A Proof

A.1 Proof of Performance Difference Distinction via State Sequences

Following the previous work [19], our analysis will make use of the discounted future state distribution, d^π, which is defined as

$$d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi, \mathcal{M})$$

It allows us to express the expected discounted total reward compactly as

$$\begin{aligned}
J(\pi) &= \sum_{t=0}^{\infty}\gamma^t E_{s_t,a_t,s_{t+1}}\left[R(s_t,a_t,s_{t+1})|\pi,\mathcal{M}\right] \\
&= \sum_{t=0}^{\infty}\gamma^t \int_{\mathcal{S}} R_\pi(s) P(s_t = s|\pi,\mathcal{M})\,\mathrm{d}s \\
&= \int_{\mathcal{S}} R_\pi(s) \sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi,\mathcal{M})\,\mathrm{d}s \qquad (1)\\
&= \frac{1}{1-\gamma}\int_{\mathcal{S}} R_\pi(s) d_\pi(s)\,\mathrm{d}s \\
&= \frac{1}{1-\gamma} E_{\substack{s\sim d^\pi\\ a\sim\pi \\ s'\sim P}}\left[R(s,a,s')\right], \qquad (2)
\end{aligned}$$

where we define $R_\pi(s) := E_{a\sim\pi, s'\sim P}[R(s,a,s')]$. It should be clear from $a \sim \pi(\cdot|s)$ and $s' \sim P(\cdot|s,a)$ that a and s' depend on s. Thus, the reward function R_π is only related to s when the policy π is fixed.

Firstly, we prove that the distance between two state sequence distributions obtained from two distinct policies serves as an upper bound on the performance difference between those policies, provided that certain assumptions regarding the reward function hold.

Theorem 1. *Suppose that the reward function $R(s,a,s') = R(s)$ is related to the state 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:*

$$|J(\pi_1) - J(\pi_2)| \le \frac{R_{max}}{1-\gamma}\cdot \left\|P(s_0,s_1,s_2,\dots|\pi_1,\mathcal{M}) - P(s_0,s_1,s_2,\dots|\pi_2,\mathcal{M})\right\|_1, \qquad (3)$$

where $P(s_0,s_1,s_2,\dots|\pi_1,\mathcal{M})$ 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 $\mathcal{M}$.

Proof. According to the equation (1), the difference in performance between two policies π_1, π_2 can be bounded as follows.

$$\begin{aligned}
|J(\pi_1) - J(\pi_2)| \le R_{\max}\cdot\sum_{t=0}^{\infty}\gamma^t\int_{\mathcal{S}}\left|P(\mathbf{s_t}=s|\pi_1,\mathcal{M}) - P(\mathbf{s_t}=s|\pi_2,\mathcal{M})\right|\mathrm{d}s \\
\le R_{\max}\cdot\sum_{t=0}^{T}\gamma^t\int_{\mathcal{S}}\bigg|\int_{\mathcal{S}^T} P(s_0,\dots,s_{t-1},s,s_{t+1},\dots,s_T|\pi_1,\mathcal{M}) \\
- P(s_0,\dots,s_{t-1},s,s_{t+1}\dots,s_T|\pi_2,\mathcal{M})\,\mathrm{d}s_0\cdots\mathrm{d}s_{t-1}\mathrm{d}s_{t+1}\cdots\mathrm{d}s_T\bigg|\,\mathrm{d}s \\
+ R_{\max}\cdot 2\sum_{t=T+1}^{\infty}\gamma^t, \quad \forall T\ge 1
\end{aligned}$$

$$
\begin{aligned}
&\leq R_{\max}\sum_{t=0}^{T}\gamma^t\int_{\mathcal{S}^{T+1}}\left|P(s_0,\dots,s_T|\pi_1,\mathcal{M})-P(s_0\dots,s_T|\pi_2,\mathcal{M})\right|\mathrm{d}s_0\cdots\mathrm{d}s_T\\
&\quad+R_{\max}\cdot 2\sum_{t=T+1}^{\infty}\gamma^t,\quad\forall T\geq 1\\
&=\frac{R_{\max}}{1-\gamma}\cdot\int_{\mathcal{S}^{T+1}}\left|P(s_0,\dots,s_T|\pi_1,\mathcal{M})-P(s_0\dots,s_T|\pi_2,\mathcal{M})\right|\mathrm{d}s_0\cdots\mathrm{d}s_T\\
&\quad+R_{\max}\cdot 2\sum_{t=T+1}^{\infty}\gamma^t,\quad\forall T\geq 1.
\end{aligned}
$$

Let $T\to\infty$, then we obtain the bound proposed by (3). □

We are further interested in bounding the performance difference between two policies by their state sequences in the frequency domain. Benefiting from the properties of the discrete-time Fourier transform (DTFT), we can constrain the performance difference using the Fourier transform over the interval $[0,2\pi]$, instead of using the distribution functions of the state sequences in unbounded space.

Theorem 2. *Suppose that $\mathcal{S}\subset\mathbb{R}^D$ the reward function $R(s,a,s')=R(s)$ is an nth-degree polynomial function with respect to $s\in\mathcal{S}$, then for any two policies π_1 and π_2, their performance difference can be bounded as follows:*

$$
|J(\pi_1)-J(\pi_2)|\leq\frac{\sqrt{D}}{1-\gamma}\cdot\sum_{k=1}^{n}\frac{\left\|R^{(k)}(0)\right\|_D}{k!}\cdot\max_{1\leq i\leq D}\sup_{\omega_i\in[0,2\pi]}\left|F_{\pi_1}^{(k)}(\omega_i)-F_{\pi_2}^{(k)}(\omega_i)\right|,\tag{4}
$$

where $F_\pi^{(k)}(\omega)$ denotes the DTFT of the time series $\mathbf{S}^{(k)}=\{\mathbf{s_0}^k,\mathbf{s_1}^k,\mathbf{s_2}^k,\dots\}$ 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.

Proof. For sake of simplicity, we define $p_t(s|\pi_i)=P(\mathbf{s_t}=s|\pi_i,\mathcal{M})$ for $i=1,2$. We denote ε_t as

$$
\varepsilon_t=\int_{\mathcal{S}}R(s)\left[p_t(s|\pi_1)-p_t(s|\pi_2))\right]\mathrm{d}s.\tag{5}
$$

Based on the Taylor series expansion, we can rewrite the reward function as $R(s)=\sum_{k=0}^{n}\frac{R^{(k)}(0)^\intercal}{k!}s^k$, then for any integer $k\in[1,n]$, we have

$$
\begin{aligned}
|\varepsilon_t|&\leq\sum_{k=0}^{n}\frac{\left\|R^{(k)}(0)\right\|_D}{k!}\cdot\left\|\int_{\mathcal{S}}\left[s^kp_t(s|\pi_1)-s^kp_t(s|\pi_2)\right]\mathrm{d}s\right\|_D\\
&=\sum_{k=0}^{n}\frac{\left\|R^{(k)}(0)\right\|_D}{k!}\left\|\underset{s\sim p_t(\cdot|\pi_1)}{E}\left[s^k\right]-\underset{s\sim p_t(\cdot|\pi_2)}{E}\left[s^k\right]\right\|_D.
\end{aligned}\tag{6}
$$

Since the inverse DTFT of $F_\pi^{(k)}(\omega)$ is the original time series $\mathbf{S}^{(k)}$, we have

$$
\underset{s_i\sim p_t(\cdot|\pi)}{E}\left[s_i^k\right]=\frac{1}{2\pi}\int_0^{2\pi}F_\pi^{(k)}(\omega_i)e^{j\omega_it}\,\mathrm{d}\omega_i,\quad\forall i=1,2,\dots,D.\tag{7}
$$

Then we have

$$
\begin{aligned}
\left|\underset{s_i\sim p_t(\cdot|\pi_1)}{E}\left[s_i^k\right]-\underset{s_i\sim p_t(\cdot|\pi_2)}{E}\left[s_i^k\right]\right|&\leq\frac{1}{2\pi}\int_0^{2\pi}\left|F_{\pi_1}^{(k)}(\omega_i)-F_{\pi_2}^{(k)}(\omega_i)\right|\cdot\left|e^{j\omega_it}\right|\mathrm{d}\omega_i\\
&\leq\sup_{\omega_i\in[0,2\pi]}\left|F_{\pi_1}^{(k)}(\omega_i)-F_{\pi_2}^{(k)}(\omega_i)\right|.
\end{aligned}\tag{8}
$$

Substituting (8) into (6), then we obtain

$$
|\varepsilon_t|\leq\sqrt{D}\cdot\sum_{k=1}^{n}\frac{\left\|R^{(k)}(0)\right\|_D}{k!}\cdot\max_{1\leq i\leq D}\sup_{\omega_i\in[0,2\pi]}\left|F_{\pi_1}^{(k)}(\omega_i)-F_{\pi_2}^{(k)}(\omega_i)\right|.
$$

For the sake of DTFT, the upper bound of ϵ_t is independent of t, then we could derive the performance difference bound as follows.

$$\begin{aligned}|J(\pi_1) - J(\pi_2)| &\leq \sum_{t=0}^{\infty} \gamma^t \cdot |\varepsilon_t| \\ &\leq \frac{1}{1-\gamma} \cdot \sqrt{D} \cdot \sum_{k=1}^{n} \frac{\left\|R^{(k)}(0)\right\|_D}{k!} \cdot \max_{1\leq i\leq D} \sup_{\omega_i\in[0,2\pi]} \left|F_{\pi_1}^{(k)}(\omega_i) - F_{\pi_2}^{(k)}(\omega_i)\right|,\end{aligned}$$

and so we immediately achieve the desired bound in (4). □

A.2 Proof of the Asymptotic Periodicity of States in MDP

This section focuses on analyzing the asymptotic behavior of the state sequences generated from an MDP. We begin by discussing the limiting process of MDP with a finite state space $\mathcal{S}$. Let P be the transition probability matrix and let μ_i be the probability distribution of the states at time t_i. Then we have $\mu_{i+1} = P\mu_i$ for any $i \geq 0$. If the sequence $\{\mu_i\}_{i=0}^{\infty}$ splits into d subsequences with d cyclic limits $\{\mu_\infty^r\}_{r=0}^{d-1}$ that follow the cycle:

$$\mu_\infty^0 \to \mu_\infty^1 \to \cdots \to \mu_\infty^{d-1} \to \mu_\infty^0,$$

then we say that the states of the MDP exhibit *asymptotic periodicity*. Such cyclic asymptotic behavior implies that the limiting distribution of the states eventually repeats in a specific period after a certain number of steps.

We begin by providing some essential definitions in the field of stochastic processes [28], which will be utilized in the following proof. Let P be a transition probability matrix corresponding to n states ($n \geq 1$). Two states i and j are said to *intercommunicate* if there exist paths from i to j as well as from j to i. The matrix P is called *irreducible* if any two states intercommunicate. A set of states is called *irreducible* if any two states in the set intercommunicate. Moreover, a state i is called *recurrent* if the probability of eventual return to i, having started from i, is 1. If this probability is strictly less than 1, the state i is called *transient*.

Note that if the whole state space $\mathcal{S}$ is irreducible, then its transition matrix P is also irreducible. The following lemma demonstrates that if the state space is irreducible, then its asymptotical periodicity is determined by the eigenvalues with modulus 1 of its transition matrix.

Lemma 1. *Suppose that the state space $\mathcal{S}$ is finite with a transition probability matrix $P \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$. If P is an irreducible matrix with d eigenvalues of modulus 1, then for any initial distribution μ_0, $P^n\mu_0$ is asymptotically periodic with a period of d when $d > 1$ and asymptotically aperiodic when $d = 1$.*

Proof. According to the Perron-Frobenius theorem for irreducible non-negative matrices, all eigenvalues of P of modulus 1 are exactly the d complex roots of the equation $\lambda^d - 1 = 0$. They can be formulated as $\lambda_0 = 1, \lambda_1 = \xi^1, \ldots, \lambda_{d-1} = \xi^{d-1}$, where $\xi = e^{\frac{2\pi j}{d}}$. Each of them is a simple root of the characteristic polynomial of the matrix P. Since P is a transition probability matrix, the remaining eigenvalues $\lambda_d, \ldots, \lambda_s$ satisfy $|\lambda_r| < 1$. Therefore, the *Jordan* matrix of P has the form

$$J = \begin{bmatrix} \lambda_0 &&&&&& \\ & \lambda_1 &&&&& \\ && \ddots &&&& \\ &&& \lambda_{d-1} &&& \\ &&&& J_d && \\ &&&&& \ddots & \\ &&&&&& J_s \end{bmatrix}, \text{where } J_k = \begin{bmatrix} \lambda_k & 1 &&& \\ & \lambda_k & 1 && \\ && \ddots & \ddots & \\ &&& \lambda_k & 1 \\ &&&& \lambda_k \end{bmatrix}.$$

We refer to J_k as *Jordan cells*.

Let $|S| = D$, we can rewrite P in its Jordan canonical form $P = XJX^{-1}$ where

$$X = [\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{D-1}].$$

Note that for $k < d$, x_k is the eigenvector corresponding to λ_k. Since the column vectors of X are linearly dependent, there exist $\vec{c} = [c_0, c_1, \ldots, c_{D-1}]$ not all zero, such that $\mu_0 = \sum_{k=0}^{D-1} c_k \vec{x}_k = X\vec{c}$. Thus, we have

$$P^n \mu_0 = \sum_{k=0}^{d-1} c_k \lambda_k^n \vec{x}_k + \sum_{k=d}^{D-1} c_k P^n \vec{x}_k. \tag{9}$$

For any Jordan cell J_k, let α_k be the multiplicity of λ_k, then

$$J_k^n = \begin{bmatrix} \lambda_k & 1 & & & \\ & \lambda_k & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_k & 1 \\ & & & & \lambda_k \end{bmatrix}_{\alpha_k \times \alpha_k}^n = \begin{bmatrix} \lambda_k^n & C_n^{n-1}\lambda_k^{n-1} & \cdots & C_n^{n-\alpha_k+1}\lambda_k^{n-\alpha_k+1} \\ & \lambda_k^n & \cdots & C_n^{n-\alpha_k+2}\lambda_k^{n-\alpha_k+2} \\ & & \ddots & \vdots \\ & & & \lambda_k^n \end{bmatrix}.$$

Since α_k is fixed for matrix P, we have $\lim_{n\to\infty} J_k^n = \mathbf{0}$ for each $k = d, \ldots, D-1$. Then the limiting vector of (9), denoted by $P^\infty \mu_0$, satisfies:

$$P^\infty \mu_0 = \lim_{n\to\infty} X J^n X^{-1} X \vec{c} = \lim_{n\to\infty} \sum_{k=0}^{d-1} c_k \lambda_k^n \vec{x}_k = \lim_{n\to\infty} \mu^{(n)},$$

where we denote $\mu^{(n)} = \sum_{k=0}^{d-1} c_k (e^{j\frac{2\pi k}{d}})^n \vec{x}_k$. Let $r = n \pmod d$, then we have

$$\mu^{(n)} = \mu^{(r)} = \sum_{k=0}^{d-1} c_k (\xi^k)^r \vec{x}_k, \quad \forall n \geq 1.$$

Therefore, the probability sequence $\{P^n \mu_0\}_{n\geq 1}$ will split into d converging subsequences and has d cyclic limiting probability distributions when $n \to \infty$, denoted as

$$\mu_\infty^r = \sum_{k=0}^{d-1} c_k (\xi^k)^r \vec{x}_k, \quad r = 0, 1, \ldots, d-1.$$

Thus, $P^n \mu_0$ is asymptotically periodic with period d if $d > 1$ and asymptotically aperiodic if $d = 1$. $\square$

We now consider a more general state space that may not necessarily be irreducible. According to the Decomposition theorem of the Markov chain [28], the finite state space S can be partitioned uniquely as a set of transient states and one or several irreducible closed sets of recurrent states. According to [29], after performing an appropriate permutation of rows and columns, we can rewrite the transition probability matrix P in its canonical form:

$$P = \left[\begin{array}{cccc|c} R_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & R_\alpha & \mathbf{0} \\ \hline T_1 & T_2 & \cdots & T_\alpha & Q \end{array}\right],$$

where $R_1, \ldots, R_\alpha$ represent the probability submatrices corresponding to the recurrent classes, Q represents the probability submatrix corresponding to the transient states, and $T_1, \ldots, T_\alpha$ represent the probability submatrices corresponding to the transitions between transient and recurrent classes $R_1, \ldots, R_\alpha$ respectively.

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 $d = \mathrm{lcm}(d_1, d_2, \ldots, d_\alpha)$ when $d > 1$ and asymptotically aperiodic when $d = 1$.*

Proof. Since P is a block upper-triangular, it can be shown that the eigenvalues of P are equal to the union of the eigenvalues of the diagonal blocks $R_1, \dots, R_\alpha, Q$. Note that the nth-power of P satisfies the following expression:

$$P^n = \left[\begin{array}{cccc|c} R_1^n & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & R_2^n & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & R_\alpha^n & \mathbf{0} \\ \hline T_1^{(n)} & T_2^{(n)} & \cdots & T_\alpha^{(n)} & Q^n \end{array}\right],$$

where $T_r^{(n)}$ is related to the $(n-1)$-th or the lower power of R_r and Q. From Theorem 4.3 of [29], we obtain that $\lim_{n\to\infty} Q^n = \mathbf{0}$, which implies that all eigenvalues of Q have modulus less than 1.

On the other hand, note that the sum of every row in matrix R_r is equal to 1, which means $\lambda = 1$ is an eigenvalue of R_r and all eigenvalues of R_r satisfy $|\lambda| \leq 1$. Thus, the spectral radius of P is equal to 1.

Note that the proof of Lemma 1 implies that the asymptotic periodicity of $P^n\mu_0$ depends on the eigenvalues of P that have modulus 1. Since R_r is non-negative irreducible with spectral radius 1, based on the Perron-Frobenius theorem used in Lemma 1, we can express the eigenvalues of R_r in modulus 1 as:

$$\lambda_{r,k} = e^{j\frac{2\pi k}{d_r}}, \quad , k = 0, 1, \dots, d_r - 1.$$

Based on the above discussion, it is easy to check that $\bigcup_{r=1}^{\alpha} \{\lambda_{r,0}, \dots, \lambda_{r,d_r-1}\}$ is the set of all eigenvalues of modulus 1 of P. Rewrite P in its Jordan canonical form $P = XJX^{-1}$, where

$$J = \begin{bmatrix} \lambda_{1,0} & & & & & & & & \\ & \ddots & & & & & & & \\ & & \lambda_{1,d_1-1} & & & & & & \\ & & & \lambda_{2,0} & & & & & \\ & & & & \ddots & & & & \\ & & & & & \lambda_{\alpha,d_\alpha-1} & & & \\ & & & & & & J_{d_1+\cdots+d_\alpha} & & \\ & & & & & & & \ddots & \\ & & & & & & & & J_s \end{bmatrix}$$

and $X = [\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{D-1}]$ is an invertible matrix. Similar to the proof in Lemma 1, we get

$$P^\infty\mu_0 = \lim_{n\to\infty} \sum_{r=1}^{\alpha} \sum_{k=0}^{d_r-1} c_k (e^{j\frac{2\pi k}{d_r}})^n \vec{x}_k := \lim_{n\to\infty} \mu^{(n)}.$$

Let $d = \operatorname{lcm}(d_1, d_2, \dots, d_\alpha)$ and $r = n \pmod d$, then we have

$$\mu^{(n)} = \mu^{(r)}, \quad \forall n \geq 1.$$

Therefore, the probability sequence $\{P^n\mu_0\}_{n\geq 1}$ will split into d converging subsequences and has d cyclic limiting probability distributions when $n \to \infty$, denoted as

$$\mu_\infty^r = \sum_{r=1}^{\alpha} \sum_{k=0}^{d_r-1} c_k e^{j\frac{2\pi k r}{d_r}} \vec{x}_k, \quad r = 0, 1, \dots, d-1.$$

Thus, $P^n\mu_0$ is asymptotically periodic with period d if $d > 1$ and asymptotically aperiodic if $d = 1$. This completes the proof. $\square$

A.3 Proof of the Convergence of Our Auxiliary Loss

In this section, we provide a detailed derivation of the learning objective of SPF. As the DTFT of discrete-time state sequences is a continuous function that is difficult to compute, we practically sample the DTFT at L equally-spaced points.

$$[\mathcal{F}\widetilde{s}_t]_k = \sum_{n=0}^{+\infty} [\widetilde{s}_t]_n \, e^{-j\frac{2\pi k}{L}n}, \quad k = 0, 1, \ldots, L-1. \tag{10}$$

As a result, the prediction target takes the form of a matrix with dimensions of $L * D$, where D denotes the dimension of the state space. The auxiliary task is designed to encourage the representation to predict the Fourier transform of the state sequences using the current state-action pair as input. Specifically, we define the prediction target $F_{\pi,p}(s_t, a_t)$ as follows:

$$F_{\pi,p}(s_t, a_t) = \mathcal{F}\widetilde{s}(s_t, a_t) = \left\{ \sum_{n=0}^{+\infty} [\widetilde{s}(s_t, a_t)]_n \, e^{-j\frac{2\pi k}{L}n} \right\}_{k=0}^{L-1}, \tag{11}$$

For simplicity of notation, we substitute $F(s_t, a_t)$ for $F_{\pi,p}(s_t, a_t)$ in the following. We can derive that the DTFT functions at successive time steps are related to each other in a recursive form:

$$\begin{aligned}
[F(s_t, a_t)]_k &= \sum_{n=0}^{+\infty} \gamma^n \cdot e^{-j\frac{2\pi k}{L}n} \cdot E_{\pi,p}\left[s_{t+n+1} \middle| s_t = s, a_t = a \right] \\
&= E_p\left[s_{t+1} \middle| s_t = s, a_t = a \right] + \gamma \cdot e^{-j\frac{2\pi k}{L}} \cdot \\
&\qquad E_{s_{t+1}\sim p, a_{t+1}\sim\pi}\left[\sum_{n=0}^{+\infty} \gamma^n \cdot e^{-j\frac{2\pi k}{L}n} \cdot E_p\left[s_{t+n+2} \middle| s_{t+1}, a_{t+1} \right] \right] \\
&= [\widetilde{s}_t]_0 + \gamma \cdot e^{-j\frac{2\pi k}{L}} \cdot E_{\pi,p}\left[[F(s_{t+1}, a_{t+1})]_k \right], \; \forall\, k = 0, 1, \ldots L-1.
\end{aligned}$$

We can further express the above equation as a matrix-form recursive formula as follows:

$$F(s_t, a_t) = \widetilde{\boldsymbol{S}}_t + \Gamma \, E_{\pi,p}\left[F(s_{t+1}, a_{t+1}) \right], \tag{12}$$

where

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

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

Similar to the TD-learning of value functions, we can prove that the above recursive relationship (12) can be reformulated as a contraction mapping $\mathcal{T}$. Due to the properties of contraction mappings, we can iteratively apply the operator $\mathcal{T}$ to compute the target DTFT function until convergence in tabular settings.

Theorem 4. *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

$$\mathcal{T}F(s_t, a_t) = \widetilde{\boldsymbol{S}}_t + \Gamma \, E_{\pi,P}\left[F(s_{t+1}, a_{t+1}) \right] \tag{13}$$

is a contraction mapping, where $\widetilde{\boldsymbol{S}}_t$ and Γ are defined as above.

Proof. For any $F_1, F_2 \in \mathcal{F}$, we have

$$\begin{aligned}
\|\mathcal{T}F_1 - \mathcal{T}F_2\|_{\mathcal{F}} &= \sup_{\substack{s\in\mathcal{S}\\ a\in\mathcal{A}}} \max_{0\le k<L} \Bigg\| s + \gamma e^{-j\frac{2\pi k}{K}} E_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\Big[\big[F_1(s',a')\big]_k \Big| s,a\Big] \\
&\qquad\qquad - s - \gamma e^{-j\frac{2\pi k}{K}} E_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\Big[\big[F_2(s',a')\big]_k \Big| s,a\Big] \Bigg\|_D \\
&\le \gamma \cdot \max_{0\le k<L} \sup_{\substack{s\in\mathcal{S}\\ a\in\mathcal{A}}} \Bigg\| E_{\substack{s'\sim P(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}\Big[\big[F_1(s',a')\big]_k - \big[F_2(s',a')\big]_k \Big| s,a\Big] \Bigg\|_D \\
&\le \gamma \cdot \max_{0\le k<L} \sup_{\substack{s'\in\mathcal{S}\\ a'\in\mathcal{A}}} \big\| \big[F_1(s',a') - F_2(s',a')\big]_k \big\|_D \\
&= \gamma \cdot \|F_1 - F_2\|_{\mathcal{F}}.
\end{aligned}$$

Note that $\gamma \in [0,1)$, which implies that $\mathcal{T}$ is a contraction mapping. □

B Pseudo-code of SPF

The training procedure of SPF is shown in the pseudo-code as follows:

Algorithm 1 State Sequences Prediction via Fourier Transform (SPF)

```
Denote parameters of the online encoder (φ_s, φ_{s,a}), predictor F, and projection ψ as θ_aux
Denote parameters of the target encoder (φ̂_s, φ̂_{s,a}), predictor F̂, and projection ψ̂ as θ̂_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 D and parameters θ_aux, θ_RL
for each environment step t do
    a_t ~ π(·|φ_s(s_t))
    s_{t+1}, r_{t+1} ~ p(·|s_t, a_t)
    D ← D ∪ (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 D
    θ_aux ← θ_aux − α_aux ∇_{θ_aux} L_pred(θ_aux, θ̂_aux)
    resampling a minibatch of {(s_t, a_t, s_{t+1}, r_{t+1})} from D
    s̄_t ← φ_s(s_t)
    z_{s_t,a_t} ← φ_{s,a}(φ_s(s_t), a_t)
    update the RL agent parameters θ_RL with the representations s̄_t, z_{s_t,a_t}
    update parameters of target networks with θ̂_aux ← τθ_aux + (1 − τ)θ̂_aux every K steps
end for
```

C Network Details

The encoders ϕ_s and $\phi_{s,a}$ share the same architecture. Each layer of the encoders uses MLP-DenseNet [16], a slightly modified version of DenseNet. For each MuJoCo task, the incremental number of hidden units per layer is selected from $\{30, 40\}$, while the number of layers is selected from $\{6, 8\}$ (see Table 1). Both the predictor $\mathcal{F}$ and the projection ψ apply a 2-layer MLP. We divide the last layer of the predictor into two heads as the real part $\mathcal{F}_{\text{Re}}$ and the imaginary part $\mathcal{F}_{\text{Im}}$, respectively, since the prediction target of our auxiliary task is complex-valued. With respect to the projection module, we add an additional 2-layer MLP (referred to as *Projection2*) after the original online projection to perform a dimension-invariant nonlinear transformation on the predicted DTFT that has been projected to a lower-dimensional space. We do not apply this nonlinear operation to the target projection. This additional step is carried out to prevent the projection from collapsing to a constant value in the case where the online and target projections share the same architecture.

In Fourier analysis, the low-frequency components of the DTFT contain information about the long-term trends of the signal, with higher signal energy, while the high-frequency components of the

Table 1: Detailed setting of the encoder for six MuJoCo tasks.

Environment	Number of Layers	Number of Units per Layer	Activation Function
HalfCheetah-v2	8	30	Swish
Walker2d-v2	6	40	Swish
Hopper-v2	6	40	Swish
Ant-v2	6	40	Swish
Swimmer-v2	6	40	Swish
Humanoid-v2	8	40	Swish

DTFT reflect the amount of short-term variation present in the state sequences. Therefore, we attempt to preserve the overall information of the low and high-frequency components of the predicted DTFT by directly computing the cosine similarity distance without undergoing the dimensionality reduction process. For the remaining frequency components of the predicted DTFT, we first utilize projection layers to perform dimensionality reduction, followed by calculating the cosine similarity distance. The sum of these three distances is used as the final loss function, which we call *freqloss*.

D Hyperparameters

Table 2: Hyperparameters of auxiliary prediction tasks.

Hyperparameter	Setting
Optimizer	Adam
Discount γ	0.99
Learning rate	0.0003
Number of batch size	256
Predictor: Number of hidden layers	1
Predictor: Number of hidden units per layer	1024
Predictor: Activation function	ReLU
Projection: Number of hidden layers	1
Projection: Number of hidden units per layer	512
Projection: Activation function	ReLU
Projection2: Number of hidden layers	1
Projection2: Number of hidden units per layer	512
Projection2: Activation function	ReLU
Number of discrete points for sampling the DTFT L	128
The dimensionality of the output of projection	512
Replay buffer size	100,000
Pre-training steps	10000
Target smoothing coefficient τ	0.01
Target update interval K	1000
Hyperparameters of SPF-SAC	
Each module: Normalization Layer	BatchNormalization
Random collection steps before pre-training	10,000
Hyperparameters of SPF-PPO	
Each module: Normalization Layer	LayerNormalization
Random collection steps before pre-training	4,000
θ_{aux} update interval K_2	
HalfCheetah-v2	5
Walker2d-v2	2
Hopper-v2	150
Ant-v2	150
Swimmer-v2	200
Humanoid-v2	1

We select $L = 128$ as the number of discrete points sampled over one period of DTFT. In practice, due to the symmetry conjugate of DTFT, the predictor $\mathcal{F}$ only predicts $\frac{L}{2} + 1$ points on the left half of our frequency map, as mentioned in Section 5.2. The projection module described in Section 5.3 projects the predicted value, a matrix with the dimension of $L * D$, into a 512-dimensional vector. To update target networks, we overwrite the target network parameters with an exponential moving average of the online network parameters, with a smoothing coefficient of $\tau = 0.01$ for every $K = 1000$ steps.

In order to eliminate dependency on the initial parameters of the policy, we use a random policy to collect transitions into the replay buffer [30] for the first 10K time steps for SAC, and 4K time steps for PPO. We also pretrain the representations with the aforementioned random collected samples to stabilize inputs to each RL algorithm, as described in [16].

The network architectures, optimizers, and hyperparameters of SAC and PPO are the same as those used in their original papers, except that we use mini-batches of size 256 instead of 100. As for PPO, we perform K_2 gradient updates of θ_{aux} for every K_2 steps of data sampling. The update interval K_2 is set differently for six MuJoCo tasks and can be found in Table 2.

E Visualization

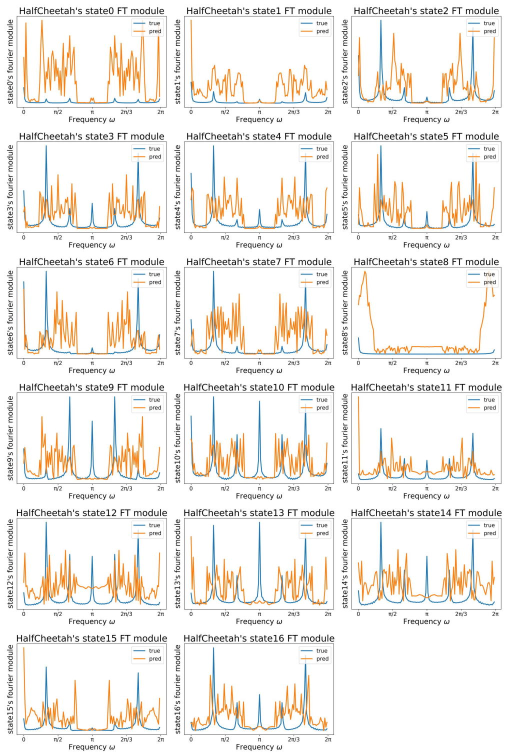
To demonstrate that the representations learned by SPF effectively capture the structural information contained in infinite-step state sequences, we compare the true state sequences with the states recovered from the predicted DTFT via the inverse DTFT.

Specifically, we first generate a state sequence from the trained policy and select a goal state s_t at a certain time step. Next, we choose a historical state s_{t-k} located k steps past the goal state and select an action a_{t-k} based on the trained policy $\pi(\cdot|s_{t-k})$ as the inputs of our trained predictor. We then obtain the DTFT $F_{t-k} := F_\pi(s_{t-k}, a_{t-k})$ of state sequences starting from the state s_{t-k+1}. Next, we compute the kth element of the inverse DTFT of F_{t-k} and obtain a recovered state $\widehat{s}_t$, which represents that we predict the future goal state using the historical state located k steps past the goal state. By selecting a sequence of states over a specific time interval as the goal states and repeating the aforementioned procedures, we will obtain a state sequence recovered by k-step prediction. In Figure 5(b), 6(b), 7(b), 8(b), 9(b) and 10(b), we visualize the true state sequence (the blue line) and the recovered state sequences (the red lines) via k-step predictions for $k = 1, 2, 3, 4, 5$. Note that the lighter red line corresponds to predictions made by historical states from a more distant time step. We conduct the visualization experiment on six MuJoCo tasks using the representations and predictors trained by SPF-SAC or SPF-PPO. Due to the large dimensionality of the states in Ant-v2 and Humanoid-v2, which contain many zero values, we have chosen to visualize only six dimensions of their states, respectively. The fine distinctions between the true state sequences and the recovered state sequences from our trained representations and predicted FT indicates that our representation effectively captures the inherent structures of future state sequences.

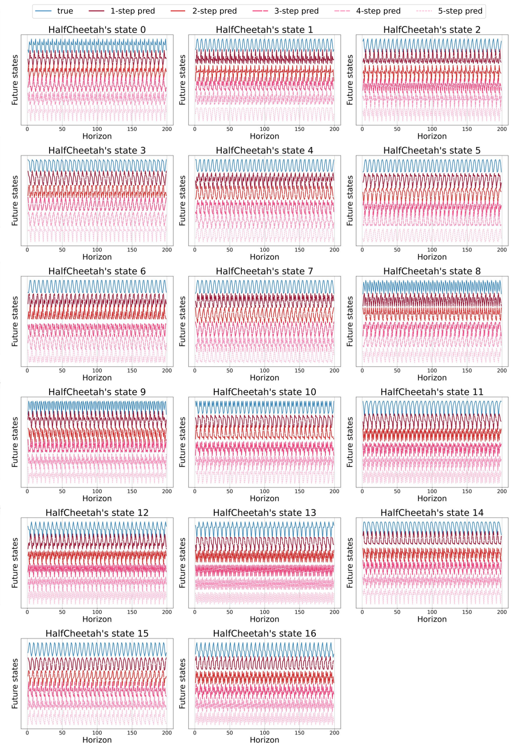
Furthermore, we provide a visualization that compares the true DTFT and the predicted DTFT in Figure 5(a), 6(a), 7(a), 8(a), 9(a) and 10(a). To accomplish this, we use our trained policies to interact with the environments and select the state sequences of the 200 last steps of an episode. The blue lines represent the true DTFT of these state sequences, while the orange line represents the predicted DTFT using the online encoder and predictor trained by our learned policies. It is evident that the true DTFT and the predicted DTFT exhibit significant differences. These results demonstrate the ability of SPF to effectively extract the underlying structural information in infinite-step state sequences without relying on high prediction accuracy.

F Code

Codes for the proposed method are available at https://anonymous.4open.science/r/spf_nips_2023-10D1/README.md.

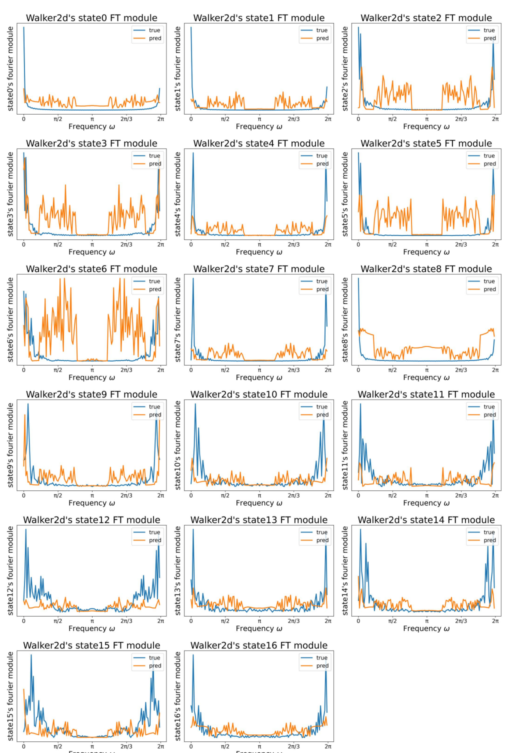


(a) True and predicted DTFT

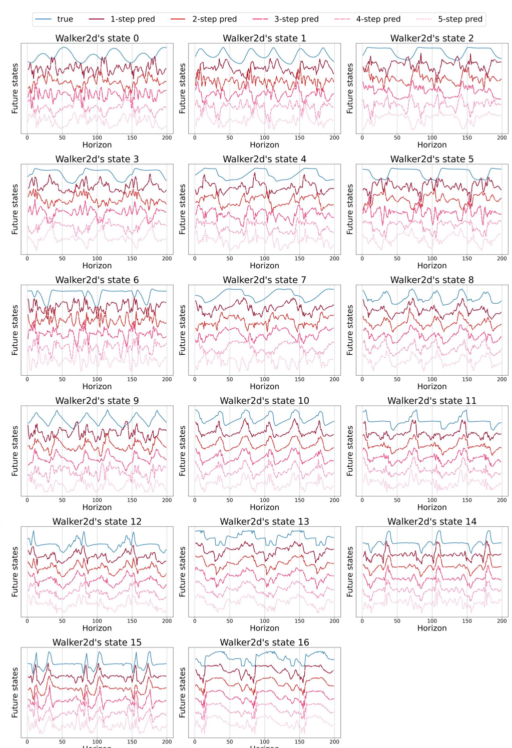


(b) True and recovered state sequences

Figure 5: Predicted values via representations trained by SPF-SAC on HalfCheetah-v2

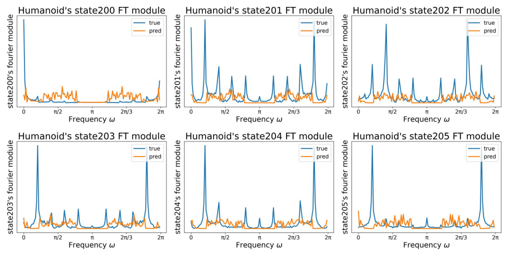


(a) True and predicted DTFT

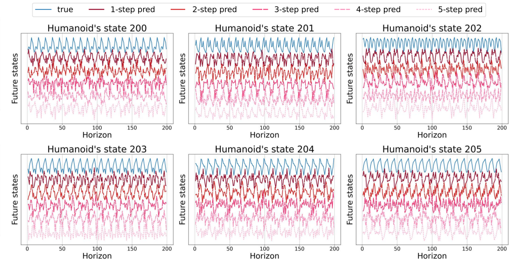


(b) True and recovered state sequences

Figure 6: Predicted values via representations trained by SPF-SAC on Walker2d-v2

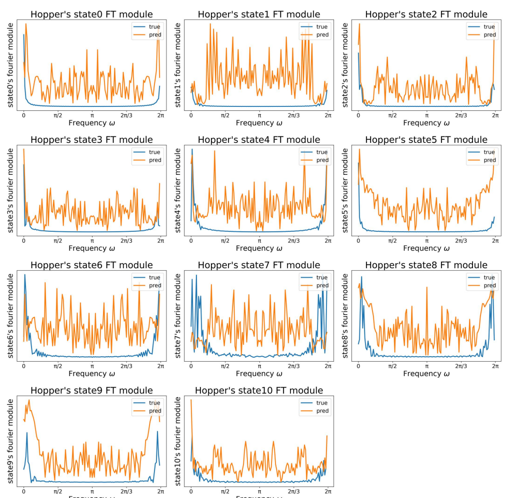


(a) True and predicted DTFT

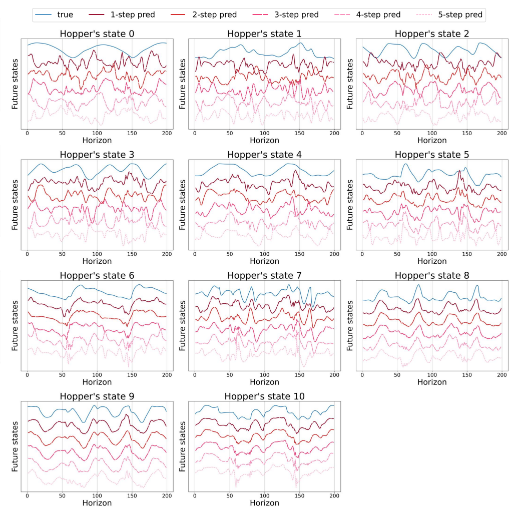


(b) True and recovered state sequences

Figure 7: Predicted values via representations trained by SPF-SAC on Humanoid-v2

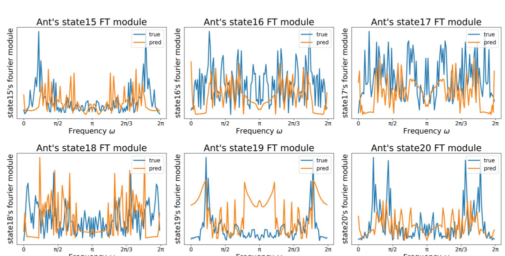


(a) True and predicted DTFT

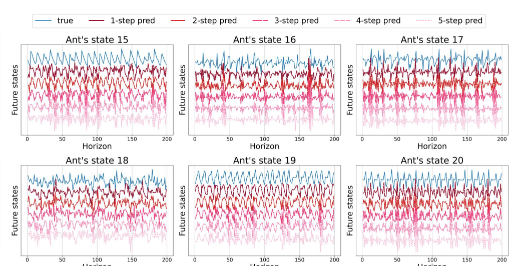


(b) True and recovered state sequences

Figure 8: Predicted values via representations trained by SPF-PPO on Hopper-v2

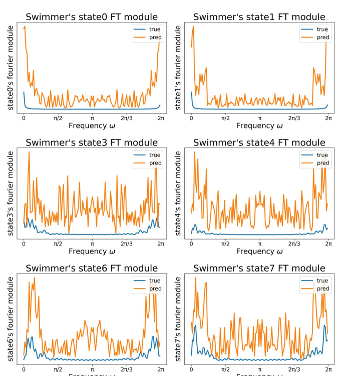


(a) True and predicted DTFT

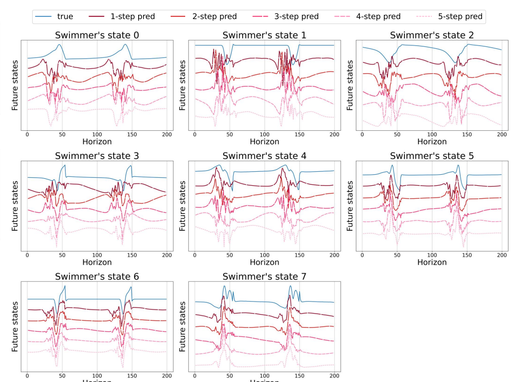


(b) True and recovered state sequences

Figure 9: Predicted values via representations trained by SPF-PPO on Ant-v2



(a) True and predicted DTFT



(b) True and recovered state sequences

Figure 10: Predicted values via representations trained by SPF-PPO on Swimmer-v2