Skill-Enhanced Reinforcement Learning Acceleration from Demonstrations

Hanping Zhang[1], **Yuhong Guo**[1,2]
[1]School of Computer Science, Carleton University, Ottawa, Canada
[2]Canada CIFAR AI Chair, Amii, Canada
jagzhang@cmail.carleton.ca, yuhong.guo@carleton.ca

Abstract

Learning from Demonstration (LfD) aims to facilitate rapid Reinforcement Learning (RL) by leveraging expert demonstrations to pre-train the RL agent. However, the limited availability of expert demonstration data often hinders its ability to effectively aid downstream RL learning. To address this problem, we propose a novel two-stage method dubbed as Skill-enhanced Reinforcement Learning Acceleration (SeRLA). SeRLA introduces a skill-level adversarial Positive-Unlabeled (PU) learning model to extract useful skill prior knowledge by enabling learning from both limited expert data and general low-cost demonstration data in the offline prior learning stage. Subsequently, it deploys a skill-based soft actor-critic algorithm to leverage this acquired prior knowledge in the downstream online RL stage for efficient training of a skill policy network. Moreover, we develop a simple skill-level data enhancement technique to further alleviate data sparsity and improve both skill prior learning and downstream skill policy training. Our experimental results on multiple standard RL environments show the proposed SeRLA method achieves state-of-the-art performance on accelerating reinforcement learning on downstream tasks, especially in the early learning phase.

1 Introduction

Despite the wide applicability of reinforcement Learning (RL) across robotics [1], video games [2, 3], and large language models [4, 5], a conventional deep RL agent often requires numerous iterative interactions with the environment to learn a useful policy by maximizing the expected discounted cumulative reward [6], resulting in prolonged training periods and limited computational efficiency that can be particularly problematic for large input scales. To overcome this problem, Learning from Demonstration (LfD), also known as imitation learning (IL), has been investigated to accelerate RL. In LfD, the agent is pre-trained on a small offline demonstration dataset provided by human experts [7, 8] to acquire knowledge and learn behaviors that can be executed in the environment, which can then be leveraged to accelerate the online learning process of the downstream RL task with fewer interactions with the environment. Due to the limited availability of expert demonstration data, some recent studies seek to supplement the expert data with a large task-agnostic demonstration dataset collected inexpensively using methods such as autonomous exploration [9, 10] or human-tele operation [11, 12].

As an RL technique that learns reusable skills from a given expert behavior [13, 14] or interacting with online environments [15, 16, 14, 9] to guide the RL training, skill-based RL shows great potential for LfD. Recently, researchers have introduced skill-based RL to LfD by learning reusable skill behaviors from the demonstration data and deploying them for the downstream tasks [11, 17–20]. However, these previous studies either focus solely on learning from the expert dataset or treat the general demonstration data as negative examples, impeding the effective utilization of the low-cost demonstration data that are readily available and can still contain valuable fragmented skills.

In this paper, we propose a novel SeRLA method, which stands for **S**kill-**e**nhanced **R**einforcement **L**earning **A**cceleration from demonstrations, to address the problem of learning from heterogeneous demonstration data and accelerating the downstream RL with the learned knowledge. SeRLA accelerates RL by pursuing *skill-level* learning in two stages with three coherent components: a skill-level adversarial Positive-Unlabeled (PU) learning model, a skill-based policy learning algorithm, and a skill-level data enhancement technique. In the offline skill prior training stage, the skill-level adversarial PU learning model induces useful skill prior knowledge by exploiting the general and task-agnostic demonstration data as unlabeled examples in addition to the *positive* expert data, instead of simply differentiating them. This strategy facilitates improved utilization of the extensive low-cost demonstration data and help alleviate the scarcity of the expert data. In the online downstream RL policy training stage, a skill-based soft actor-critic algorithm is deployed to integrate skills learned in the offline stage and accelerate skill policy learning. Moreover, a simple but novel Skill-level Data Enhancement (SDE) technique is introduced to improve the robustness of skill learning and adaptation at both stages. We conduct experiments on four standard RL environments by comparing the proposed SeRLA with the state-of-the-art skill-based imitation learning methods: SPiRL [17] and model-based SkiMo [18].

The main contributions of this paper can be summarized as follows:

- This is the first work that conducts skill-level Positive-Unlabeled Learning for LfD/IL. The proposed SeRLA takes the low-cost general demonstration data as unlabeled examples to statistically support skill learning from the limited positive examples (i.e., the expert demonstration data) through skill-level adversarial PU learning.
- We propose a simple but novel skill-level data enhancement (SDE) technique, which automatically augments the skill-level representations for both the skill prior learning and the downstream policy learning processes to improve the robustness of the learned skill prior and accelerate the skill-policy function training.
- The proposed SeRLA produces effective empirical results for accelerating downstream RL tasks. It largely outperforms the standard skill prior learning method, SPiRL, while producing notable improvements over the state-of-the-art model-based skill-level method, SkiMo, in the early downstream training stage.

2 Related Works

RL from Demonstrations Learning from Demonstration (LfD) or imitation learning (IL), unlike offline RL which learns optimal policies solely from offline data, focuses on accelerating downstream RL tasks by pre-training the RL agent on a small set of expert demonstrations without reward signals [7, 8]. In addition to the limited expert demonstrations, large task-agnostic demonstration datasets can also be collected inexpensively [9–12] from the environment for extracting potential learnable behaviors through LfD. There are three major methods to solve LfD/IL problems: Behavior Cloning (BC) [7], Inverse Reinforcement Learning (IRL) [21], and Generative Adversarial Imitation Learning (GAIL) [22]. BC learns a direct mapping from observations to actions through supervised learning but struggles with generalization and distribution shifts [23, 24]. IRL infers reward functions from demonstrations and trains the agent using standard RL algorithms, but it is computationally expensive and relies on the reward model's effectiveness [25]. GAIL uses a generative adversarial network where a discriminator distinguishes between agent and expert behaviors, achieving strong performance despite needing many interactions with the environment [26].

Skill-Based RL Skill-based RL methods extract reusable skills as abstracted long-horizon behavior sequences of actions [13, 15, 17, 27, 14, 9, 11, 16], where the skills are predefined by experts or extracted from datasets. In recent works, skill-Prior RL (SPiRL) accelerates downstream RL tasks using learned skill priors from offline demonstration data [17]. Skill-based Learning with Demonstration (SkiLD) regularizes policy learning in downstream tasks using a skill posterior [28]. Few-shot Imitation with Skill Transition Models (FIST) learns skills from few-shot demonstration data and generalizes to unseen tasks [19]. Adaptive Skill Prior for RL (ASPiRe) adaptively learns distinct skill priors from different datasets [20]. Skill-based Model-based RL (SkiMo) applies planning on downstream tasks using a skill dynamics model [18].

Positive-Unlabeled Learning Unlike standard binary classifiers that use positive and negative examples, PU learning utilizes only positive and unlabeled data [29], where unlabeled examples can belong to either class. PU learning has proven valuable in applications like medical diagnosis [30] and knowledge base construction [31, 32]. Prior work has focused on loss functions and optimizers [33, 34]. Kiryo et al. [35] extended PU learning to deep neural networks, and Xu and Denil [36] applied it to imitation reward learning by replacing the adversarial loss in GAIL [22] with PU loss, creating a PU-based reward function for RL agents. By contrast, our work delivers a skill-level two-stage training framework that learns skill priors simultaneously from both limited expert data and low-cost general demonstration data via a PU learning model to accelerate the downstream RL tasks.

3 Problem Setting

Reinforcement Learning from Demonstrations (LfD) aims to accelerate the online downstream RL procedure by leveraging offline demonstration datasets. Specifically, we assume LfD has access to two demonstration datasets: a limited expert dataset D_{π_e} and a low-cost general demonstration dataset D_π. The expert dataset is a task-specific small offline dataset that contains expert demonstration trajectories (state-action sequences) $D_{\pi_e} = \{s_0, a_0, \cdots, s_t, a_t, \cdots\}$, which are generated by human experts or fully trained RL agents. The general demonstration dataset is a much larger task-agnostic offline dataset that consists of randomly collected trajectories $D_\pi = \{s_0, a_0, \cdots, s_t, a_t, \cdots\}$. The action sequences contained in D_{π_e} and D_π can be denoted as A_{π_e} and A_π, respectively. While the general demonstration dataset does not provide as much useful information as the expert dataset, it may still contain short-horizon behaviors that, if properly extracted, can guide the RL agent to behave with feasible actions and propel the policy learning.

The downstream RL task is a standard reinforcement learning problem that can be represented as a Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, as described in [6]. In this MDP, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the transition dynamics $p(s_{t+1}|s_t, a_t)$, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The goal is to learn an optimal policy $\pi^\star : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected discounted cumulative reward (return): $\pi^\star = \arg\max_\pi \; J_r(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r_t]$. The goal of this work is to learn useful skill prior from the offline demonstration datasets and then deploys such skill-level knowledge to facilitate fast policy training for the downstream RL task.

4 Method

The proposed SeRLA method has two stages: the offline skill prior training stage with skill-level adversarial PU learning and the online skill-based downstream policy training. The skill prior training induces useful high-level skill knowledge in form of skill prior from the given demonstration datasets (D_{π_e} and D_π), which is then used to accelerate the downstream skill-based policy training through a Skill-based Soft Actor-Critic (SSAC) algorithm. Moreover, a simple skill-level data enhancement technique is further devised for the two training stages to improve the overall performance. Below, we elaborate on these approach components.

4.1 Skill Prior Training with PU Learning

In the skill prior training stage, we build a regularized deep autoencoder model with skill-level adversarial PU learning to learn a conditional skill prior distribution function $q_\psi(z_t|s_t)$ from the trajectories provided in the two demonstration datasets, where latent variables $\{z_t\}$ are used to capture the high level representations of skills, each of which can be interpreted as an action sequence. The model includes a skill encoder network $q_\mu(\cdot)$, a skill decoder network $p_\nu(\cdot)$, a skill prior network $q_\psi(\cdot)$, and a discriminator $\mathcal{D}_\zeta$. The first three components can be learned from the expert data D_{π_e} within a conventional autoencoder framework, while the discriminator is innovatively deployed to incorporate the general demonstration data D_π into the skill prior training process and alleviate skill data sparsity via adversarial PU learning.

Conventional Skill Learning Framework Under a deep autoencoder framework [17], the skill encoder $q_\mu(z_t|\mathbf{a}_t)$ takes an action sequence $\mathbf{a}_t = \{a_t, ..., a_{t+H-1}\}$ with length H as input and maps it to a latent skill embedding z_t. Conversely, the skill decoder $p_\nu(\hat{\mathbf{a}}_t|z_t)$ reconstructs an action sequence $\hat{\mathbf{a}}_t$ from the given skill z_t. The autoencoder can be learned by minimizing a *reconstruction loss* on the expert action sequences A_{π_e}:

$$L_{rec}(\nu, \mu) = \mathbb{E}_{\mathbf{a}_t \sim A_{\pi_e}} \ell_{ls}\big(\hat{\mathbf{a}}_t \sim p_\nu\big(\,\cdot\,|z_t \!\sim\! q_\mu(\cdot|\mathbf{a}_t)\big), \mathbf{a}_t\big), \tag{1}$$

where $\ell_{ls}(\cdot,\cdot)$ is the standard least squares loss. The skill prior network $q_\psi(z_t|s_t)$ generates a skill z_t from a given starting state s_t. It is designed to support policy network training for the downstream tasks by encoding the expert behavior in a given state in terms of skills from the expert dataset. Ideally, given a pair of observed state and action sequence $(s_t, \mathbf{a}_t)$, the skills produced by the encoder $q_\mu(z_t|\mathbf{a}_t)$ and the prior network $q_\psi(z_t|s_t)$ should be consistent. Hence, the skill prior network can be learned together with the encoder by minimizing the following *prior training loss*:

$$L_{prior}(\psi, \mu) = \mathbb{E}_{(s_t, \mathbf{a}_t) \sim D_{\pi_e}} \mathcal{L}_{KL}(q_\mu(z_t|\mathbf{a}_t), q_\psi(z_t|s_t)), \tag{2}$$

where $\mathcal{L}_{KL}(\cdot,\cdot)$ denotes the Kullback Leibler (KL) divergence function. Moreover, a standard Gaussian distribution prior $p(z_t) = \mathcal{N}(0, 1)$ can be deployed to regularize the skill embedding space with the following *regularization loss*:

$$L_{reg}(\mu) = \mathbb{E}_{\mathbf{a}_t \sim A_{\pi_e}} \mathcal{L}_{KL}(q_\mu(z_t|\mathbf{a}_t), p(z_t)) \tag{3}$$

4.1.1 Skill-level Adversarial PU Learning

Different from the expert data, the randomly collected large demonstration data D_π can present a great number of short-horizon behaviors, some of which can be meaningful while many others can be spontaneous or arbitrary. Hence it is not suitable to directly deploy D_π in the autoencoder model above in the same way as the expert data. To effectively filter out the noisy behaviors and exploit the useful ones from D_π, we deploy a PU learning scheme to perform skill learning simultaneously from both the small expert data D_{π_e} and the large general demonstration data D_π. Specifically, we treat the skills (capturing the behaviors) from the expert data, $Z_e = q_\mu(A_{\pi_e})$, as *positive examples* (i.e., useful skills), and treat skills from the general demonstration data, $Z = q_\mu(A_\pi)$, as *unlabeled examples* that can be either positive or negative. Then we propose to learn a binary probabilistic discriminator $\mathcal{D}_\zeta$ from the positive and unlabeled skill examples by adapting a standard non-negative PU risk function derived in the literature [35] into the skill-level learning:

$$L^{pu}_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e}), q_\mu(A_\pi)) = \lambda L^1_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e})) + \max(-\xi, L^0_{\mathcal{D}_\zeta}(q_\mu(A_\pi)) - \lambda L^0_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e}))) \tag{4}$$

where $\lambda > 0$ and $\xi \geq 0$ are hyperparameters. Here the true positive risk $L^1_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e}))$ is calculated on positive skill examples Z_e, while the true negative risk is calculated on both positive and unlabeled data (Z_e and Z) using two terms, $L^0_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e}))$ and $L^0_{\mathcal{D}_\zeta}(q_\mu(A_\pi))$. These risk terms are defined in terms of the discriminator $\mathcal{D}_\zeta$ as follows:

$$L^1_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e})) = \mathop{\mathbb{E}}_{\mathbf{a}_t \sim A_{\pi_e}} [\log(1 - \mathcal{D}_\zeta(z_t \!\sim\! q_\mu(\cdot|\mathbf{a}_t)))] \tag{5}$$

$$L^0_{\mathcal{D}_\zeta}(q_\mu(A_\pi)) = \mathop{\mathbb{E}}_{\mathbf{a}_t \sim A_\pi} [\log(\mathcal{D}_\zeta(z_t \sim q_\mu(\cdot|\mathbf{a}_t)))] \tag{6}$$

$$L^0_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e})) = \mathop{\mathbb{E}}_{\mathbf{a}_t \sim A_{\pi_e}} [\log(\mathcal{D}_\zeta(z_t \sim q_\mu(\cdot|\mathbf{a}_t)))] \tag{7}$$

where $\mathcal{D}_\zeta(z_t)$ predicts the probability of the given skill vector z_t being a positive example and $(1 - \mathcal{D}_\zeta(z_t))$ denotes the probability of the given skill vector z_t being a negative example.

This PU loss $L^{pu}_{\mathcal{D}_\zeta}$ can be integrated into the deep skill learning model in an *adversarial* manner to enable the exploitation of the large demonstration data D_π: the discriminator $\mathcal{D}_\zeta$ will be learned to *minimize* the PU loss in Eq.(4) given the skill examples extracted, while the skill encoder $q_\mu(\cdot)$ will be learned to *maximize* the PU loss, aiming to alleviate the scarcity of expert data and generalize the skill learning to the large general demonstration data D_π.

Skill Prior Training Algorithm The total loss function for adversarial PU-learning based skill prior training can be expressed as the sum of four terms:

$$L(\mu, \nu, \psi) = L_{rec}(\nu, \mu) + L_{prior}(\psi, \mu) + \beta L_{reg}(\mu) - \rho \min\nolimits_\zeta L^{pu}_{\mathcal{D}_\zeta}(q_\mu(A_{\pi_e}), q_\mu(A_\pi)) \tag{8}$$

where the reconstruction loss L_{rec} enforces consistency between the skill embedding z_t and the action sequence $\mathbf{a}_t$; the prior training loss L_{prior} ensures that the generated skill is consistent with the

Algorithm 1 Skill Prior Training via PU Learning

```
Input: Expert data $D_{\pi_e}$, general demonstration data $D_\pi$
Initialize: Encoder $q_\mu(\cdot)$, decoder $p_\nu(\cdot)$, skill prior $q_\psi(\cdot)$, and discriminator $\mathcal{D}_\zeta(\cdot)$
Output: Trained skill prior network $q_\psi(z_t|s_t)$, and decoder $p_\nu(a_{t:t+H-1}|z_t)$
Procedure:
for each iteration do
    for every $H$ environment steps do
        Sample $s_t$ and sequence $a_{t:t+H-1}$ from $D_{\pi_e}$
        Sample action sequence $a'_{t':t'+H-1}$ from $D_\pi$
        Update $\mu$, $\nu$, $\psi$ by minimizing Eq.(8)
        Update $\zeta$ by maximizing Eq.(8)
    end for
end for
```

current state and action sequence; L_{reg} regularizes the skill embedding space; and the PU loss $L^{pu}_{D_\zeta}$ is used to effectively incorporate large random demonstration data into skill learning in an adversarial manner. The joint training of all these components is expected to effectively and proficiently learn valuable skill knowledge from the unified heterogeneous offline demonstration datasets.

Algorithm 1 outlines the main steps of stochastic skill prior training. At each timestep t, we collect the current state s_t and an action sequence $a_{t:t+H-1}$ from the expert dataset D_{π_e} in an H-step rollout. Similarly we collect a sequence of actions $a_{t':t'+H-1}$ from the general demonstration dataset. We jointly learn the parameters μ, ν, and ψ for the skill prior network $q_\psi(z_t|s_t)$, skill encoder $q_\mu(z_t|a_{t:t+H-1})$, and skill decoder $p_\nu(a_{t:t+H-1}|z_t)$ by minimizing Eq.(8). Adversarially, the discriminator $\mathcal{D}_\zeta$ is updated by minimizing Eq.(4), which is equivalent to maximizing Eq.(8).

4.2 Downstream Policy Training

In the downstream policy training stage, we aim to exploit the skill knowledge learned from the demonstration data, encoded by the skill prior network $q_\psi(\cdot)$ and decoder network $p_\nu(\cdot|z_t)$, to accelerate the online RL process. To this end, we train a skill-based policy network $\pi_\theta(z_t|s_t)$ for the downstream online RL task with skill-level behavior cloning.

Specifically, following the work [17], when interacting with the environment, we sample a skill z_t from the skill policy network $\pi_\theta(\cdot|s_t)$ given the current state s_t. The skill z_t is then decoded to an action sequence $a_{t:t+H-1}$ using the skill decoder $p_\nu(\cdot|z_t)$ to guide the RL agent to reach state s_{t+H} in H steps. The cumulative reward over the H steps, i.e., the H-step reward, can be collected from the environment as $\tilde{r}_t = \sum_t^{t+H} r_t$. With the online skill-based transition data $D = \{(s_t, z_t, \tilde{r}_t, s_{t+H})\}$, we deploy a *Skill-based* soft actor-critic (SSAC) algorithm utilized in [17] to conduct skill-based policy learning with skill-level behavior cloning. SSAC extends soft actor-critic (SAC) [37] to learn the skill policy function network $\pi_\theta(z_t|s_t)$ with the support of a skill-based soft Q-function network $Q_\phi(s_t, z_t)$. It utilizes a KL-divergence regularization term to enforce the skill policy function $\pi_\theta(z_t|s_t)$ to clone the behavior of the pre-trained skill prior network $q_\psi(z_t|s_t)$. In particular, SSAC learns the skill policy function network by maximizing the following regularized expected skill-based Q-value:

$$J_\pi(\theta) = \mathop{\mathbb{E}}_{\substack{s_t \sim D,\\ z_t \sim \pi_\theta}} \left[Q_\phi(s_t, z_t) - \kappa \mathcal{L}_{KL}(\pi_\theta(z_t|s_t), q_\psi(z_t|s_t)) \right]. \tag{9}$$

4.3 Skill-Level Data Enhancement

Collecting expert demonstrations can be challenging and expensive, due to the involvement of human experts [8]. The scarcity of the limited expert data however hinders the robust and effective learning of skills. To alleviate the problem, in addition to incorporating general random demonstration data during the prior training stage, we further propose a Skill-level Data Enhancement (SDE) technique to augment the skill-level data in both the skill prior training stage and the downstream policy training stage, improving the robustness of learning at the skill-level.

Algorithm 2 Online Skill-Policy Training with SDE

```
Input: Skill prior network $q_\psi(z_t|s_t)$, and decoder $p_\nu(a_{t:t+H-1}|z_t)$, $\eta \in (0,1)$
Initialize: Replay buffer $D$, skill policy $\pi_\theta(z_t|s_t)$, and skill-based $Q$ value functions
Output: Trained skill policy network $\pi_\theta(z_t|s_t)$
Procedure:
for each iteration do
    for every $H$ environment steps do
        Sample skill $z_t$ from policy: $z_t \sim \pi_\theta(z_t|s_t)$
        Sample $\mathbf{a}_t$ = $a_{t:t+H-1}$ from decoder $p_\nu(\cdot|z_t)$
        Sample state $s_{t+H}$ and cumulative reward $\tilde{r}_t$ by interacting with environment using $\mathbf{a}_t$
        Update buffer: $D \leftarrow D \cup \{s_t, z_t, \tilde{r}_t, s_{t+H}\}$
        Sample $\epsilon \sim \mathcal{N}^m(0,1)$, and let $\hat{z}_t = z_t + \eta\epsilon$
        Augment buffer: $D \leftarrow D \cup \{s_t, \hat{z}_t, \tilde{r}_t, s_{t+H}\}$
    end for
    for each gradient step do
        Update skill-policy function and Q-function
    end for
end for
```

Conventional data augmentation is typically applied to the input data, e.g., state observations. It is not applicable to the action space since actions (continuous or discrete) cannot be easily re-represented or rescaled. By using a latent variable model to learn skills as latent representations for behaviours captured by action sequences, our proposed approach provides a *new augmentation space at the skill level*, without interfering with the real action space. Specifically, we propose to augment our skill level data with Gaussian noise altered versions as follows. For each skill embedding z_t, we can add a Gaussian noise altered version $\hat{z}_t = z_t + \eta\epsilon$ into learning, where $\epsilon \sim \mathcal{N}^m(0,1)$ is a Gaussian noise vector sampled from a m-dimensional independent standard Gaussian distribution, m is the dimension of the skill embedding, and $\eta \in (0,1)$ is a very small scaling factor. We then enforce z_t and $\hat{z}_t$ *correspond to the same action sequence*, aiming to achieve stable and differentiable representations for different behaviors in the skill embedding space, and enhance the robustness of skill learning.

In the skill prior training stage, SDE is realized by adding the following auxiliary reconstruction loss into the learning objective in Eq.(8):

$$\hat{L}_{rec}(\nu,\mu) = \alpha\, \mathbb{E}_{\mathbf{a}_t \sim A_{\pi_e}} \ell_{ls}\left(\hat{\mathbf{a}}_t \sim p_\nu\big(\cdot\,|\hat{z}_t \sim \eta\epsilon + q_\mu(\cdot|\mathbf{a}_t)\big), \mathbf{a}_t\right) \tag{10}$$

where α is a trade-off parameter. This augmenting loss adds Gaussian noise to the encoded skill vector, aiming to enforce the robustness of the skill encoder and decoder functions, and make them resistant to minor variations in the skill embedding vectors.

In the downstream skill policy training stage, SDE is realized efficiently by augmenting the skill-based transition data. For each observed skill-based transition $\{s_t, z_t, \tilde{r}_t, s_{t+H}\}$, we produce an altered skill vector and add an augmenting transition $\{s_t, \hat{z}_t, \tilde{r}_t, s_{t+H}\}$ to buffer D, without any extra interaction with the environment. The goal is to make the skill-policy network more robust to small variations in the skill embedding space, accelerating the learning process. The online skill policy training process augmented with SDE is outlined in Algorithm 2.

5 Experiment

5.1 Experimental Setting

Environments We conduct experiments with four demonstration-guided tasks with long-horizon and sparse rewards in four different environments that are commonly used for skill learning: Maze, Kitchen, Mis-aligned Kitchen from the D4RL datasets [38] and CALVIN [18] from CALVIN challenge [39]. **Maze** is a navigation environment, in which a point mass agent is required to find a path between a randomly initialized starting point and a goal point [28]. **Kitchen** is a robotic manipulation environment in which a robotic arm completes a sequence of four sub-tasks (Microwave-Kettle-Bottom Burner-Light) [11]. **Mis-aligned Kitchen** is modified from the Kitchen environment

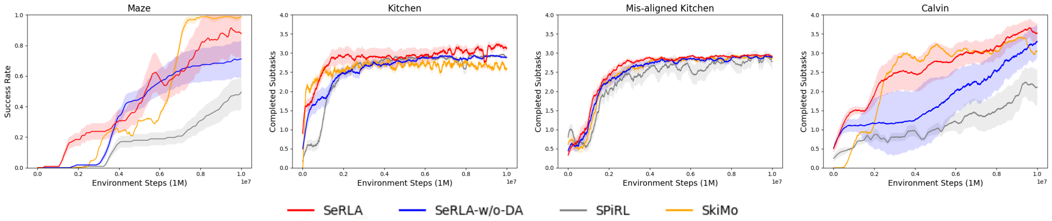

Figure 1: The comparison results on the four long-horizon sparse-reward tasks (Maze, Kitchen, Mis-aligned Kitchen, and CALVIN) are presented in the figure. Each plot presents the average per-trajectory return (i.e., reward) v.s. environment steps for each environment during the downstream training process.

with a different task sequence (Microwave-Light-Slide Cabinet-Hinge Cabinet) [28, 18]. **CALVIN** is a Language-Conditioned Policy Learning challenge[39], which has been adapted for skill learning in SkiMo [18], requiring the RL agent to complete a sequence of four sub-tasks in order: Open Drawer, Turn on Lightbulb, Move Slider Left, and Turn on LED.

Comparison Methods We compared the proposed SeRLA with two state-of-the-art skill-based methods, **SPiRL** [17] and **SkiMo** [18], which train skill priors for the downstream tasks.

5.2 Experimental Results

We conducted experiments on the four environments to compare the proposed full method, SeRLA, and its variant without SDE, SeRLA-w/o-SDE, with the other two skill-based methods, SPiRL and SkiMo. The skills were learned on the demonstration data prior to the downstream task and the reward was evaluated in the downstream RL learning process over 10^7 environment steps. The maximum trajectory reward for Maze environment is 1, while for Kitchen, Mis-aligned Kitchen, and CALVIN environments, it is 4. The experimental results are presented in Figure 1. The four plots on the left side report results (return v.s. environment steps) on the four environments separately. We can see that the proposed SeRLA-w/o-SDE largely outperforms the baseline SPiRL across all the four environments, especially on Maze, Mis-aligned Kitchen and CALVIN, which validates the effectiveness and contribution of the proposed PU Learning component and SSAC algorithm in extracting and deploying skill knowledge. The proposed full approach SeRLA further boosts the performance over SeRLA-w/o-SDE with notable gains across the four plots, which verifies the effectiveness of the proposed skill-level data enhancement technique. SeRLA also outperforms the model-based state-of-the-art SkiMo and produces best results on Kitchen and Mis-aligned Kitchen. On the other two environments, Maze and CALVIN, SeRLA yields comparable overall performance to SkiMo throughout the downstream RL training, while producing the best results in the early training stage.

6 Conclusion

In this paper, we proposed a novel two-stage skill-level learning method SeRLA to exploit offline demonstration data and accelerate downstream RL tasks. SeRLA deploys skill-level adversarial PU learning to learn reusable skills from both limited expert demonstration data and large low-cost demonstration data. Then a skill-based soft actor-critic algorithm is deployed to utilize the learned skill prior knowledge and accelerate the online downstream RL through skill-based behavior cloning. The proposed approach conveniently provides a new augmentation space at the skill level without interfering with the real action space, which enables novel skill-level data enhancement (SDE) in both training stages. Our experimental results on four benchmark environments demonstrate that SeRLA outperforms two state-of-the-art skill learning methods, SPiRL and SkiMo, especially in the early downstream training stage. Given the promise and effectiveness of our simple but innovative skill-level data enhancement technique, we plan to further investigate various data augmentation strategies in the high level latent skill space in the future.

References

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, 2013.

[2] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser *et al.*, "Starcraft ii: A new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[3] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[4] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," in *NeurIPS*, 2020.

[5] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," in *NeurIPS*, 2022.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, 2009.

[8] T. Brys, A. Harutyunyan, H. B. Suay, S. Chernova, M. E. Taylor, and A. Nowé, "Reinforcement learning from demonstration through shaping," in *IJCAI*, 2015.

[9] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller, "Learning an embedding space for transferable robot skills," in *ICLR*, 2018.

[10] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," in *ICLR*, 2020.

[11] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," in *CoRL*, 2019.

[12] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *CoRL*, 2018.

[13] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim, "Composing complex skills by learning transition policies," in *ICLR*, 2019.

[14] M. Dalal, D. Pathak, and R. R. Salakhutdinov, "Accelerating robotic reinforcement learning via parameterized action primitives," in *NeurIPS*, 2021.

[15] Y. Lee, J. Yang, and J. J. Lim, "Learning to coordinate manipulation skills via skill behavior diversification," in *ICLR*, 2020.

[16] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," in *ICLR*, 2019.

[17] K. Pertsch, Y. Lee, and J. Lim, "Accelerating reinforcement learning with learned skill priors," in *CoRL*, 2021.

[18] L. X. Shi, J. J. Lim, and Y. Lee, "Skill-based model-based reinforcement learning," in *CoRL*, 2022.

[19] K. Hakhamaneshi, R. Zhao, A. Zhan, P. Abbeel, and M. Laskin, "Hierarchical few-shot imitation with skill transition models," in *ICLR*, 2022.

[20] M. Xu, M. Veloso, and S. Song, "ASPire: Adaptive skill priors for reinforcement learning," in *NeurIPS*, 2022.

[21] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *ICML*, 2004.

[22] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *NeurIPS*, 2016.

[23] S. Ross and D. Bagnell, “Efficient reductions for imitation learning,” in *AISTATS*, 2010.

[24] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *AISTATS*, 2011.

[25] Q. Li, J. Zhang, D. Ghosh, A. Zhang, and S. Levine, “Accelerating exploration with unlabeled prior data,” in *NeurIPS*, 2023.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” in *NeurIPS*, 2014.

[27] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu, “Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization,” in *CoRL*, 2021.

[28] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim, “Demonstration-guided reinforcement learning with learned skills,” in *CoRL*, 2021.

[29] J. Bekker and J. Davis, “Learning from positive and unlabeled data: A survey,” *Machine Learning*, 2020.

[30] M. Claesen, F. De Smet, P. Gillard, C. Mathieu, and B. De Moor, “Building classifiers to predict the start of glucose-lowering pharmacotherapy using belgian health expenditure data,” *arXiv preprint arXiv:1504.07389*, 2015.

[31] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek, “Fast rule mining in ontological knowledge bases with amie+,” *The VLDB Journal*, 2015.

[32] K. Zupanc and J. Davis, “Estimating rule quality for knowledge base completion with the relationship between coverage assumption,” in *WWW*, 2018.

[33] M. C. Du Plessis, G. Niu, and M. Sugiyama, “Analysis of learning from positive and unlabeled data,” in *NeurIPS*, 2014.

[34] G. Patrini, F. Nielsen, R. Nock, and M. Carioni, “Loss factorization, weakly supervised learning and label noise robustness,” in *ICML*, 2016.

[35] R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama, “Positive-unlabeled learning with non-negative risk estimator,” in *NeurIPS*, 2017.

[36] D. Xu and M. Denil, “Positive-unlabeled reward learning,” in *CoRL*, 2021.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *ICML*, 2018.

[38] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, “D4rl: Datasets for deep data-driven reinforcement learning,” *arXiv preprint arXiv:2004.07219*, 2020.

[39] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, “Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks,” *IEEE Robotics and Automation Letters*, 2022.