Long-term Feature Extraction via Frequency Prediction for Efficient Reinforcement Learning

Jie Wang*, *Senior Member, IEEE,* Mingxuan Ye, Yufei Kuang, Rui Yang, Wengang Zhou, *Senior Member, IEEE,* Houqiang Li, *Fellow, IEEE,* and Feng Wu, *Fellow, IEEE*

Abstract—Sample efficiency remains a key challenge for the deployment of deep reinforcement learning (RL) in real-world scenarios. A common approach is to learn efficient representations through future prediction tasks, facilitating the agent to make farsighted decisions that benefit its long-term performance. Existing methods extract predictive features by predicting multi-step future state signals. However, they do not fully exploit the structural information inherent in sequential state signals, which can potentially improve the quality of long-term decision-making but is difficult to discern in the time domain. To tackle this problem, we introduce a new perspective that leverages the frequency domain of state sequences to extract the underlying patterns in time series data. We theoretically show that state sequences contain structural information closely tied to policy performance and signal regularity and analyze the fitness of the frequency domain for extracting these two types of structural information. Inspired by that, we propose a novel representation learning method, **S**tate Sequences **P**rediction via **F**ourier Transform (SPF), which extracts long-term features by predicting the Fourier transform of infinite-step future state sequences. The appealing features of our frequency prediction objective include: 1) simple to implement due to a recursive relationship; 2) providing an upper bound on the performance difference between the optimal policy and the latent policy in the representation space. Experiments on standard and goal-conditioned RL tasks demonstrate that the proposed method outperforms several state-of-the-art algorithms in terms of both sample efficiency and performance.

Index Terms—Reinforcement learning, Representation learning, State sequences prediction, Fourier transform.

1 Introduction

Deep reinforcement learning (RL) has made significant advancements in addressing complex sequential decision-making tasks, such as computer games [1], robotic manipulation [2], combinatorial optimization [3], and language modeling [4]. However, these methods often require large amounts of training data to achieve satisfying control, which limits their practical applications due to the high cost of data collection in real-world scenarios. To enhance the sample efficiency of RL methods, prior studies have emphasized the importance of representation learning [5], [6], [7]. This approach aims to extract adequate and valuable information from raw sensory data and train RL agents in a learned representation space, which enhances the agent's exploration efficiency and consequently leads to greater data efficiency [8], [9], [10], [11]. A variety of these algorithms rely on auxiliary self-supervision tasks, such as predicting future reward signals [8] and reconstructing future observations [9], to incorporate prior knowledge of the environment into the representations.

Given the sequential nature of RL tasks, multi-step future signals inherently offer richer information for long-term decision-making compared to immediate future signals.

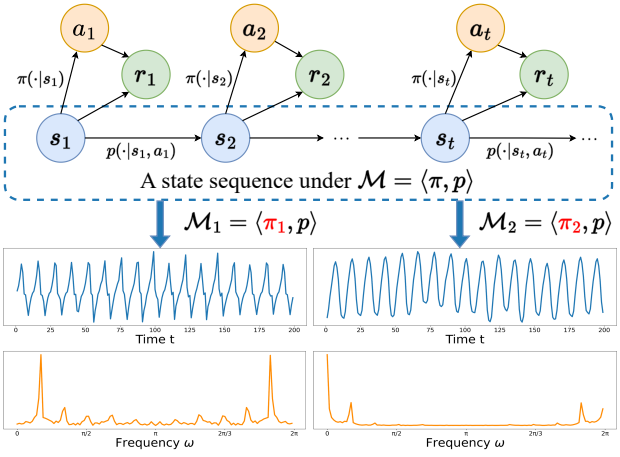


Fig. 1: **State Sequence Generation Process**. The top row, marked by the blue dashed box, displays a sample of state sequences generated from an MDP $\mathcal{M}$. The MDP $\mathcal{M}$ is modeled by the policy π and the transition function p. The bottom two rows visualize two state sequences in the time domain (blue lines) and the frequency domain (orange lines) respectively. Each column corresponds to a state sequence generated from a different policy.

Recent studies have demonstrated that leveraging future reward sequences as supervisory signals effectively boosts the generalization capabilities of visual RL algorithms [12], [13]. Nonetheless, we argue that state sequences provide more informative supervisory signals than sparse rewards. As illustrated in the top part of Fig. 1, the sequence of

- *Jie Wang is with MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei 230027, China. (Corresponding author, e-mail: jiewangx@ustc.edu.cn).*
- *Mingxuan Ye, Yufei Kuang, Rui Yang, Wengang Zhou, Houqiang Li, and Feng Wu are with MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China, Hefei 230027, China. E-mail: {mingxuanye, yfkuang, yr0013}@mail.ustc.edu.cn, {zhwg, lihq, fengwu}@ustc.edu.cn.*

Manuscript received April 19, 2005; revised August 26, 2015.

future states essentially determines future actions and further influences the sequence of future rewards. Therefore, state sequences maximally preserve the influence of the transition intrinsic to the environment and the effect of actions generated from the current policy.

Although state sequences present various benefits, extracting features from these sequential data through long-term prediction tasks remains a notable challenge. A major obstacle lies in the difficulty of learning accurate long-term prediction models for feature extraction. The previous approaches have sought to make multi-step predictions using a one-step dynamic model by repeatedly feeding the prediction back into the learned model [14], [15], [16]. However, these approaches require a high degree of accuracy in the one-step model to avoid accumulating errors in multi-step predictions [16]. An alternative strategy involves learning a multi-step dynamics model that predicts an entire state sequence of fixed length at one time [17]. However, the prediction horizons considered by these prediction models are inherently limited, as the model parameters and computation complexity scale proportionally with the prediction length for time series. In addition, the self-supervised objectives of existing methods treat states at each time step as independent variables, ignoring the dependence structures present in the entire time series.

To address these challenges, we introduce a new perspective that leverages the frequency domain of state sequences with arbitrary lengths to extract the structure information inherent in time series. In Section 4, we theoretically establish two types of sequential dependency structures inherent in state sequences. The first structure involves the dependency between reward sequences and state sequences. In other words, our findings indicate that state sequences implicitly reflect the performance of the current policy, and the distributions of state sequences may exhibit notable differences under good and poor policies. The second structure pertains to the temporal dependencies among the state signals, namely the regularity patterns exhibited by the state sequences. By exploiting the structural information, representations can focus on the underlying critical information of long-term signals, thus circumventing the need for an accurate prediction model that may introduce fine-granted features irrelevant to long-term decisions.

We conduct a deeper analysis of the suitability of the frequency domain for extracting these two types of structural information in Section 5.1. Leveraging the frequency domain offers several key advantages. Firstly, the frequency domain directly manifests the regularity properties of the time-series data, as regular patterns can be explicitly expressed as frequency components through the Fourier transform [18], [19], [20]. Secondly, we demonstrate in Section 5.1 that the Fourier transform of state sequences retains the ability to indicate policy performance under certain assumptions. Moreover, Fig. 1 provides an intuitive understanding that the frequency domain enables more effective discrimination of two similar temporal signals that are difficult to differentiate in the time domain, thereby containing more compact and effective features useful for policy learning.

Building upon the above intuition and theoretical analyses, we propose **S**tate Sequences **P**rediction via **F**ourier Transform (**SPF**), a novel representation method that effectively captures long-term features by predicting the frequency domain of state sequences. Specifically, our method utilizes an auxiliary self-supervision task that predicts the Fourier transform (FT) of infinite-step state sequences to improve the efficiency of representation learning. To facilitate practical implementation, we reformulate the Fourier transform of state sequences as a recursive form. This allows the prediction loss to be expressed as a TD error [21], which depends only on single-step future state. Thus, SPF is simple to implement and eliminates the need to store infinite-step state sequences when computing the labels of FT.

To further highlight our method's advantages in facilitating far-sighted decisions, we extend SPF to goal-conditioned RL (GCRL) tasks. In GCRL tasks, the desired outcomes for the agent are specified as a single goal state, rather than a mathematical reward function. Thus, the agent requires more foresight to ensure that it approaches the desired final goal appropriately under the current policy. Considering that GCRL agents show little periodicity and require more accurate guidance of future directions, we propose an enhanced version of SPF on GCRL tasks, which adds supervision in the time domain to calibrate the prediction in the frequency domain. Experiments demonstrate that our method outperforms several state-of-the-art algorithms in terms of both sample efficiency and performance on six MuJoCo tasks and five GCRL tasks. Additionally, we visualize the fine distinctions between the recovered states from our predicted FT and the true states, which indicates that our representation effectively captures the inherent structures of future state sequences. By visualizing the goal trajectories of GCRL tasks, we show that adding temporal supervision helps the representation to capture richer temporal dependencies, such as trends and directions, providing more accurate guidance for future decisions.

An earlier version of this work has been published at NeurIPS 2023 as a spotlight [22]. This paper extends the conference version by extending our method to the goal-reaching RL settings, expanding the technical contributions, and adding new theoretical analyses. We summarize the main extensions as follows. First, we extend our method to deal with GCRL tasks, which exhibit sparse rewards and non-periodic state signals, and conduct more experiments on two types of goal-oriented tasks (see Section 6). Second, considering that GCRL requires more accurate guidance of future trajectory, we propose an enhanced version of SPF, named **S**tate Sequences **P**rediction in both the **F**requency domain and **T**ime domain(**SPFT**), which calibrates the prediction in the frequency domain by adding supervision in the time domain (see Section 5.3). Third, we extend the assumption of original theorems about reward functions to make them more compatible with the GCRL setting (see Theorem 2 & 4). Fourth, we propose the bounded sub-optimality theorem to bridge the gap between the motivation to leverage the frequency distribution of state sequences and the implementation of our method (see Theorem 6).

2 Related Work

2.1 Representation Learning in RL

Representation learning is critical for automatically extracting features and enhancing the exploration efficiency of

downstream tasks in a variety of real-world domains, including biomedicine [23], [24], neuroscience [19], and dialogue systems [25]. Learning good representations to improve the efficiency of RL methods has been studied for a long time [26]. One well-explored solution is leveraging the self-supervised data collected automatically from RL tasks, which can be decomposed into two categories. Works in the first category explore the usage of auxiliary tasks, such as predicting the future signals [9], [27], designing contrastive objectives to distinguish positive and negative samples [28], [29], [30], and learning distance metrics in the representation space [7], [31], [32]. The second category, specifically tailored to model-based RL methods, leverages generative models of environment dynamics, enforcing the representations to encode enough information about the environments [14], [15], [16].

Many existing approaches employ auxiliary tasks that predict single-step future reward or state signals to improve the efficiency of representation learning [2], [9], [27]. However, multi-step future signals inherently contain more valuable features for long-term decision-making than single-step signals. Recent studies have shown the effectiveness of using future reward sequences as supervisory signals to improve the generalization performance of visual RL algorithms [12], [13]. Several studies propose making multi-step predictions of state sequences using a one-step dynamic model by repeatedly feeding the prediction back into the learned model, which is applicable to both model-free [14] and model-based [15] RL. However, these approaches demand a high level of accuracy in the one-step model to prevent accumulating errors [16]. To address this issue, the existing method learned a prediction model to directly predict multi-step future states [17], which results in significant additional storage for the prediction labels. In our work, we propose to predict the FT of state sequences, which reduces the demand for high prediction accuracy and eliminates the need to store multi-step future states as prediction targets.

2.2 Goal-Conditioned RL

GCRL typically reshapes the reward function into a binary bonus for reaching the desired goal, eliminating the need for manually-designed rewards, but leading to the sparse reward problem [33], [34]. This paper focuses on two ways to tackle the sparse reward problem and improve the sample efficiency of GCRL algorithms: 1) designing self-supervised goal representation learning methods, and 2) extending the idea of hindsight relabeling.

Self-supervised representation learning for the goal-conditioned setting leverages denser training signals, including states and actions. Many of these works are derived from the standard RL setting and can be similarly classified as predictive-based [35], contrastive-based [36], [37], and distance-based methods [38], [39]. For example, the predictive-based method learns a good state-goal representation by reconstructing the difference between the current state and the goal state, thus making the model more aware of task-relevant information [35]. Contrastive-based methods learn representations that cluster observations from the same trajectory [36] or cluster goals achieved in adjacent time steps [37]. Distance-based methods group states that require similar actions to reach [38] or group analogous tasks based on the behavioral similarity between state-goal pairs [39]. Our method belongs to the predictive-based category. However, unlike prior works, our method leverages the long-term future information based on the directed goals.

On the other hand, the works of hindsight relabeling aims to make the reward signal denser and can be traced back to Hindsight Experience Replay (HER) [33], which reuses the samples from failed trajectories by replacing the raw desired goal with another achievable goal in the same trajectory. Many works use HER as the backbone and improve the idea of hindsight relabeling by curriculum learning [40], [41], goal generation [42], [43], and prioritized experience replay [44], [45]. There exists work proposing to learn and utilize a dynamics model to generate virtual trajectories for relabeling [46], [47]. Foresight goal relabeling prevents labeling homogeneous goals limited in historical data and plans new goals that the current policy can achieve. Similar in spirit to this work, our method relabels the goals with future states generated by the Fourier prediction model, thus mitigating concerns about error accumulation.

2.3 Incorporating the Fourier Features

There are various traditional RL methods designed to capture Fourier features. Early works investigated the use of fixed bases, such as the Fourier basis, which allows for the decomposition of functions into a sum of simpler periodic components. [21], [48]. Another research explored enriching the representational capacity using random Fourier features of the observations. Moreover, in the field of self-supervised pre-training, neuro2vec [19] performs representation learning by predicting the Fourier transform of the masked part of the input signal, which is similar to our work but requires storing the entire signal as a label.

3 PRELIMINARIES

3.1 MDP Notation

Markov decision process (MDP) provides a formal modeling approach for the interaction between the environment and the agent in RL tasks. We consider the standard MDP framework [49], in which the environment is given by the tuple $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, R, P, \mu, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [-R_{\max}, R_{\max}]$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, $\mu : \mathcal{S} \to [0, 1]$ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. A policy π defines a probability distribution over actions conditioned on the state, i.e. $\pi(a|s)$. The environment starts at an initial state $s_0 \sim \mu$. At time $t \geq 0$, the agent follows a policy π and selects an action $a_t \sim \pi(\cdot|s_t)$. The environment then stochastically transitions to a state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and produces a reward $r_t = R(s_t, a_t, s_{t+1})$. The goal of RL is to select an optimal policy π^* that maximizes the cumulative sum of future rewards. Following previous work [50], [51], we define the *performance* of a policy π as its expected sum of future discounted rewards:

$$J(\pi, \mathcal{M}) := \mathbb{E}_{\tau \sim (\pi, \mathcal{M})} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \right], \qquad (1)$$

where $\tau := (s_0, a_0, s_1, a_1, \cdots)$ denotes a trajectory generated from the interaction process and $\tau \sim (\pi, \mathcal{M})$ indicates that the distribution of τ depends on π and the environment model $\mathcal{M}$. For simplicity, we write $J(\pi)$ and $\tau \sim \pi$ as shorthand since our environment is stationary. We also interest about the discounted future state distribution d^π, which is defined by $d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi, \mathcal{M})$. It allows us to express the expected discounted total reward compactly as

$$J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^\pi \\ a\sim\pi \\ s'\sim P}}\left[R(s,a,s')\right]. \tag{2}$$

The proof of (2) can be found in [52] or Section A.1.1 in the appendix.

Since our methods seek to train an encoder that effectively captures the useful aspects of the environment, we also consider a latent MDP $\overline{\mathcal{M}} = \langle \overline{\mathcal{S}}, \mathcal{A}, \overline{R}, \overline{P}, \overline{\mu}, \gamma \rangle$, where $\overline{\mathcal{S}} \subset \mathbb{R}^D$ for finite D and the action space $\mathcal{A}$ is shared by $\mathcal{M}$ and $\overline{\mathcal{M}}$. We aim to learn an embedding function $\phi : \mathcal{S} \to \overline{\mathcal{S}}$, which connects the state spaces of these two MDPs. We similarly denote $\overline{\pi}^*$ as the optimal policy in $\overline{\mathcal{M}}$. For ease of notation, we use $\overline{\pi}(\cdot|s) := \overline{\pi}(\cdot|\phi(s))$ to represent first using ϕ to map $s \in \mathcal{S}$ to the latent state space $\overline{\mathcal{S}}$ and subsequently using $\overline{\pi}$ to generate the probability distribution over actions.

3.2 Goal-conditioned Reinforcement Learning

Goal-conditioned reinforcement learning (GCRL) learns a policy that guides the agent to achieve desired outcomes, usually specified as goals, rather than maximizing a heuristic reward function. Unlike the standard RL policy solely depending on the states, GCRL additionally requires the agent to make decisions according to different goals.

Formally, GCRL can be characterized by the tuple $\mathcal{M}_{\text{goal}} := \langle \mathcal{S}, \mathcal{A}, G, R_{\text{goal}}, P, \mu, \gamma \rangle$, which differs from the standard MDP by introducing the goal set G and the goal-conditioned reward function $R_{\text{goal}} : \mathcal{S} \times G \to \mathbb{R}$. Given a desired goal g, the agent is tasked to learn a goal-conditioned policy $\pi_{\text{goal}} : \mathcal{S} \times G \to \mathcal{A}$ to reach g as soon as possible from the current state s_t. Until the agent reaches the goal, it receives a negative reward $r_{t+1} = R_{\text{goal}}(s_t; g)$, which is commonly redefined as a binary signal:

$$R_{\text{goal}}(s_t; g) = \begin{cases} 0 & \|\phi_{\text{goal}}(s_{t+1}) - g\|_2^2 < \text{threshold} \\ -1 & \text{otherwise} \end{cases}. \tag{3}$$

Here, $\phi_{\text{goal}} : \mathcal{S} \to G$ is a tractable mapping function that provides goal representation. Note that a continuous distance measure between the achievable goal $\phi_{\text{goal}}(s_{t+1})$ and the desired goal g provides a denser reward signal than (3). However, it may cause additional local optima challenges in some cases where the agent must first increase the distance to the goal before finally reaching it [53].

In the goal-conditioned setting, the agent aims to maximize the expected cumulative rewards given any s and g, which equals the negated total cost:

$$J(\pi_{\text{goal}}, \mathcal{M}_{\text{goal}}; g) = \mathbb{E}_{\tau\sim(\pi_{\text{goal}}, \mathcal{M}_{\text{goal}})}\left[\sum_{t=0}^{\infty}\gamma^t R_{\text{goal}}(s_{t+1}; g)\right] \tag{4}$$

We then formulate the goal-conditioned value function $V^\pi(s; g)$ for π. There exists an optimal policy π^* that is universally optimal:

$$V^{\pi^*}(s; g) = \max_\pi V^\pi(s; g), \quad \forall\, s \in \mathcal{S}, g \in G. \tag{5}$$

We thus define the optimal value function $V^* := V^{\pi^*}$.

Similarly, we can define the optimal state-action value function, i.e., Q-function:

$$Q^*(s, a; g) := \mathbb{E}_{s'\sim P(s,a)}\left[R_{\text{goal}}(s'; g) + \gamma\, V^*(s'; g)\right]. \tag{6}$$

3.3 Discrete-Time Fourier Transform

The discrete-time Fourier transform (DTFT) is a powerful mathematical tool to decompose a time-domain signal into different frequency components. It converts a real or complex sequence $\{x_n\}_{n=-\infty}^{+\infty}$ into a complex-valued function

$$F(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-j\omega n},$$

where ω is a frequency variable.

Owing to the discrete-time nature of the original signal, the DTFT function is 2π-periodic for its frequency variable, i.e., $F(\omega + 2\pi) = F(\omega)$. Therefore, all of our interest lies in the range $\omega \in [0, 2\pi]$ that contains all the necessary information of the infinite-horizon time series.

It is worth mentioning that the signals considered in this paper, namely state sequences, are real-valued, which guarantees the conjugate symmetry property of DTFT, i.e., $F(2\pi - \omega) = F^*(\omega)$. Therefore, in practice, it suffices to predict the DTFT only on the range of $[0, \pi]$, which can reduce the number of parameters and save storage space.

The *inverse DTFT* recovers the discrete sequential data from the continuous frequency function $F(\omega)$:

$$x_n = \frac{1}{2\pi}\int_{2\pi} F(\omega)e^{j\omega n}\,\mathrm{d}\omega.$$

For numerical computation, a common practice is to uniformly sample L points within one cycle of the frequency function. The discretized DTFT is equivalent to the discrete Fourier transform (DFT) of the periodic summation of the original time series $\{x_n\}$:

$$\begin{aligned} F\left(\frac{2\pi k}{L}\right) &= \sum_{n=-\infty}^{\infty} x_n e^{-j\frac{2\pi k}{L}n},\ k = 0, 1, \cdots, L-1 \\ &= \sum_{n=0}^{L-1} x_L[n] e^{-j\frac{2\pi k}{L}n},\ k = 0, 1, \cdots, L-1, \end{aligned}$$

where $x_L[n] = \sum_{m=-\infty}^{\infty} x_{n-mL}$ denotes a periodic summation of $\{x_n\}$ with a period of L. Note that when the maximum length of the time series $\{x_n\}$ is less than or equal to the number of discrete samples N, we have $x_L[n] = x_n$. Thus, with sufficient discretization refinement, we can deduce the inverse FT from the inverse DFT formula:

$$\widehat{x_n} = \frac{1}{L}\sum_{k=0}^{L-1} F\left(\frac{2\pi k}{L}\right)e^{j\frac{2\pi k}{L}n},\ n = 0, 1, \cdots L. \tag{7}$$

In the following paper, we will use Equation (7) to recover the state signals from the predicted frequency domain.

4 Structural Information in State Sequences

This section presents a theoretical exploration of the inherent structural information found in state sequences. We propose that state sequences contain two types of sequential dependency structures, which are valuable for indicating policy performance and capturing regularity features of the states, respectively.

4.1 Policy Performance Distinction via State Sequences

In the field of RL, it is commonly acknowledged that a greedy manner of pursuing the highest reward at each time step does not guarantee the maximum long-term benefits. Consequently, RL algorithms maximize the cumulative rewards over an episode, rather than the immediate reward, to encourage the agent to make farsighted decisions. This motivates previous work that leverages information from future reward sequences to capture long-term features for enhanced representation learning [12].

We argue that, compared to the sparse reward signals, sequential state signals contain richer information. In MDP, the stochasticity of a trajectory derives from random actions selected by the agent and the environment's subsequent transitions to the next state and reward. These two sources of stochasticity are modeled as the policy $\pi(a|s)$ and the transition $p(s', r|s, a)$, respectively. Both of them are conditioned on the current state. Over long interaction periods, the dependencies of action and reward sequences on state sequences become more evident. That is, the sequence of future states largely determines the sequence of actions that the agent selects and further determines the corresponding sequence of rewards, which implies the trend and performance of the current policy, respectively. Thus, state sequences not only explicitly contain information about the environment's dynamics model, but also implicitly reveal information about policy performance.

Furthermore, we provide theoretical support for the previous statement, which demonstrates that the distribution distance between two state sequences produced by different policies serves as an upper bound on the performance difference between those policies, under certain assumptions about the reward function.

Theorem 1. *Suppose that the reward function is solely dependent on the* ***current state s****, expressed as $R(s, a, s') = R(s)$, then the performance difference between two arbitrary policies π_1 and π_2 is bounded by the L1 norm of the difference between their state sequence distributions:*

$$|J(\pi_1) - J(\pi_2)| \leq \frac{R_{max}}{1-\gamma} \cdot \|P(s_0, s_1, s_2, \dots|\pi_1, \mathcal{M}) - P(s_0, s_1, s_2, \dots|\pi_2, \mathcal{M})\|_{L1}, \quad (8)$$

where $P(s_0, s_1, s_2, \dots|\pi, \mathcal{M})$ means the joint distribution of the infinite-horizon state sequence $\mathbf{S} = \{\mathbf{s_0}, \mathbf{s_1}, \mathbf{s_2}, \dots\}$ conditioned on the policy π and the environment model $\mathcal{M}$.

The proof of this theorem is provided in Appendix A.1.1. The theorem demonstrates that the greater the difference in policy performance, the greater the difference in their corresponding state sequence distributions. When we adjust the ratio $\frac{R_{max}}{1-\gamma}$ to take a relatively small value by scaling the reward, the theorem indicates that good and bad policies generate significantly different state sequence distributions. Furthermore, it confirms that learning via state sequences can substantially influence the search for policies with good performance.

Remark 1. The L-1 norm used in the theorems of our paper pertains to probability functions. According to *Scheffé's identity* [54], the L1-norm distance between two probability density functions p and q is half the total variation (TV) norm, that is, $\|p - q\|_{L1} = 2\|p - q\|_{TV} = 2 \sup_{B \in \mathcal{B}(R^D)} \left|\int_B p(x)\,\mathrm{d}x - \int_B q(x)\,\mathrm{d}x\right|$. Furthermore, the total variation distance can be upper bounded by the Kullback-Leibler (KL) divergence, that is, $\|p - q\|_{TV} \leq \sqrt{D_{KL}(p\|q)}$. Therefore, in Theorem 1, the L1-norm can be replaced with the commonly used KL divergence for measuring distances between two state sequences.

We provide a complementary theorem based on Theorem 1, extending the assumption of reward function from $R(s, a, s') = R(s)$ to $R(s, a, s') = R(s')$. The extended assumption is more compatible with the reward function (3) in the goal-conditioned setting.

Theorem 2. *Suppose that the reward function is solely dependent on the* ***next state s'****, expressed as $R(s, a, s') = R(s')$, then the performance difference between two arbitrary policies π_1 and π_2 is bounded by the L1 norm of the difference between their state sequence distributions:*

$$|J(\pi_1) - J(\pi_2)| \leq \frac{R_{max}}{\gamma(1-\gamma)} \cdot \|P(s_0, s_1, s_2, \dots|\pi_1, \mathcal{M}) - P(s_0, s_1, s_2, \dots|\pi_2, \mathcal{M})\|_{L1}, \quad (9)$$

where $P(s_0, s_1, s_2, \dots|\pi, \mathcal{M})$ means the joint distribution of the infinite-horizon state sequence $\mathbf{S} = \{\mathbf{s_0}, \mathbf{s_1}, \mathbf{s_2}, \dots\}$ conditioned on the policy π and the environment model $\mathcal{M}$.

In a goal-conditioned setting, the reward function has the form $r_t = R(s_{t+1}; g)$. From Theorem 2, we have

$$|J(\pi_1; g) - J(\pi_2; g)| \leq \frac{R_{max}}{\gamma(1-\gamma)} \|P(s_0, s_1, s_2, \dots|\pi_1, \mathcal{M}, g) - P(s_0, s_1, s_2, \dots|\pi_2, \mathcal{M}, g)\|_{L1}. \quad (10)$$

It is worth noting that the assumption of the reward function depending on the next state is inherently more challenging than the dependence on the current state, considering that the expected reward involves a connection with the policy distribution when $R(s, a, s') = R(s')$. The detailed proof is provided in Appendix A.1.1.

4.2 Asymptotic Periodicity of States in MDP

A variety of real-world tasks exhibit cyclic behaviors due to the periodic characteristics of their environment dynamics. Examples include industrial robots, autonomous driving in specific traffic scenarios, and financial portfolio management. Consider the assembly robot as an illustrative example. The robot is trained to assemble parts together to create a final product. Upon reaching a stable policy, it executes a sequence of periodic movements to efficiently

perform the assembly. In the case of MuJoCo tasks, particularly those involving locomotion control, the cyclical nature of behaviors is often evident when the agent reaches a stable policy. We provide a corresponding video in the supplementary material to show the periodic locomotion of several MuJoCo tasks.

Motivated by these cases, we perform a theoretical analysis to establish that, under some assumptions about the transition probability matrices, state sequences in finite state space may exhibit asymptotically periodic behaviors when the agent reaches a stable policy.

Theorem 3. *Suppose that the state space $\mathcal{S}$ is finite with a transition probability matrix $P \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ and $\mathcal{S}$ has α recurrent classes. Let $R_1, R_2, \ldots, R_\alpha$ be the probability submatrices corresponding to the recurrent classes and let $d_1, d_2, \ldots, d_\alpha$ be the number of the eigenvalues of modulus 1 that the submatrices $R_1, R_2, \ldots, R_\alpha$ has. Then for any initial distribution μ_0, $P^n\mu_0$ is asymptotically periodic with period $d = \mathrm{lcm}(d_1, d_2, \ldots, d_\alpha)$.*

The proof of the theorem is provided in Appendix A.2. Note that in an infinite state space, if the Markov chain contains one or more recurrent classes, the state will inevitably enter one of the recurrent classes after a sufficient number of transition steps. In such cases, the asymptotic periodicity of the state sequences can be also analyzed using the aforementioned theorem.

Theorem 3 demonstrates that there exist highly-structured regularity features in the state sequences. These regular patterns can be explicitly expressed as frequency components through the Fourier transform. Thus, the frequency domain promises a more compact and efficient way to capture these inherent structures of state signals than temporal feature extraction. Furthermore, even in cases where the state sequences lack a strictly periodic pattern, the frequency domain retains complete information about the state signals without any information loss due to the reversibility of the Fourier transform. In Section 6.2.2, we test our method on Arm Manipulation tasks to show the effectiveness of our method in non-periodic environments.

5 METHOD

In the previous section, we demonstrate that state sequences contain rich structural information that implicitly indicates the policy performance and regular behavior of states. However, such information is not explicitly visible in the time domain. In this section, we will discuss how to effectively leverage the inherent structural information of time series data through the frequency domain.

5.1 Learning via Frequency Domain of State Sequences

Here, we will discuss the advantages of leveraging the frequency pattern of state sequences for capturing the inherent structural information explicitly and efficiently.

According to the following theorem, we find that the FT of the state sequences preserves the property in the time domain. That is, the distribution difference between state sequences in the frequency domain also controls the performance difference between the corresponding two policies.

Theorem 4. *Let $\mathcal{S} \subset \mathbb{R}^D$ and $A = \max_{s\in\mathcal{S}} \|s\|$. Suppose that the reward function satisfies **one of** the following conditions:*

1) *R(s,a,s')=R(s) depends solely on the current state s, and $R(s)$ is K-Lipschitz with respect to s;*
2) *R(s,a,s')=R(s') depends solely on the next state s', and $R(s')$ is K-Lipschitz with respect to s'.*

For any two policies π_1 and π_2, their performance difference can be bounded as follows:

$$|J(\pi_1) - J(\pi_2)| \leq \frac{K}{1-\gamma} \cdot \sum_{i=1}^{D}\left[\left\|F_{\pi_1}^{(i)} - F_{\pi_2}^{(i)}\right\|_\infty + 8A\sqrt{\left\|F_{\pi_1}^{(i)} - F_{\pi_2}^{(i)}\right\|_\infty}\right], \quad (11)$$

where $\|F\|_\infty = \sup_{\boldsymbol{\omega}\in[0,2\pi]^D} |F(\boldsymbol{\omega})|$ denotes the DTFT of the time series $\mathbf{S}^{(k)} = \{\mathbf{s_0}^k, \mathbf{s_1}^k, \mathbf{s_2}^k, \ldots\}$ for any integer $k \in [1, n]$ and $\mathbf{S}^{(k)}$ means the kth power of the state sequence produced by the policy π. The dimensionality of ω is the same as s.

Remark 2. Compared to the theorem presented in our conference paper [22], we relax the assumptions regarding the reward function. Specifically, we extend the assumption from a n-th polynomial function to a K-Lipschitz function and consider an additional case of $R(s, a, s') = R(s')$. These modifications notably broaden the applicability of the theorem, specifically making it more encompassing in goal-conditioned RL settings.

We provide the proof in Appendix A.1.2. Similar to the analysis of Theorem 1 and Thorem 2, the above theorem shows that state sequences in the frequency domain can indicate the policy performance and can be leveraged to enhance the search for optimal policies. Furthermore, the Fourier transform can decompose the state sequence signal into multiple physically meaningful components. This operator enables the analysis of time-domain signals in a higher dimensional space, making it easier to distinguish between two segments of signals that appear similar in the time domain.

Moreover, periodic signals have distinctive characteristics in their Fourier transforms, which are expressed as discrete spectra. We provide a visualization of the DTFT of the state sequences in Appendix F, which reveals that the DTFT of the periodic state sequence is approximately discrete. This observation suggests that the periodic information of the signal can be explicitly extracted in the frequency domain, particularly for the periodic cases provided by Theorem 3. For non-periodic cases, the frequency domain still retains complete information about the state signals without any information loss due to the reversibility of the Fourier transform. In Section 6.2.2, we test our method on Arm Manipulation tasks to show the effectiveness of our method in non-periodic environments.

In addition to those advantages, the operation of the Fourier transforms also yields a concise auxiliary objective similar to the TD-error loss [21], which we will discuss in detail in the following section.

5.2 Learning Objective of SPF

In this part, we propose our method, **S**tate Sequences **P**rediction via **F**ourier Transform (SPF), and describe how

to utilize the frequency domain of state sequences to learn an expressive representation. Specifically, our method performs an auxiliary self-supervision task by predicting the discrete-time Fourier transform (DTFT) of infinite-step state sequences to capture the long-term structural features in the state sequences, thus improving the sample efficiency of representation learning.

Now we model the auxiliary self-supervision task. Given the current observation s_t and the current action a_t, we define the expectation of future state sequence $\widetilde{s}_t$ over infinite horizon as

$$[\widetilde{s}_t]_n = [\widetilde{s}(s_t,a_t)]_n = \begin{cases} \gamma^n \mathbb{E}_{\pi,p}\left[s_{t+n+1}\middle|s_t,a_t\right] & n=0,1,2,\dots \\ 0 & n=-1,-2,-3\dots \end{cases}. \tag{12}$$

Then the DTFT of $\widetilde{s}_t$ is $\mathcal{F}\widetilde{s}_t(\omega)=\sum_{n=0}^{+\infty}[\widetilde{s}_t]_n e^{-j\omega n}$, where ω represents the frequency variable. The discount factor γ in (12) is used to ensure the convergence of the Fourier transform and also serves as the contraction factor in the following Theorem 5. Since the state sequences are discrete-time signals, the corresponding DTFT is 2π-periodic with respect to ω. Based on this property, a common practice for operational feasibility is to compute a discrete approximation of the DTFT over one period, by sampling the DTFT at discrete points over $[0,2\pi]$. In practice, we take L equally-spaced samples of the DTFT. Then the prediction target is a matrix with size $L*D$, where D is the dimension of the state space. We can derive that the DTFT functions at successive time steps are related to each other in a recursive form:

$$F_{\pi,p}(s_t,a_t) = \widetilde{\boldsymbol{S}}_t + \Gamma\,\mathbb{E}_{\pi,p}\left[F_{\pi,p}(s_{t+1},a_{t+1})\right], \tag{13}$$

where

$$\widetilde{\boldsymbol{S}}_t = [[\widetilde{s}_t]_0,\dots,[\widetilde{s}_t]_0]^{\mathsf{T}} \in \mathbb{R}^{L\times D},$$

$$\Gamma = \gamma \begin{bmatrix} 1 & & & & \\ & e^{-j\frac{2\pi}{L}} & & & \\ & & e^{-j\frac{4\pi}{L}} & & \\ & & & \ddots & \\ & & & & e^{-j\frac{(L-1)\pi}{L}} \end{bmatrix}.$$

Based on the recursive formula (13), we can obtain the prediction loss by computing the difference between the estimated Fourier value $F_{\pi,p}(s_t,a_t)$ and the better estimate $\widetilde{\boldsymbol{S}}_t + \Gamma\,\mathbb{E}_{\pi,p}\left[F(s_{t+1},a_{t+1})\right]$, just like the TD error. Thus, our auxiliary objective is defined as follows:

$$L_{\text{pred}} = \mathbb{E}_{\substack{(s,a,s')\sim\mathcal{D}\\ a'\sim\pi}}\left[\widetilde{\boldsymbol{S}}_t + \Gamma F_\pi(s',a') - F_\pi(s,a)\right], \tag{14}$$

where $\mathcal{D}$ is the replay buffer.

Similar to the TD-learning of value functions, the recursive relationship can be reformulated as contraction mapping $\mathcal{T}$, as shown in the following theorem (see proof in Appendix A.3).

Theorem 5. *Let $\mathcal{F}$ denote the set of all functions $F:\mathcal{S}\times\mathcal{A}\to\mathbb{C}^{L*D}$ and define the norm on $\mathcal{F}$ as*

$$\|F\|_{\mathcal{F}} := \sup_{\substack{s\in\mathcal{S}\\ a\in\mathcal{A}}} \max_{0\le k<L} \left\|\left[F(s,a)\right]_k\right\|_D,$$

where $\left[F(s,a)\right]_k$ represents the kth row vector of $F(s,a)$. We show that the mapping $\mathcal{T}:\mathcal{F}\to\mathcal{F}$ defined as

$$\mathcal{T}F(s_t,a_t) = \widetilde{\boldsymbol{S}}_t + \Gamma\,\mathbb{E}_{\pi,p}\left[F(s_{t+1},a_{t+1})\right] \tag{15}$$

is a contraction mapping, where $\widetilde{\boldsymbol{S}}_t$ and Γ are defined in Appendix A.3.

Due to the properties of contraction mappings, we can iteratively apply the operator $\mathcal{T}$ to compute the target DTFT function of long-term state sequences until convergence in tabular settings. When calculating the prediction loss, we only need to utilize the current state s_t, the current action a_t, and the next state s_{t+1}. Therefore, one notable advantage of SPF is that there is no need to store multi-step future states as labels for predicting future state sequences.

In practice, we update the representation by minimizing the auxiliary objective (14). Based on Theorem 4, we propose a bounded sub-optimality theorem to bridge the gap between the motivation to leverage the frequency distribution of state sequences and the implementation of our method.

Theorem 6 (Sub-optimality Bound on Policy Performance)**.** *Let $\mathcal{S}\subset\mathbb{R}^D$ and $A=\max_{s\in\mathcal{S}}\|s\|$. Suppose that the reward function satisfies **one of** the following conditions:*

1) *R(s,a,s')=R(s) depends solely on the current state s, and $R(s)$ is K-Lipschitz with respect to s;*
2) *R(s,a,s')=R(s') depends solely on the next state s', and $R(s')$ is K-Lipschitz with respect to s'.*

Then the performance difference between the optimal policy π^* and the latent policy $\overline{\pi}\circ\phi$ may be bounded as:

$$|J(\pi^*) - J(\overline{\pi}\circ\Phi)| \le \frac{K}{1-\gamma}\cdot \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\overline{\delta}_{\overline{\pi}\circ\phi}(s,a,s') + 8A\sqrt{\overline{\delta}_{\overline{\pi}\circ\phi}(s,a,s')}\right], \tag{16}$$

where $\overline{\delta}_\pi(s,a,s') := s' + e^{-j\omega}\gamma\mathbb{E}_{a'\sim\pi}\left[F_\pi(s',a')\right] - F_\pi(s,a)$ is the minimized objective of SPF.

The above theorem demonstrates that the performance of the latent policy on the representation space learned by SPF could approach the performance of the optimal policy as we minimize the objective of SPF. We provide the proof in Appendix A.4.

As the Fourier transform of the real state signals has the property of conjugate symmetry, we only need to predict the DTFT on a half-period interval $[0,\pi]$. Therefore, we reduce the row size of the prediction target by half to reduce redundant information and save storage space. In practice, we train a parameterized prediction model $\mathcal{F}$ to predict the DTFT of state sequences. Note that the value of the prediction target is on the complex plane, so the prediction network employs two separate output modules $\mathcal{F}_{\text{Re}}$ and $\mathcal{F}_{\text{Im}}$ as real and imaginary parts respectively. Then we define the

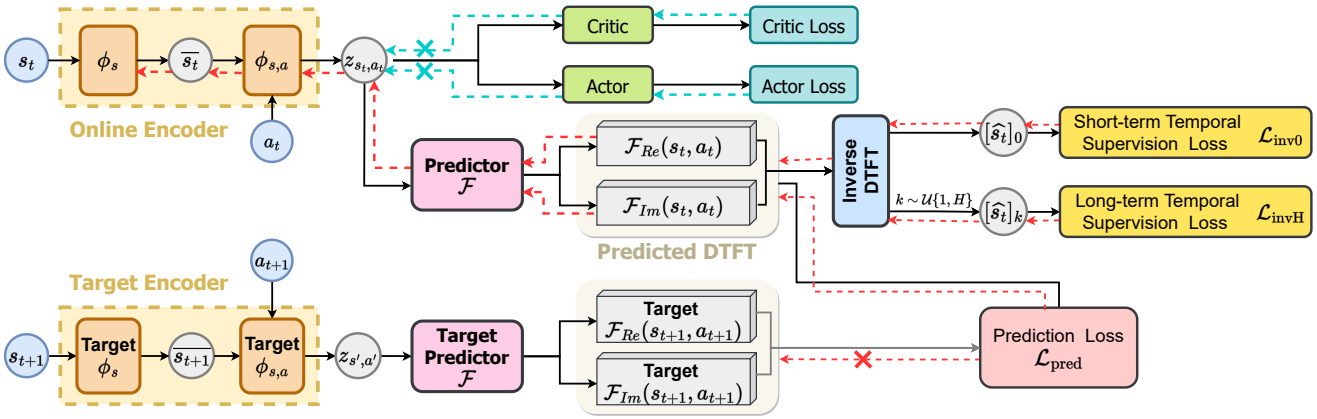


Fig. 2: **The network architecture of SPF**. The online encoder ϕ outputs the representations used in the RL task and the predictor $\mathcal{F}$ predicts the complex-valued Fourier transform (FT) of the state sequences starting from the state-action pair (s_t, a_t). The target encoder and the target predictor provide a better estimate of FT for our TD-style prediction loss $\mathcal{L}_{\text{pred}}$. Through the inverse DTFT, we transform the predicted FT into the time domain and leverage the recovered state signals to compute the two temporal supervision losses $\mathcal{L}_{\text{inv0}}$ and $\mathcal{L}_{\text{invH}}$. During training, (s_t, a_t, s_{t+1}) are previously experienced states and actions sampled from a replay buffer. The dashed line shows how gradients flow back to model weights. We prevent the gradient of RL losses from updating the online encoder and prevent the gradient of prediction loss from updating the target encoder and the target predictor.

auxiliary prediction loss function as:

$$
\begin{aligned}
L_{\text{pred}}(\phi, \mathcal{F}) = d\Big(\widetilde{\boldsymbol{S}}_t + \big[\Gamma_{\text{Re}}\mathcal{F}_{\text{Re}}(\overline{s_{t+1}}, \pi(\overline{s_{t+1}})) \\
- \Gamma_{\text{Im}}\mathcal{F}_{\text{Im}}(\overline{s_{t+1}}, \pi(\overline{s_{t+1}}))\big],\ \mathcal{F}_{\text{Re}}(\overline{s_t}, a_t)\Big) \\
+ d\Big(\big[\Gamma_{\text{Im}}\mathcal{F}_{\text{Re}}(\overline{s_{t+1}}, \pi(\overline{s_{t+1}})) \\
+ \Gamma_{\text{Re}}\mathcal{F}_{\text{Im}}(\overline{s_{t+1}}, \pi(\overline{s_{t+1}}))\big],\ \mathcal{F}_{\text{Im}}(\overline{s_t}, a_t)\Big),
\end{aligned} \tag{17}
$$

where $\overline{s_t} = \phi(s_t)$ means the representation of the state, Γ_{Re} and Γ_{Im} denote the real and imaginary parts of the constant Γ, and d denotes an arbitrary similarity measure. We choose d as cosine similarity in practice. The training procedure of SPF is shown in the pseudo-code of Algorithm 1.

5.3 Improved Objective with Temporal Supervision

In this section, we consider improving and calibrating the prediction in the frequency domain by adding supervision in the time domain to the original SPF.

In practice, the auxiliary loss (14) suffers from the prone to local optimal solution due to the approximation operation of cosine similarity metric and discretized frequency domain. Therefore, we propose **S**tate Sequences **P**rediction in both the **F**requency domain and **T**ime domain–**SPFT**, an enhanced version of SPF. SPFT introduces additional constraints in the time domain to improve and calibrate the Fourier prediction. Specifically, in addition to directly predicting the Fourier Transform of state sequences, SPFT transforms the predicted frequency domain back into the time domain using Equation (7) and minimizes the error loss between the inversely transformed state signals and the original state signals saved in the replay buffer. Based on Equation (7), we denote the recovered state signals as follows:

$$
[\widehat{s}_t]_n = \frac{1}{\gamma^k L} \sum_{k=0}^{L-1} [F_\pi(s_t, a_t)]_k \, e^{j\frac{2\pi k}{L}n}, \tag{18}
$$

where $[F_\pi(s_t, a_t)]_k$ is the k-th row of the predicted Fourier

Algorithm 1 SPF Framework

Require: Denote parameters of the online encoder $(\phi_s, \phi_{s,a})$ and predictor $\mathcal{F}$ as θ_{aux};
Denote parameters of the target encoder $(\widehat{\phi}_s, \widehat{\phi}_{s,a})$ and predictor $\widehat{\mathcal{F}}$ as $\widehat{\theta}_{\text{aux}}$;
Denote parameters of actor model π and critic model Q for RL agents as θ_{RL};
Denote the smoothing coefficient and update interval for target network updates as τ and K.
Initialize replay buffer $\mathcal{D}$ and parameters $\theta_{\text{aux}}, \theta_{\text{RL}}$;
Warmup the predictor model $\mathcal{F}$ by minimizing L_{pred} in Eq. (17) with random samples;
for each environment step t **do**
  $a_t \sim \pi(\cdot|\phi_s(s_t))$;
  $s_{t+1}, r_{t+1} \sim p(\cdot|s_t, a_t)$;
  $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, s_{t+1}, r_{t+1})$;
  Sample a minibatch of $\{(s_t, a_t, s_{t+1}, r_{t+1})\}$ from $\mathcal{D}$;
  $\theta_{\text{aux}} \leftarrow \theta_{\text{aux}} - \alpha_{\text{aux}} \nabla_{\theta_{\text{aux}}} L_{\text{pred}}(\theta_{\text{aux}}, \widehat{\theta}_{\text{aux}})$;
  Resampling a minibatch of $\{(s_t, a_t, s_{t+1}, r_{t+1}) \sim \mathcal{D}$;
  $\overline{s_t} \leftarrow \phi_s(s_t)$;
  $z_{s_t,a_t} \leftarrow \phi_{s,a}(\phi_s(s_t), a_t)$;
  Update the RL agent parameters θ_{RL} with the representations $\overline{s_t}$, z_{s_t,a_t};
  Update parameters of target networks with $\widehat{\theta}_{\text{aux}} \leftarrow \tau\theta_{\text{aux}} + (1-\tau)\widehat{\theta}_{\text{aux}}$ every K steps;
end for

metric $F_\pi(s_t, a_t)$.

The temporal supervision of SPFT includes two types, short-term and long-term calibrations. For short-term calibration, we employ the first inversely state signal obtained after applying inverse DTFT as the supervision target:

$$L_{\text{inv0}} = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}}\big[\,\|[\widehat{s_t}]_0 - s_{t+1}\|_{l_2}^2\,\big].$$

For long-term calibration, we define the maximum step of the future state signals used for calibration as *inverse horizon* H, and uniformly select one of the future states within the timestep range of $\{1, 2, \cdots, H\}$ as the supervision target:

$$L_{\text{invH}} = \mathbb{E}_{\substack{k\sim\mathcal{U}\{1,H\}\\(s_t,a_t,s_{t+k+1})\sim\mathcal{D}}}\Big[\,\|[\widehat{s_t}]_k - s_{t+k+1}\|_{l_2}^2\,\Big].$$

By integrating the above two temporal supervision losses with our original prediction loss, we formulate the learning objective of our model, SPFT, as follows:

$$L_{\text{SPFT}} = L_{\text{pred}} + L_{\text{inv0}} + L_{\text{invH}} \tag{19}$$

We typically utilize the HER strategy for goal relabeling. Furthermore, to evaluate the quality of our learned prediction model, we introduce a novel relabeling method called *SPFT-spfLabel*. This method replaces the goal with the future state computed from the predicted frequency domain to increase the diversity of goals selected by HER. In practice, we perform a pre-relabeling step on the minibatch data using *future strategy* HER with a probability of p_{future}. Subsequently, we select a portion of tuples from the minibatch data with a probability of p_{spfLabel}. For each selected tuple $(s_t^i, a_t^i, s_{t+1}^i, r_{t+1}^i, g^i)$, we randomly sample a future step k within the maximum episode length and relabel g^i with $\phi_{\text{goal}}([\widehat{s_t^i}]_k)$, where $[\widehat{s_t^i}]_k$ is computed using equation (18). The training procedure of SPFT implemented in the setting of GCRL is provided in Algorithm 2.

5.4 Network Architecture of SPFT

Here we provide more details about the network architecture of our method. Following the prior work [14], [55], we compute target representations and target values of the predicted DTFT using a *target encoder* and a *target predictor* for more stable performance. We periodically overwrite the target network parameters with an exponential moving average of the online network parameters to stabilize training [1].

Compared to our conference paper [22], we have made some minor modifications to our approach. Specifically, we have removed the projection head that was used to deduce the dimension of the mid-range frequency band in the predicted DTFT. This change was made because this module was found to have a marginal impact on performance improvement.

During the training process, we optimize the encoder ϕ and the predictor $\mathcal{F}$ by minimizing the summation of the auxiliary prediction loss $\mathcal{L}_{\text{pred}}$ and two temporal supervision losses $\mathcal{L}_{\text{inv0}}$ and $\mathcal{L}_{\text{invH}}$. We concurrently update the actor-critic models of RL tasks with the output representations of the trained encoder ϕ. We illustrate the overall architecture of SPF and the gradient flows during training in Fig. 2.

Algorithm 2 SPFT with Goal Relabeling

Require: Model parameters $\theta_{\text{aux}}, \widehat{\theta}_{\text{aux}}, \theta_{\text{RL}}$ (same definition as in Algorithm 1), smoothing coefficient τ and update interval K for target networks, batch size N.

Initialize replay buffer $\mathcal{D}$ and parameters $\theta_{\text{aux}}, \theta_{\text{RL}}$;
Warmup the predictor model $\mathcal{F}$ by minimizing L_{SPFT} in Eq. (19) with random samples;
for $episode = 1, 2, \cdots, M$ **do**
  Sample a desired goal g;
  Collect a trajectory τ with the policy π;
  $\mathcal{D} \leftarrow \tau \cup \mathcal{D}$;
  Sample a minibatch b from the replay buffer: $\{(s_t^i, a_t^i, s_{t+1}^i, r_{t+1}^i, g^i)\}_{i=1}^N \sim \mathcal{D}$;
  for $i = 1, 2, \cdots, N$ **do**
    Relabel g^i with $\phi_{\text{goal}}(s_{t+k})$ by *future strategy* HER;
    Recompute r_{t+1}^i using reward function in Eq. (3);
  end for
  Update the encoder and predictor model with b: $\theta_{\text{aux}} \leftarrow \theta_{\text{aux}} - \alpha_{\text{aux}} \nabla_{\theta_{\text{aux}}} L_{\text{SPFT}}(\theta_{\text{aux}}, \widehat{\theta}_{\text{aux}})$
  Resample a minibatch b' from the replay buffer: $\{(s_t^i, a_t^i, s_{t+1}^i, r_{t+1}^i, g^i)\}_{i=1}^N \sim \mathcal{D}$;
  for $i = 1, 2, \cdots, N$ **do** ▷ spfLabel
    Relabel g^i with $\phi_{\text{goal}}([\widehat{s_t^i}]_k)$ derived from Eq. (18);
    Recompute r_{t+1}^i using reward function in Eq. (3);
  end for
  update the RL agent parameters θ_{RL} with b';
  Update parameters of target networks with $\widehat{\theta}_{\text{aux}} \leftarrow \tau\theta_{\text{aux}} + (1-\tau)\widehat{\theta}_{\text{aux}}$ every K steps;
end for

6 Experiments

We conduct experiments to demonstrate the effectiveness of our proposed representation method in capturing long-term future information. In particular, 1) we evaluate the sample efficiency and the overall performance of SPF and SPFT on the standard and goal-reaching RL benchmarks, respectively. 2) We design ablation experiments to illustrate the effects of multiple components, hyperparameters, and variants of future prediction tasks. 3) We provide visualizations of the recovered state sequences derived from the predicted frequency domain to show the quality of our prediction model.

6.1 Environment Setup

For the standard RL benchmark, we employ six standard MuJoCo environments [56] implemented in OpenAI Gym [57]. For the goal-reaching RL benchmark, we consider two quadrupedal robot locomotion tasks [47], [58] and three arm manipulation tasks [59]. The details of each benchmark are as follows:

MuJoCo. We test our approach on six MuJoCo tasks: HalfCheetah-v2, Hopper-v2, Walker2d-v2, Ant-v2, Swimmer-v2, and Humanoid-v2. In each task, the agent is a robot composed of several links and joints. At every time step, the agent performs a continuous control that determines the torque applied to each joint. Each episode in these environments consists of 1000 timesteps. We train each

TABLE 1: **Mean and standard error results on six MuJoCo tasks at 500K step and 1M step**. All means and standard errors are calculated over 10 seeds. The highest mean scores are bolded. SPF outperforms other SOTA methods in all 6 MuJoCo tasks with an average 19.5% boost at 500K step and a 14.5% boost at 1M step.

	Environment	**SAC-SPF**	SAC-OFE	SAC-raw	**PPO-SPF**	PPO-OFE	PPO-raw
500K step	HalfCheetah	$\mathbf{12426 \pm 117}$ $(+4\%)$	11932 ± 75	9004 ± 150	$\mathbf{2569 \pm 282}$ $(+6\%)$	2419 ± 296	1475 ± 238
	Hopper	$\mathbf{3402 \pm 30}$ $(+14\%)$	2983 ± 123	2343 ± 183	$\mathbf{2484 \pm 289}$ $(+15\%)$	1529 ± 157	2156 ± 245
	Walker2d	$\mathbf{4698 \pm 73}$ $(+25\%)$	3762 ± 162	2023 ± 269	$\mathbf{1374 \pm 267}$ $(+66\%)$	339 ± 32	827 ± 156
	Ant	$\mathbf{6744 \pm 48}$ $(+21\%)$	5587 ± 253	2691 ± 76	844 ± 32	696 ± 47	$\mathbf{929 \pm 15}$
	Swimmer	$\mathbf{46 \pm 0}$ $+2\%)$	45 ± 0	42 ± 0	$\mathbf{97 \pm 9}$ $(+24\%)$	78 ± 12	66 ± 6
	Humanoid	$\mathbf{6187 \pm 127}$ $(+2\%)$	6090 ± 120	1989 ± 220	388 ± 12	$\mathbf{405 \pm 18}$	383 ± 9
1M step	HalfCheetah	$\mathbf{15426 \pm 186}$ $(+7\%)$	14425 ± 112	10745 ± 159	$\mathbf{3238 \pm 407}$ $(+6\%)$	3066 ± 326	2259 ± 344
	Hopper	$\mathbf{3470 \pm 43}$ $(+9\%)$	3197 ± 147	3056 ± 91	$\mathbf{2775 \pm 267}$ $(+2\%)$	2370 ± 258	2721 ± 222
	Walker2d	$\mathbf{5206 \pm 90}$ $(+8\%)$	4833 ± 59	3367 ± 110	$\mathbf{2389 \pm 247}$ $(+4\%)$	1080 ± 236	2302 ± 158
	Ant	$\mathbf{7381 \pm 36}$ $(+10\%)$	6738 ± 150	4220 ± 164	$\mathbf{1259 \pm 106}$ $(+38\%)$	913 ± 18	867 ± 17
	Swimmer	$\mathbf{46 \pm 0}$ $(+2\%)$	45 ± 0	43 ± 0	$\mathbf{105 \pm 11}$ $(+27\%)$	73 ± 10	83 ± 9
	Humanoid	$\mathbf{8156 \pm 203}$ $(+20\%)$	7999 ± 87	4997 ± 216	$\mathbf{469 \pm 16}$ $+5\%)$	448 ± 17	439 ± 18

environment for three million timesteps, except for Hopper-v2, for which we record only one million timesteps due to its quickly converging training process.

Robot locomotion tasks. These two environments involve controlling an Ant robot to reach a target position within the square $[-5, 5]^2$. They are implemented based on the Ant environment in MuJoCo, and the difference is that the state space of the goal-reaching version includes two additional goal dimensions, the X and Y position of the agent's torso. We evaluate two robot locomotion tasks: Fixed Ant Locomotion and Diverse Ant Locomotion. The main difference between them is that the target position in Diverse Ant Locomotion is chosen from the box space $[-2, 2]^2$, while the target in Fixed Ant Locomotion is fixed to $(2, 2)$. In these two tasks, the maximum timesteps allowed to reach the desired goal is 100. The reward is a sparse indicator function being nonnegative only when the torso position of the ant is within 0.1 of the goal. The detailed implementations of these two environments are based on [47].

Arm manipulation tasks. Each environment in this benchmark involves a 7-DOF Fetch Robotics arm with a two-fingered parallel gripper as an end-effector, simulated using the MuJoCo physics engine in OpenAI Gym. We use three challenging tasks for evaluation, including FetchPush, FetchSlide, And FetchPickAndPlace. For FetchPush, a box is placed in front of the robot and the goal is to move the box to a target location on the table. For FetchSlide, a puck is placed on a long slippery table and the goal for the robot is to hit the puck with a force and make the puck slide and stop at the target location. For FetchPickAndPlace, a box is placed in front of the robot and the goal is to grasp the box and move it to the target location which may be on the table or in the air above the table. In these environments, the states in the simulation consist of positions, linear and angular velocities of all robot joints and an object. Goals represent the desired position of the object. We set a distance threshold of 0.05 to determine whether the robot has reached the goal. We consider sparse rewards: if the object is outside the distance threshold of the goal, the agent receives a reward signal of -1; otherwise, the reward signal is 0. The maximum timesteps allowed to reach the desired goal is 50.

6.2 Comparative Experiments

6.2.1 Standard MuJoCo Tasks

To evaluate the effect of learned representations, we measure the performance of two traditional RL methods SAC [60] and PPO [61] with raw states, OFENet representations [27], and SPF representations. We train our representations by predicting the Fourier transform of infinite-horizon state sequences, which means the representations learn to capture the information of infinite-step future states. Hence, we compare our representation with OFENet, which uses the auxiliary task of predicting single-step future states for representation learning. For a fair comparison, we use an encoder with the same network structure as that of OFENet. See Appendix B and C for more details about network architectures and hyperparameters setting. Note that we set the discount factor $\gamma = 0.98$ in the Humanoid environment, and $\gamma = 0.99$ in the other five environments, as we find that such configuration can significantly improve performance on the Humanoid task. We additionally provide sensitivity analyses for four sets of hyperparameters, examining how our methods respond to these parameters and providing underlying interpretations (see Appendix D for more details).

As described above, our comparable methods include: 1) **SAC-SPF**: SAC with SPF representations; 2) **SAC-OFE**: SAC with OFENet representations; 3) **SAC-raw**: SAC with raw states; 4) **PPO-SPF**: PPO with SPF representations; 5) **PPO-OFE**: PPO with OFENet representations; 6) **PPO-raw**: PPO with raw states. Compared to OFENet, SPF features two main novelties. First, SPF predicts infinite-step state sequences, which contain long-term features that are more conducive to far-sighted decision-making than the single-step states. Second, the prediction target of SPF is defined in the frequency domain, which provides an overall perspective for the encoder to capture the structural information of the sequential signals. In Fig. 3, SPF shows superior performance compared to the original algorithms, SAC-raw and PPO-raw, across all six MuJoCo tasks and also outperforms OFENet in terms of both sample efficiency and asymptotic performance on all six MuJoCo tasks. The results suggest that multi-step future prediction offers more informative

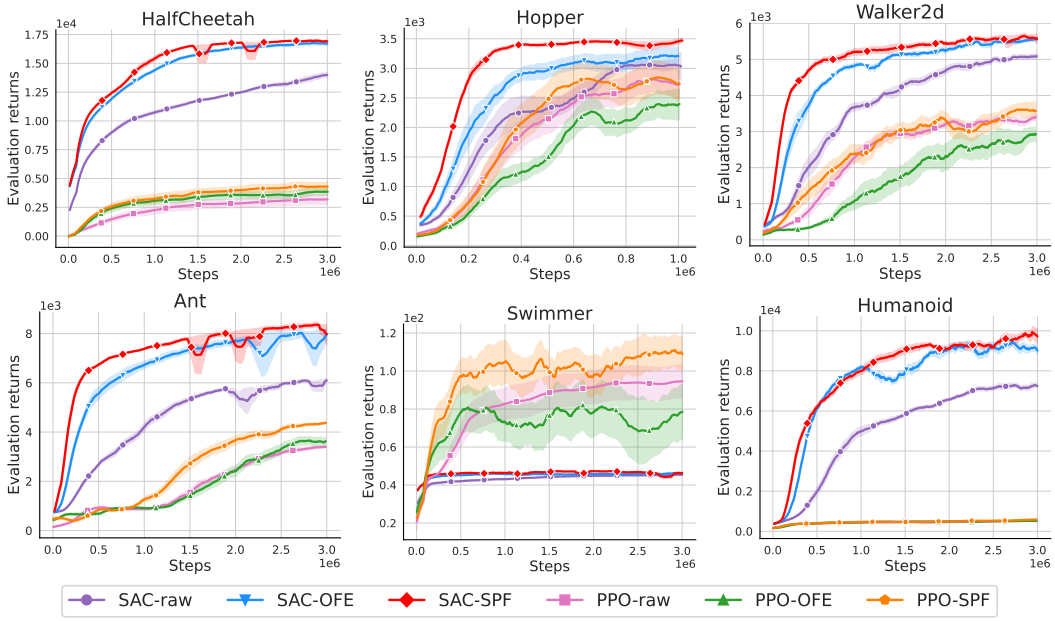


Fig. 3: **Results on six MuJoCo tasks**. The solid curves denote the means and the shaded regions denote the standard errors over 10 seeds. Each checkpoint is averaged by 10 episodes in evaluated environments. Curves are smoothed for visual clarity. We apply our representation method, SPF, on both off-policy (SAC) and on-policy (PPO) algorithms. We compare SPF (red and orange) with raw states (purple and pink) and OFENet representations (blue and green). Our results show that SPF outperforms the other methods in sample efficiency and asymptotic performance on all six MuJoCo tasks.

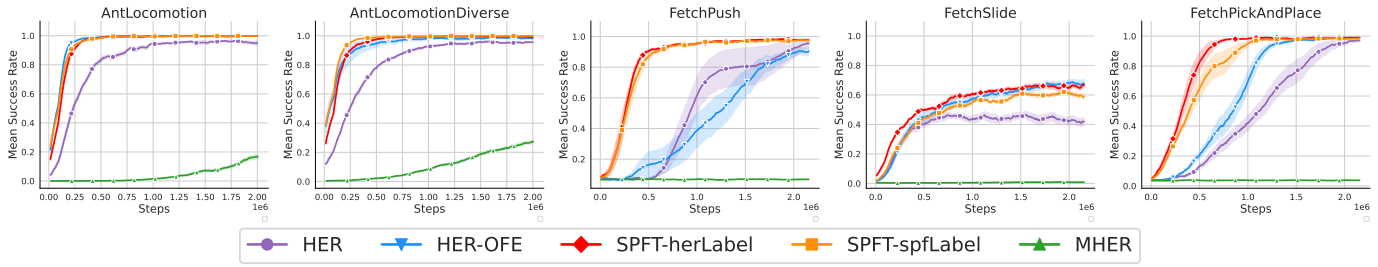


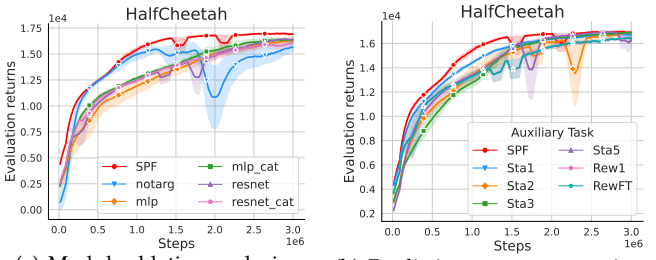
Fig. 4: **Results on two robot locomotion tasks and three robot arm manipulation tasks**. The solid curves denote the means and the shaded regions denote the standard errors over 10 seeds. Each checkpoint is averaged by 10 episodes in evaluated environments. Curves are smoothed for visual clarity. We apply our improved representation method (SPFT) to HER, the standard baseline in GCRL. We compare the performance of HER with raw state inputs (purple), OFENet representations (blue), and SPFT representations (red). We also evaluate two other goal-relabeling methods that replace goals with the state signals recovered from our predicted DTFT (orange) and those generated from a learned dynamic model (green). SPFT exhibits superior performance compared to the other methods.

signals than single-step prediction. According to the results in Table 1, SPF achieves an average gain of +19.5% over other methods at 500K step and +14.5% at 1M step. These results indicate that our approach learns more quickly than other methods when the number of interactions is limited.

6.2.2 *Goal-oriented Locomotion and Manipulation Tasks*

We integrate our improved representation method SPFT into DDPG+HER, a classic GCRL baseline that combines the goal-relabeling strategy HER [33] with the off-policy algorithm DDPG [62]. Following the evaluation for standard RL, we compare our representation method with OFENet to show the potential difference in information gains yielded by short-term and long-term future predictions. Furthermore, to assess the quality of our learned prediction model, we additionally provide a novel relabeling method called SPFT-spfLabel, which replaces the goal with the future state derived from the predicted frequency domain. We compare this relabeling method with SPFT-herLabel, which also utilizes the SPFT representation but employs HER strategy to relabel goals. The purpose is to verify whether the virtual distribution of state sequences generated by the predictor could achieve the effect of historical distribution in the replay buffer and even augment the diversity of selected goals. The encoder structure is the same as the configuration in the standard RL setting. See Appendix B and C for more details about network architectures and hyperparameters.

Our comparable methods for goal-reaching tasks include: 1) **HER**: standard DDPG with the HER relabeling strategy that replaces the desired goals with achieved goals



(a) Module ablation analysis (b) Prediction targets comparison

Fig. 5: **Results of additional trials on SPF**. **(a)** Module ablation analysis: training curves of five variants of SPF on SAC. Each method is evaluated over 5 seeds on HalfCheetah-v2. The results indicate that both the target networks and the encoder architecture significantly influence the performance. **(b)** Comparison of prediction targets: training curves of SAC with different auxiliary prediction tasks. Each method is evaluated over 5 seeds on HalfCheetah-v2. We observe that the frequency prediction for state sequences (SPF) outperforms the other competitors in sample efficiency.

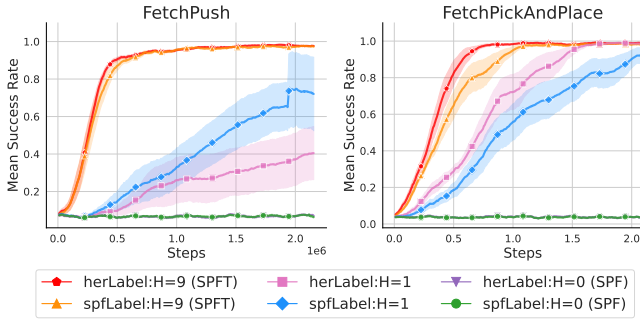


Fig. 6: **Module ablation analysis of SPFT**. We compare the training curves of SPFT (**H=9**) and its two variants: 1) **H=0**: without both short-term and long-term temporal prediction; 2) **H=1**: without long-term temporal prediction. Each method is evaluated over 10 seeds on FetchPush and FetchPickAndPlace. The results highlight the importance of adding short-term and long-term temporal supervisions to calibrate the frequency prediction, leading to more informative features and more accurate guidance in GCRL tasks.

in the same trajectory; 2) **HER-OFE**: DDPG with representations learned by OFENet, using HER for the strategy of goal relabeling; 3) **SPFT-herLabel**: DDPG with representations learned by SPFT, using HER for the strategy of goal relabeling; 4) **SPFT-spfLabel**: DDPG with representations learned by SPFT, using future states computed by the predicted DTFT as the relabeled goal; 5) **MHER** [46]: standard DDPG that relabels goals with virtual goals generated from the rollouts with a learned dynamics model.

We test the advantages of SPFT in two aspects. First, similar to SPF, SPFT introduces long-term features into representations via frequency prediction, allowing it to capture more future information than the other baselines that use raw states and OFENet representations. By comparing the curves labeled *HER* (purple), *HER-OFE* (blue), and *SPFT-herLabel* (red) in Fig. 4, we can observe the significant performance gains achieved by our learned representations. The results indicate that multi-step future prediction provides more informative signals than single-step prediction for GCRL tasks, which require foresighted decision-making. Second, we compare our proposed goal-relabeling strategy, SPFT-spfLabel, with two other strategies, HER and MHER. Compared to HER, our relabeling method prevents labeling homogeneous goals limited in historical data and may augment the diversity of selected goals. Compared to MHER, our relabeling method could mitigate concerns about error accumulation from the dynamics model. By comparing the curves labeled *SPFT-herLabel* (red), *SPFT-spfLabel* (orange), and *MHER* (green) in Fig. 4, we observe that SPFT-spfLabel exhibits superior performance to MHER while showing similar performance with SPFT-herLabel. It verifies the quality of our learned prediction model, where the virtual distribution of state sequences generated by the predictor could achieve the effect of the historical distribution.

6.3 Ablation Study

6.3.1 *Module Ablation Analysis on SPF*

In this part, we will verify that just predicting the FT of state sequences may fall short of the expected performance and that using SPF is necessary to get better performance. To this end, we conducted an ablation study to identify the specific components that contribute to the performance improvements achieved by SPF. Fig. 5(a) shows the ablation study over SAC with HalfCheetah-v2 environment.

notarg removes the target networks of both the encoder Φ and the predictor $\mathcal{F}$ from SPF. Based on the empirical results, the variant of SPF exhibits notably reduced performance, particularly in the later stages of training, when the auxiliary loss utilizes target estimations generated by the online encoder without a stop-gradient. These findings indicate that employing a separate target encoder is vital, as it can significantly improve the stability and convergence properties of our algorithm.

mlp changes the layer block of the encoder from MLP-DenseNet to MLP. The much lower scores of *mlp* indicate that both the raw state and the output of hidden layers contain important information that contributes to the quality of the learned representations. This result underscores the importance of leveraging sufficient information for representation learning.

mlp-cat uses a modified block of MLP as the layer of the encoder, which concatenates the output of MLP with the raw state. The performance of *mlp-cat* does increase compared to *mlp*, but is still not as good as SPF in terms of both sample efficiency and performance.

resnet changes the layer block of the encoder from MLP-DenseNet to MLP-ResNet. Based on the results, the *resnet* encoder shows inferior performance compared to the Densenet-style encoder, in terms of both the sample efficiency and convergence properties. This suggests that the rich feature hierarchies captured by the DenseNet-style encoder may be important to the effectiveness of representation learning.

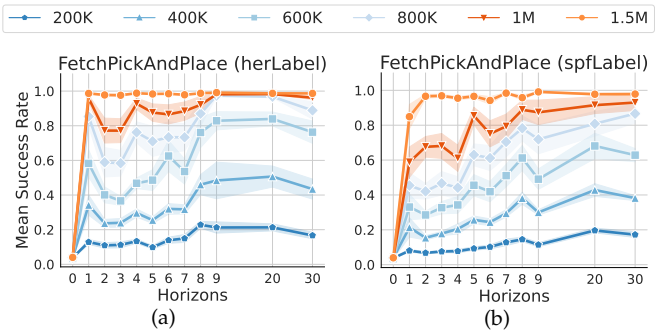


Fig. 7: **Study on inverse horizon.(a)** SPFT-herLabel. **(b)** SPFT-spfLabel. The x-axis is the maximum length for inverse prediction in the time domain, i.e., the inverse horizon H. The y-axis is the average success rate over 10 episodes. The color indicates the observed interaction timesteps. The confidence intervals denote the standard error over 10 seeds. Each method is evaluated on FetchPickAndPlace under the setting of $p_{\text{future}} = 0.8$ and $p_{\text{spfLabel}} = 0.5$. We observe a general trend of incremental performance improvement with increasing inverse horizon, peaking at $H = 20$.

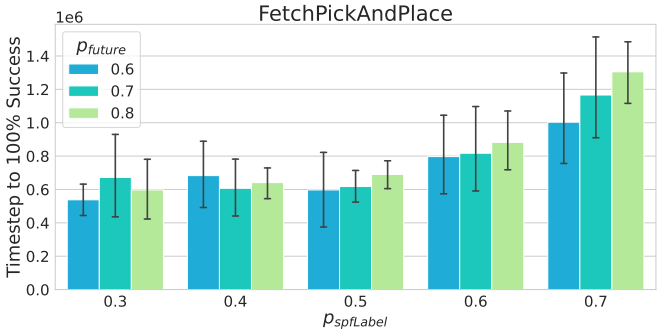


Fig. 8: **Study on Relabel Probability**. This grouped bar chart compares the interaction steps spent to achieve 100% success rate for different probabilities of relabel strategies. The x-axis depicts the probability of relabeling by recovered states of SPFT, p_{spfLabel}. Bar color indicates the probabilities of pre-relabeling by HER future strategy, p_{future}. The bar's value represents the average first timestep that achieves 100% success rate on FetchPickAndPlace over 5 seeds. The error bars denote the 85% confidence intervals over 5 seeds. We observe that sample efficiency is relatively insensitive to these two parameters in the lower range of $p_{\text{spfLabel}} \leq 0.5$, where our recovered state signals could serve as a reasonable substitute for the true state sequence distribution.

resnet-cat uses a modified block of MLP-ResNet as the layer of the encoder, which concatenates the output of MLP-ResNet with the raw state. While the performance of *resnet-cat* exhibits greater stability compared to the *resnet* encoder, it still falls short of the SPF method in terms of both sample efficiency and overall task performance.

6.3.2 Study on Auxiliary Prediction Targets

This section aims to test the effect of our prediction target—infinite-step state sequences in the frequency domain—on the efficiency of representation learning. We test five types of prediction targets: 1) **Sta1**: single-step future state; 2) **StaN**: N-step state sequences, where we choose $N = 2, 3, 5$; 3) **SPF**: infinite-step state sequences in frequency domain; 4) **Rew1**: single-step future reward; 5) **RewFT**: infinite-step reward sequences in frequency domain.

As shown in Fig. 5(b), SPF outperforms all other competitors in sample efficiency, which indicates that infinite-step state sequences in the frequency domain contain more underlying valuable information that can facilitate efficient representation learning. Since Sta1 and SPF outperform Rew1 and RewFT respectively, it can be referred that learning via states is more effective for representation learning than learning via rewards. Notably, the lower performance of StaN compared to Sta1 could be attributed to the model's tendency to prioritize prediction accuracy over capturing the underlying structured information in the sequential data, which may impede its overall learning efficiency.

6.3.3 Module Ablation Analysis on SPFT

In this part, we will analyze the role of the improved auxiliary loss augmented by inverse prediction in the time domain. To this end, we conduct an ablation study to identify the specific components that contribute to the performance improvements achieved by SPFT. We evaluate two versions of SPFT that combine with two different goal-relabeling strategies. Fig. 6 shows the ablation study for algorithms SPFT-herLabel and SPFT-spfLabel on the FetchPush and FetchPickAndPlace environments.

w/o *inv* & *invH* removes both the short-term and the long-term temporal supervision losses from SPFT. Note that such a variant is equal to the original version of SPF. Based on the results, this variant of SPFT fails on both FetchPush and FetchPickAndPlace environments. This suggests that the quality of the learned representation and the prediction module is poor without calibration in the time domain, which negatively impacts the training process. In Fig. 6, this variant corresponds to the legend of $H = 0$, which means that no inverse prediction was performed.

w/o *invH* removes the long-term temporal supervision loss L_{invH} from SPFT. Based on the results, such variant of SPFT exhibits significantly reduced performance. Therefore, multi-step calibration in the time domain is vital, which enhances the quality of the prediction module and incorporates more meaningful long-term features into the representation. In Fig. 6, this variant corresponds to the legend of $H = 1$, which means it only predicts one step future state for calibration.

6.3.4 Study on Inverse Horizon

We explore the effect of different inverse horizons H in this section. We assess SPFT-herLabel and SPFT-spfLabel on FetchPickAndPlace environment with $H = 0, 1, 2, 3, 4.5.6, 7, 8, 9, 20, 30$. Note that the special case of $H = 0$ and $H = 1$ represents the ordinary version of SPF and SPFT w/o invH, respectively.

As shown in Fig. 7, the sample efficiency of our method exhibits an overall upward trend as the inverse horizon increases on both SPFT-spfLabel and SPFT-herLabel. This suggests that multi-step future prediction with correction

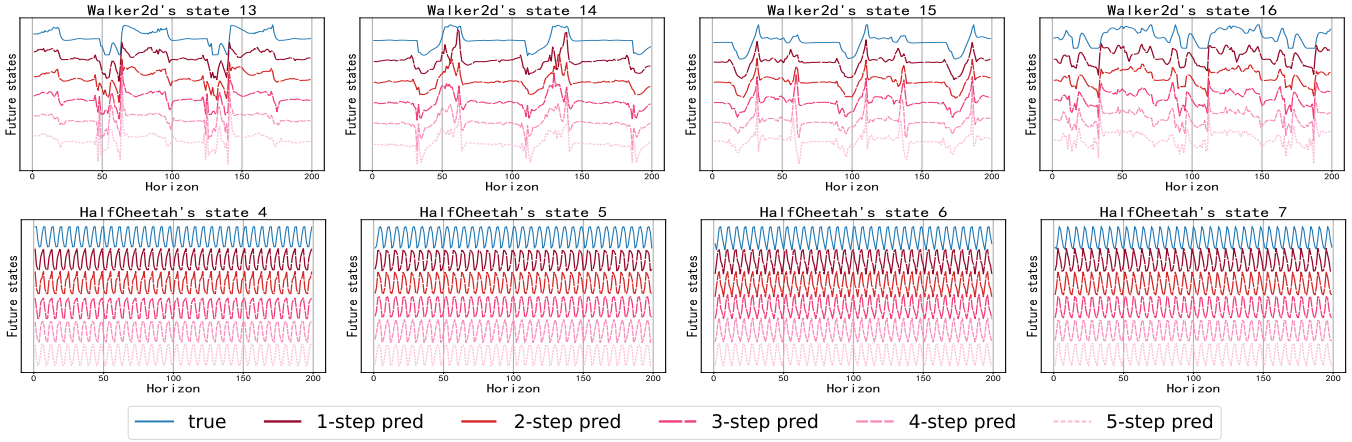


Fig. 9: **Visualization of recovered states from the predicted DTFT**. The blue line represents the true state sequence, while the red line represents the recovered state sequence. The lighter red line corresponds to predictions made by historical states from a more distant time step. These subfigures represent four sampled dimensions of the state spaces on Walker2d and HalfCheetah, respectively. We observe similar sequence patterns between the true and recovered state sequences, which indicates that our learned representations effectively capture the structural information of state sequences through frequency domain prediction.

in the time domain helps the agent to select actions more efficiently. However, the performance is observed to decline when the inverse horizon H exceeds a threshold of approximately 20, indicating that satisfactory results can be achieved by selecting an appropriate horizon value in practice. Additionally, SPFT-spfLabel appears to be more sensitive to the inverse horizon parameter compared to SPFT-herLabel.

6.3.5 *Study on Relabel Probability*

This section explores the effect of different relabel probabilities, including the probability p_{future} of using HER future strategy and the probability p_{spfLabel} of using recovered states by SPFT. To assess the impact of these two hyperparameters, we conduct an ablation study on FetchPickAndPlace environment, testing all combinations of the two parameter sets: $p_{\text{future}} \in \{0.6, 0.7, 0.8\}$ and $p_{\text{spfLabel}} \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$.

We use the interaction timesteps that a GCRL agent requires to achieve 100% success rate as the measurement for sample efficiency. As shown in Fig. 8, sample efficiency is relatively insensitive to these two parameters in the lower range of $p_{\text{spfLabel}} \leq 0.5$. However, it may negatively impact sample efficiency as the probability p_{spfLabel} increases beyond that range. These results suggest that the future states recovered by our predicted Fourier transform could serve as a reasonable substitute for the true state sequence distribution in guiding the GCRL agents' long-term decisions, provided the parameter p_{spfLabel} is maintained within the relatively low-sensitivity range of 0.5 or below.

6.4 Visualization

6.4.1 *Visualization of Recovered State Sequences*

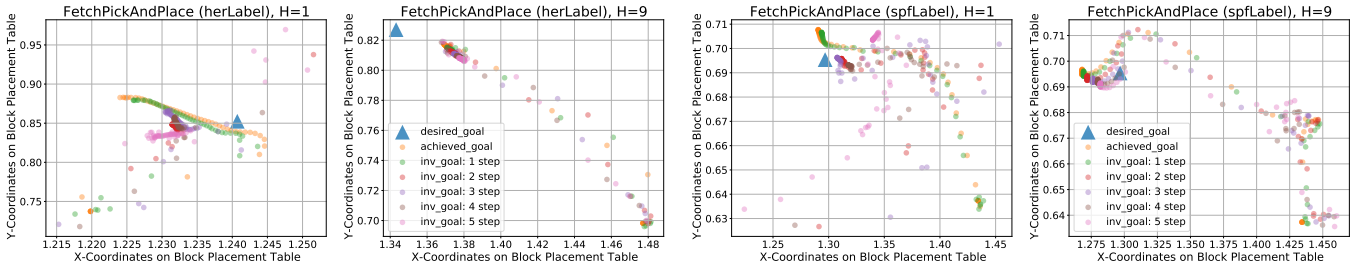
This section aims to demonstrate that the representations learned by SPF effectively capture the structural information contained in infinite-step state sequences. To this end, we compare the true state sequences with the states recovered from the predicted DTFT via the inverse DTFT (See Appendix F for more implementation details). Fig. 9 shows that the learned representations can recover the true state sequences even using the historical states that are far from the current time step. In Appendix F, we also provide a visualization of the predicted DTFT, which is less accurate than the results in Fig. 9. Those results highlight the ability of SPF to effectively extract the underlying structural information in infinite-step state sequences without relying on high prediction accuracy.

We further provide a comparison table that measures the distance between the real DTFT and the predicted DTFT using cosine similarity. The results provided in Appendix G indicate that the prediction module $\mathcal{F}$ exhibits moderate predictive accuracy in approximating the real Fourier transform, with an average cosine similarity value of -0.6 for SAC-SPF and -0.8 for PPO-SPF.

6.4.2 *Visualization of Inversed Goals*

In this section, we visualize different goals of GCRL tasks by illustrating the (x, y) coordinates of the block projection onto the placement table in the "FetchPickAndPlace" task. In Fig. 10, we display three types of goals generated in one episode. First, the blue triangle indicates the desired goal, which is the desired final position of the block on the table. Second, we use a series of yellow solid points to depict the block's position at each time step throughout the episode. Third, we use points in the other five colors to depict the kth-step ($k = 1, 2, \ldots, 5$) inversed goals derived from the predicted DTFT.

Specifically, we choose a historical state s_{t-k} from the whole episode and the corresponding action a_{t-k} selected by the trained policy $\pi(\cdot|s_{t-k})$. We use the state-action pair (s_{t-k}, a_{t-k}) as the input of the trained predictor to obtain the predicted DTFT. We then transform the DTFT back to the time domain, obtain the kth element of the inverse DTFT as the kth-step recovered state, and take the goal dimension from the recovered state as the kth-step inversed



(a) SPFT-herLabel with $H = 1$ (left) and $H = 9$ (right)

(b) SPFT-spfLabel with $H = 1$ (left) and $H = 9$ (right)

Fig. 10: **Visualization of three types of goals in one episode. (a)** SPFT-herLabel under the setting of $p_{\text{future}} = 0.8$. **(b)** SPFT-spfLabel under the setting of $p_{\text{future}} = 0.7$ and $p_{\text{spfLabel}} = 0.5$. We visualize the block positions on the table during an episode of FetchPickAndPlace. The x-axis and y-axis represent the x and y coordinates of the table plane where the block is placed, respectively. The desired goal, marked by a blue triangle, represents the desired final position of the block on the table. The achieved goal, marked by the yellow solid point, represents the current position of the block at each time step. The points in other colors indicate the kth-step ($k = 1, 2, \ldots, 5$) inversed goals derived from the predicted DTFT, using the encoder and predictor models trained by our methods. Specifically, we use SPFT-herLabel and SPFT-spfLabel as training methods, with inverse horizons set to $H = 1$ and $H = 9$.

goal. During training, we normalize the mini-batch data using the means and standard deviations of states and goals stored in the replay buffer. In the visualization graphs, we denormalize the inversed goals based on the statistics of achieved goals throughout the recorded episode. Therefore, we focus on comparing the trends and directions of both inversed and real achieved goal trajectories.

In Fig. 10, we use SPFT-herLabel and SPFT-spfLabel as the training methods. For each method, we compare the goal trajectories under two settings of inverse horizons, $H = 1$ and $H = 9$. The results show that with $H = 9$, the inversed goal trajectories closely align with the real achieved ones at each k-th step, whereas with $H = 1$, this similarity appears only at the first step. This suggests that providing temporal supervision over a longer horizon can improve the alignment between the distribution of goals generated by our trained predictor and that of real achieved goals. Furthermore, calibrating frequency predictions by adding multi-step supervision in the time domain may help the representation capture richer temporal dependencies, such as trends and directions, thereby providing more accurate guidance for future decisions in GCRL tasks.

7 Conclusion

In this paper, we theoretically analyzed the existence of structural information in state sequences, which is closely related to policy performance and signal regularity. We then introduced State Sequences Prediction via Fourier Transform (SPF), a representation learning method that predicts the FT of state sequences to extract the underlying structural information in state sequences for learning expressive representations efficiently. We also prove that the frequency prediction objective of SPF has a theoretical guarantee of bounded suboptimality. SPF outperforms several state-of-the-art algorithms in terms of both sample efficiency and performance. Our additional experiments and visualization show that SPF encourages representations to place a greater emphasis on capturing the underlying pattern of time-series data, rather than pursuing high accuracy of prediction tasks. Furthermore, we augment SPF with temporal supervision to calibrate the frequency prediction and extend the augmented method SPFT to goal-conditioned RL tasks where the state sequences do not exhibit obvious asymptotic periodicity. The empirical results demonstrate the efficiency of SPFT in handling non-periodic GCRL tasks. The visualization for goal trajectories illustrates that temporal supervision helps the representation to capture richer temporal dependencies, such as trends and directions, providing more accurate guidance for future decisions of GCRL agents.

Limitation. The key issue of our approach is that the extracted frequency features inherently depend on the policy and task, which limits their reusability across multiple tasks. Further research is needed to explore techniques that could promote the generalization of the learned representations. Furthermore, our methods may struggle with multi-stage tasks where the distributions of state sequences vary across different stages, resulting in limited periodicity. We provide additional results of SPF and SPFT on Adroit tasks [63] in Appendix E While these results show some performance improvement, our methods do not consistently outperform the other baselines across all four Adroit tasks. For Longer aperiodic tasks, such as *AntMaze* [64], our methods may benefit from combination with high-level planning techniques. The planning approach would help segment the task into stages with periodic behaviors or shorter horizons.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their insightful comments. This work was supported by National Key R&D Program of China under contract 2022ZD0119801, National Nature Science Foundations of China grants U23A20388, 62021001, U19B2026, and U19B2044.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] T. de Bruin, J. Kober, K. Tuyls, and R. Babuska, "Integrating state representation learning into deep reinforcement learning," *IEEE Robotics Autom. Lett.*, vol. 3, no. 2, pp. 1394–1401, 2018.

[3] T. D. Barrett, W. R. Clements, J. N. Foerster, and A. I. Lvovsky, "Exploratory combinatorial optimization with reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3243–3250.

[4] OpenAI, "GPT-4 technical report," *CoRR*, 2023.

[5] N. Botteghi, M. Poel, and C. Brune, "Unsupervised representation learning in deep reinforcement learning: A review," *CoRR*, 2022.

[6] X. Zhang, Y. Song, M. Uehara, M. Wang, A. Agarwal, and W. Sun, "Efficient reinforcement learning in block mdps: A model-free representation learning approach," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 26 517–26 547.

[7] A. Zhang, R. T. McAllister, R. Calandra, Y. Gal, and S. Levine, "Learning invariant representations for reinforcement learning without reconstruction," in *9th International Conference on Learning Representations (ICLR)*, 2021.

[8] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," in *5th International Conference on Learning Representations (ICLR)*, 2017.

[9] J. Munk, J. Kober, and R. Babuska, "Learning state representation for deep actor-critic control," in *55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 4667–4673.

[10] Z. Wang, J. Wang, Q. Zhou, B. Li, and H. Li, "Sample-efficient reinforcement learning via conservative model-based actor-critic," in *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 8612–8620.

[11] Z. Wang, T. Pan, Q. Zhou, and J. Wang, "Efficient exploration in resource-restricted reinforcement learning," in *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, B. Williams, Y. Chen, and J. Neville, Eds., 2023, pp. 10 279–10 287.

[12] R. Yang, J. Wang, Z. Geng, M. Ye, S. Ji, B. Li, and F. Wu, "Learning task-relevant representations for generalization via characteristic functions of reward sequence distributions," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2242–2252.

[13] J. Wang, R. Yang, Z. Geng, Z. Shi, M. Ye, Q. Zhou, S. Ji, B. Li, Y. Zhang, and F. Wu, "Generalization in visual reinforcement learning with the reward sequence distribution," 2023.

[14] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. C. Courville, and P. Bachman, "Data-efficient reinforcement learning with self-predictive representations," in *9th International Conference on Learning Representations, ICLR*, 2021.

[15] D. Hafner, T. P. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *Proceedings of the 36th International Conference on Machine Learning(ICML)*, vol. 97, 2019, pp. 2555–2565.

[16] Z. D. Guo, B. Á. Pires, B. Piot, J. Grill, F. Altché, R. Munos, and M. G. Azar, "Bootstrap latent-predictive representations for multitask reinforcement learning," in *Proceedings of the 37th International Conference on Machine Learning(ICML)*, vol. 119, 2020, pp. 3875–3886.

[17] N. Mishra, P. Abbeel, and I. Mordatch, "Prediction and control with temporal segment models," in *Proceedings of the 34th International Conference on Machine Learning(ICML)*, vol. 70, 2017, pp. 2459–2468.

[18] S. Moon and A. Helmy, "Understanding periodicity and regularity of nodal encounters in mobile networks: A spectral analysis," in *Proceedings of the Global Communications Conference. GLOBECOM*. IEEE, 2010, pp. 1–5.

[19] D. Wu, S. Li, J. Yang, and M. Sawan, "neuro2vec: Masked fourier spectrum prediction for neurophysiological representation learning," *arXiv preprint arXiv:2204.12440*, 2022.

[20] A. Tompkins and F. Ramos, "Fourier feature approximations for periodic kernels in time-series modelling," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 4155–4162.

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

[22] M. Ye, Y. Kuang, J. Wang, Y. Rui, W. Zhou, H. Li, and F. Wu, "State sequences prediction via fourier transform for representation learning," in *Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[23] Y. Guo, D. Zhou, X. Ruan, and J. Cao, "Variational gated autoencoder-based feature extraction model for inferring disease-mirna associations based on multiview features," *Neural Networks*, vol. 165, pp. 491–505, 2023. [Online]. Available: https://doi.org/10.1016/j.neunet.2023.05.052

[24] J. Xue, B. Wang, H. Ji, and W. Li, "Rt-transformer: retention time prediction for metabolite annotation to assist in metabolite identification," *Bioinform.*, vol. 40, no. 3, 2024. [Online]. Available: https://doi.org/10.1093/bioinformatics/btae084

[25] C. Lu, J. Yin, H. Yang, and S. Sun, "Enhancing multi-modal fusion in visual dialog via sample debiasing and feature interaction," *Inf. Fusion*, vol. 107, p. 102302, 2024. [Online]. Available: https://doi.org/10.1016/j.inffus.2024.102302

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

[27] K. Ota, T. Oiki, D. K. Jha, T. Mariyama, and D. Nikovski, "Can increasing input dimensionality improve deep reinforcement learning?" in *Proceedings of the 37th International Conference on Machine Learning, ICML*, vol. 119, 2020, pp. 7424–7433.

[28] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: contrastive unsupervised representations for reinforcement learning," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, vol. 119. PMLR, 2020, pp. 5639–5650.

[29] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[30] A. Anand, E. Racah, S. Ozair, Y. Bengio, M. Côté, and R. D. Hjelm, "Unsupervised state representation learning in atari," in *Advances in Neural Information Processing Systems 32: NeurIPS 2019*, 2019, pp. 8766–8779.

[31] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare, "Deepmdp: Learning continuous latent space models for representation learning," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, vol. 97. PMLR, 2019, pp. 2170–2179.

[32] P. S. Castro, T. Kastner, P. Panangaden, and M. Rowland, "Mico: Improved representations via sampling-based state similarity for markov decision processes," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, 2021, pp. 30 113–30 126.

[33] M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017, pp. 5048–5058.

[34] M. Liu, M. Zhu, and W. Zhang, "Goal-conditioned reinforcement learning: Problems and solutions," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, 2022, pp. 5502–5511.

[35] S. Nair, S. Savarese, and C. Finn, "Goal-aware prediction: Learning to model what matters," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 7207–7219.

[36] B. Eysenbach, T. Zhang, S. Levine, and R. Salakhutdinov, "Contrastive learning as goal-conditioned reinforcement learning," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS*, 2022.

[37] Q. Zou and E. Suzuki, "Contrastive goal grouping for policy generalization in goal-conditioned reinforcement learning," in *Neural Information Processing - 28th International Conference, ICONIP*, ser. Lecture Notes in Computer Science, vol. 13108, 2021, pp. 240–253.

[38] D. Ghosh, A. Gupta, and S. Levine, "Learning actionable representations with goal conditioned policies," in *7th International Conference on Learning Representations, ICLR*, 2019.

[39] P. Hansen-Estruch, A. Zhang, A. Nair, P. Yin, and S. Levine, "Bisimulation makes analogies in goal-conditioned reinforcement learning," in *International Conference on Machine Learning, ICML*, vol. 162, 2022, pp. 8407–8426.

[40] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, 2019, pp. 12 602–12 613.

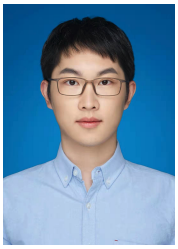
[41] Y. Luo, Y. Wang, K. Dong, Q. Zhang, E. Cheng, Z. Sun, and B. Song, "Relay hindsight experience replay: Self-guided continual

reinforcement learning for sequential object manipulation tasks with sparse rewards," *Neurocomputing*, vol. 557, p. 126620, 2023.
[42] Z. Ren, K. Dong, Y. Zhou, Q. Liu, and J. Peng, "Exploration via hindsight goal generation," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, 2019, pp. 13 464–13 474.
[43] P. Liu, C. Bai, Y. Zhao, C. Bai, W. Zhao, and X. Tang, "Generating attentive goals for prioritized hindsight reinforcement learning," *Knowl. Based Syst.*, vol. 203, p. 106140, 2020.
[44] R. Zhao and V. Tresp, "Energy-based hindsight experience prioritization," in *2nd Annual Conference on Robot Learning, CoRL*, ser. Proceedings of Machine Learning Research, vol. 87, 2018, pp. 113–122.
[45] R. Zhao and V. Tresp, "Curiosity-driven experience prioritization via density estimation," *CoRR*, vol. abs/1902.08039, 2019. [Online]. Available: http://arxiv.org/abs/1902.08039
[46] R. Yang, M. Fang, L. Han, Y. Du, F. Luo, and X. Li, "MHER: model-based hindsight experience replay," *CoRR*, vol. abs/2107.00306, 2021. [Online]. Available: https://arxiv.org/abs/2107.00306
[47] M. Zhu, M. Liu, J. Shen, Z. Zhang, S. Chen, W. Zhang, D. Ye, Y. Yu, Q. Fu, and W. Yang, "Mapgo: Model-assisted policy optimization for goal-oriented tasks," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*, 2021, pp. 3484–3491.
[48] S. Mahadevan, "Proto-value functions: Developmental reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 553–560.
[49] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. Wiley, 1994.
[50] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 22–31.
[51] O. Nachum and M. Yang, "Provable representation learning for imitation with contrastive fourier features," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, 2021, pp. 30 100–30 112.
[52] S. M. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Machine Learning, Proceedings of the Nineteenth International Conference*, 2002, pp. 267–274.
[53] A. Trott, S. Zheng, C. Xiong, and R. Socher, "Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, 2019, pp. 10 376–10 386.
[54] H. Scheffé, "A useful convergence theorem for probability distributions," *The Annals of Mathematical Statistics*, vol. 18, no. 3, pp. 434–438, 1947.
[55] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - A new approach to self-supervised learning," vol. 33, 2020, pp. 21 271–21 284.
[56] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *International Conference on Intelligent Robots and Systems, IROS*. IEEE, 2012, pp. 5026–5033.
[57] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540
[58] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, vol. 80. PMLR, 2018, pp. 1514–1523.
[59] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *CoRR*, vol. abs/1802.09464, 2018. [Online]. Available: http://arxiv.org/abs/1802.09464
[60] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning, ICML*, vol. 80, 2018, pp. 1856–1865.
[61] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, 2017.
[62] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*, 2016.
[63] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, Eds., 2018.
[64] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, vol. 48. JMLR.org, 2016, pp. 1329–1338.
[65] C. Villani *et al.*, *Optimal transport: old and new*. Springer, 2009, vol. 338.
[66] G. Grimmett and D. Stirzaker, *Probability and random processes*. Oxford university press, 2020.
[67] E. Seneta, *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006.
[68] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 1582–1591.

Jie Wang received the B.Sc. degree in electronic information science and technology from University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree in computational science from the Florida State University, Tallahassee, FL, in 2011. He is currently a Professor in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include reinforcement learning, knowledge graph, large-scale optimization, deep learning, etc. He is a Senior Member of IEEE.

Mingxuan Ye received the B.Sc. degree in information and computing sciences from Nanjing University, Nanjing, China, in 2018. She is currently an M.Sc. candidate in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. Her research interests include reinforcement learning.

Yufei Kuang received the B.Sc. degree in Statistics from Nanjing University, Nanjing, China, in 2020. He is currently a Ph.D. candidate in Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include reinforcement learning and learning to optimize.



Rui Yang received the B.Sc. degree in information and computing science from Hefei University of Technology, Hefei, China, in 2019. He is currently a Eng.D. candidate in the Department of Electronic Engineering and Information Science at University of Science and Technology of China, Hefei, China. His research interests include reinforcement learning and representation learning.

Wengang Zhou received the BE degree in electronic information engineering from Wuhan University, China, in 2006, and the PhD degree in electronic engineering and information science from the University of Science and Technology of China (USTC), China, in 2011. From September 2011 to 2013, he worked as a post-doctoral researcher with Computer Science Department, University of Texas, San Antonio. He is currently a professor with the EEIS Department, USTC. His research interests include multimedia information retrieval, computer vision, and computer game. In those fields, he has published more than 100 papers in ACM/IEEE Transactions and CCF Tier-A International Conferences. He is the recipient of the Best Paper Award for ICIMCS 2012. He served as the publication chair of IEEE ICME2021.

Houqiang Li received the BS, MEng, and PhD degrees in electronic engineering from the University of Science and Technology of China, Hefei, China, in 1992, 1997, and 2000, respectively, where he is currently a professor with the Department of Electronic Engineering and Information Science. His research interests include reinforcement learning and computer game, computer vision, image/video coding, etc.. He has authored and coauthored more than 200 papers in journals and conferences. He is the winner of National Science Funds (NSFC) for distinguished young scientists, the distinguished professor of Changjiang Scholars Program of China, and the leading scientist of Ten Thousand Talent Program of China. He is the associate editor (AE) of IEEE TMM, and served as the AE of IEEE Transactions on Circuits and Systems for Video Technology from 2010 to 2013. He served as the general co-chair of ICME 2021 and the TPC co-chair of VCIP 2010. He is the recipient of National Technological Invention Award of China (second class), in 2019 and the recipient of National Natural Science Award of China (second class), in 2015. He was the recipient of the Best Paper Award for VCIP 2012, the recipient of the Best Paper Award for ICIMCS 2012, and the recipient of the Best Paper Award for ACMMUM, in 2011.

Feng Wu received the B.Sc. degree in electronic engineering from Xidian University, Xi'an, China, in 1992, and received the M.Sc. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1996 and 1999, respectively. He is now a Professor and Vice President at the University of Science and Technology of China, Hefei, China. Previously, he was a Principle Researcher and Research Manager with Microsoft Research Asia, Beijing, China. His research interests include image and video compression, media communication, and media analysis and synthesis. He has authored or co-authored over 200 high-quality articles. His 15 techniques have been adopted into international video coding standards. He serves or had served as the Editor-in-Chief for IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) and as an Associate Editor for IEEE Transactions on Image Processing (TIP) and IEEE Transactions on Multimedia. He also serves as General Chair in ICME 2019, TPC Chair in MMSP 2011, VCIP 2010, and PCM 2009. He received the IEEE CAS Mac Van Valkenburg Award in 2021, the best paper awards in IEEE TCSVT 2009, VCIP 2016, PCM 2008, and VCIP 2007, and the Best Associate Editor Award of IEEE Transactions on Image Processing (TIP) in 2018. He is a Fellow of IEEE.