension of our previous work in [52], which requires an auxiliary set of networks to approximate the reward or state transition function. In this article, we capture task relatedness using Bayesian inference on the Bellman residual, thus introducing only a single set of networks to concurrently train the policy and parameterize the task.

VI. Conclusion and Future Work

In the article, we proposed a scalable lifelong RL method that dynamically expands the network capacity to quickly accommodate new knowledge while stably preserving past memories. The nonstationary task distribution is modeled by a Dirichlet process mixture that clusters the task-specific parameters in a latent space. Governed by the Bayesian nonparametric framework, the mixture is maintained via an EM procedure, in conjunction with a CRP prior, to dynamically adapt the model complexity without explicit task boundaries or hand-designed heuristics. Based on capturing task relatedness by estimating the likelihood of task-to-cluster assignments, our method successfully enhances stability by modulating transferability across tasks, and promotes plasticity by recognizing outlier tasks that require a more significant degree of adaptation. Furthermore, the domain randomization technique is employed to train robust task models for initializing the mixture components, thereby providing better generalization ability when adapting to unseen tasks. Experiments conducted on a suite of continuous control domains verify that our method facilitates scalable lifelong learning performance to nonstationary task distributions.

A few interesting research directions are worth investigating for future work. One is to evaluate our method on different domains, such as Atari games [27] and StarCraft II learning environment [4]. Another is to improve the accuracy of task inference, which is the main bottleneck of our method and could be addressed from several aspects. For example, task relatedness is captured using Bayesian inference on the Bellman residual, where the "pseudo" ground truth relies on the Q-network and will gradually change as the learning proceeds, analogous to the classical DQN algorithm [27]. To better capture task relatedness and modulate

task transferability, we could use powerful variational inference approaches [53] to more accurately approximate posterior distributions of task-to-cluster assignments. For another example, we neglect the input marginal likelihood in (12) for simplifying the posterior derivation. We could employ efficient density estimators, for example, VAE [54], to describe the marginal likelihood for more accurate posterior inference.

REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019.

[2] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001.

[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[4] O. Vinyals *et al.*, "Grandmaster level in StarCraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[5] X. Xu, L. Zuo, L. Qian, J. Ren, and Z. Sun, "A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3884–3897, Oct. 2020.

[6] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019.

[7] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 621–632, Apr. 2019.

[8] Z. Wang, H.-X. Li, and C. Chen, "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2861–2871, Jun. 2020.

[9] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2687–2700, Jun. 2020.

[10] J.-A. Li *et al.*, "Quantum reinforcement learning during human decision-making," *Nat. Human Behav.*, vol. 4, pp. 294–307, Jan. 2020.

[11] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 348–358.

[12] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2994–3003.

[13] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[14] C. Kaplanis, M. Shanahan, and C. Clopath, "Policy consolidation for continual reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3242–3251.

[15] A. A. Rusu *et al.*, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[16] R. Glatt, F. L. Da Silva, R. A. da Costa Bianchi, and A. H. R. Costa, "DECAF: Deep case-based policy inference for knowledge transfer in reinforcement learning," *Expert Syst. Appl.*, vol. 156, Oct. 2020, Art. no. 113420.

[17] J. Schwarz *et al.*, "Progress & compress: A scalable framework for continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4528–4537.

[18] R. Glatt and A. H. R. Costa, "Policy reuse in deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4929–4930.

[19] E. Brunskill and L. Li, "PAC-inspired option discovery in lifelong reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 316–324.

[20] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[21] S. Lee, J. Ha, D. Zhang, and G. Kim, "A neural Dirichlet process mixture model for task-free continual learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[22] M. L. Koga, V. Freire, and A. H. R. Costa, "Stochastic abstract policies: Generalizing knowledge to improve reinforcement learning," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 77–88, Jan. 2015.

[23] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.

[24] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[25] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[26] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[27] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, Jan. 2007.

[29] G. Jerfel, E. Grant, T. Griffiths, and K. A. Heller, "Reconciling meta-learning and continual learning with online mixtures of tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9119–9130.

[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.

[31] F. Muratore, M. Gienger, and J. Peters, "Assessing transferability from simulation to reality for reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1172–1183, Apr. 2021.

[32] B. Krause, E. Kahembwe, I. Murray, and S. Renals, "Dynamic evaluation of neural sequence models," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2766–2775.

[33] A. Nagabandi *et al.*, "Learning to adapt in dynamic, real-world environments through Meta-reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[34] A. Chaudhry *et al.*, "On tiny episodic memories in continual learning," 2019, *arXiv:1902.10486*.

[35] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2497–2506.

[36] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6467–6476.

[37] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with PopArt," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3796–3803.

[38] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource transfer double DQN based on actor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2227–2238, Jun. 2018.

[39] T. Yang *et al.*, "Efficient deep reinforcement learning via adaptive policy transfer," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 3094–3100.

[40] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[41] A. Robins, "Catastrophic forgetting, rehearsal and pseudo-rehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, 1995.

[42] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3302–3309.

[43] W. Yu, C. K. Liu, and G. Turk, "Policy transfer with strategy optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[44] M. K. Benna and S. Fusi, "Computational principles of synaptic memory consolidation," *Nat. Neurosci.*, vol. 19, no. 12, pp. 1697–1708, 2016.

[45] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.

[46] F. Fernández and M. Veloso, "Probabilistic policy reuse in a reinforcement learning agent," in *Proc. Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2006, pp. 720–727.

[47] F. Fernández, J. García, and M. Veloso, "Probabilistic policy reuse for inter-task transfer learning," *Robot. Auton. Syst.*, vol. 58, no. 7, pp. 866–871, 2010.

[48] R. Bonini, F. L. Da Silva, R. Glatt, E. Spina, and A. H. R. Costa, "A framework to discover and reuse object-oriented options in reinforcement learning," in *Proc. Brazil. Conf. Intell. Syst.*, 2018, pp. 109–114.

[49] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.

[50] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Proc. Conf. Robot Learn.*, vol. 78, 2017, pp. 262–270.

[51] G. I. Parisi, J. Tani, C. Weber, and S. Wermter, "Lifelong learning of human actions with deep neural network self-organization," *Neural Netw.*, vol. 96, pp. 137–149, Dec. 2017.

[52] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online Bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 11, 2021, doi: 10.1109/TNNLS.2021.3055499.

[53] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303–1347, 2013.

[54] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014.

Zhi Wang (Member, IEEE) received the B.E. degree in automation from Nanjing University, Nanjing, China, in 2015, and the Ph.D. degree in machine learning from the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, in 2019.

He is currently an Assistant Professor with the School of Management and Engineering, Nanjing University. He had the visiting position with the University of New South Wales, Canberra, ACT, Australia, and holds the visiting position with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include reinforcement learning, machine learning, and robotics.

Chunlin Chen (Senior Member, IEEE) received the B.E. degree in automatic control and the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

He was with the Department of Chemistry, Princeton University, Princeton, NJ, USA, from September 2012 to September 2013. He had visiting positions with the University of New South Wales, Canberra, ACT, Australia, and the City University of Hong Kong, Hong Kong. He is currently a Professor and the Vice Dean of the School of Management and Engineering, Nanjing University, Nanjing, China. His current research interests include machine learning, intelligent control, and quantum control.

Dr. Chen serves as the Chair for the Technical Committee on Quantum Cybernetics, IEEE Systems, Man and Cybernetics Society.

Daoyi Dong (Senior Member, IEEE) received the B.E. degree in automatic control and the Ph.D. degree in engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

He was an Alexander von Humboldt Fellow with AKS, University of Duisburg–Essen, Duisburg, Germany. He was with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China, and Zhejiang University, Hangzhou, China. He had visiting positions with Princeton University, Princeton, NJ, USA; RIKEN, Wako, Japan; and The University of Hong Kong, Hong Kong. He is currently a Scientia Associate Professor with the University of New South Wales, Canberra, ACT, Australia. His research interests include quantum control and machine learning.

Dr. Dong was awarded the ACA Temasek Young Educator Award by the Asian Control Association and was a recipient of the International Collaboration Award and the Australian Postdoctoral Fellowship from the Australian Research Council, and a Humboldt Research Fellowship from the Alexander von Humboldt Foundation of Germany. He is a Member-at-Large, Board of Governors, and was the Associate Vice President for Conferences and Meetings, IEEE Systems, Man and Cybernetics Society. He served as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2015 to 2021. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS.