A More Details on Predecessor Representation

Here we provide proofs of the reciprocal relationship between the SR and the PR.

Proposition A.1. $\mathbf{N}diag(\mathbf{z}) = diag(\mathbf{z})\mathbf{M}$, *where* $diag(\mathbf{z})$ *is the diagonal matrix with the diagonal elements as the vector* $\mathbf{z}$, *and* $\mathbf{z}$ *is the vector of stationary distribution of* $\mathcal{P}^\pi$ *(i.e.,* $\mathbf{z}[i] = \lim_{t\to\infty} \mathbb{E}_{\mathcal{P}^\pi}[s_t = i]$.

Proof. Given the formal definition of the SR and the PR (Eq. 3; 11), we have the following analytical expressions.

$$\mathbf{M} = (\mathbf{I} - \gamma\mathcal{P}^\pi)^{-1}; \quad \mathbf{N} = (\mathbf{I} - \gamma\tilde{\mathcal{P}}^\pi)^{-1}; \tag{19}$$

where $\tilde{\mathcal{P}}^\pi$ is the temporally reversed transition distribution. Assume matrix formulation of $\mathcal{P}^\pi$ and $\tilde{\mathcal{P}}^\pi$, $\mathbf{P}$ and $\tilde{\mathbf{P}}$ in $\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, we have the following.

$$\begin{aligned} \tilde{\mathbf{P}}_{ij} = \mathbb{P}(s_t = i | s_{t+1} = j) = \frac{\mathbb{P}(s_{t+1} = j | s_t = i)\mathbb{P}(s_t = i)}{\mathbb{P}(s_{t+1} = j} = \frac{\mathbf{P}_{ij}\mathbf{z}_i}{\mathbf{z}_j}, \\ \Rightarrow \tilde{\mathbf{P}}\text{diag}(\mathbf{z}) = \text{diag}(\mathbf{z})\mathbf{P}, \end{aligned} \tag{20}$$

Substituting the reciprocal relationship between $\tilde{\mathbf{P}}$ and $\mathbf{P}$ into the definition of the PR, we have the following.

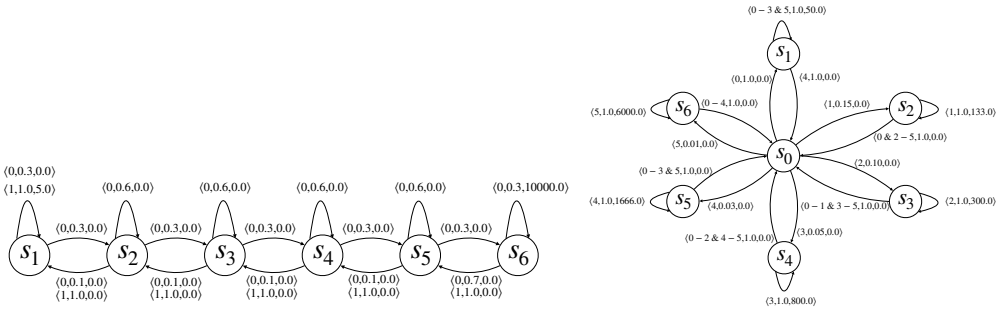
$$\begin{aligned} \mathbf{N} &= \left(\mathbf{I} - \gamma\text{diag}(\mathbf{z})\mathbf{P}\text{diag}(\mathbf{z})^{-1}\right)^{-1}, \\ \mathbf{N}\text{diag}(\mathbf{z}) &= \left(\mathbf{I} - \gamma\text{diag}(\mathbf{z})\mathbf{P}\text{diag}(\mathbf{z})^{-1}\right)^{-1}\text{diag}(\mathbf{z}) \\ &= \left(\text{diag}(\mathbf{z})^{-1}(\mathbf{I} - \gamma\text{diag}(\mathbf{z})\mathbf{P}\text{diag}(\mathbf{z})^{-1})\right)^{-1} \\ &= \left((\mathbf{I} - \gamma\mathbf{P})\text{diag}(\mathbf{z})^{-1}\right)^{-1} \\ &= \text{diag}(\mathbf{z})\left((\mathbf{I} - \gamma\mathbf{P})\right)^{-1} \\ &= \text{diag}(\mathbf{z})\mathbf{M} \end{aligned} \tag{21}$$

□

B Further results on tabular hard exploration tasks.

B.1 Graphical illustration of tabular hard-exploration tasks.

The demos of RiverSwim and SixArms is shown in Figure 5. In both tasks, the environmental transition dynamics impose asymmetry, biasing the agent towards low-rewarding states that are easier to reach, with greater rewards available in hard-to-reach states.



(a) RiverSwim (b) SixArms

Figure 5: Discrete MDPs. Transition probabilities are denoted by $\langle$action, probability, reward$\rangle$. In RiverSwim (a), the agent starts in state 1 or 2. In SixArms (b), the agent starts in state 0.

Table 3: Evaluations on RiverSwim and SixArms with intrinsic rewards based on fixed SR/FR (averaged over 100 seeds, numbers in the parentheses represents standard errors).

	SARSA-SR	SARSA-FR	SARSA-SRR
RiverSwim	327,402	278,096	**3,096,913**
	(787,118)	(666,752)	(230,059)
SixArms	969,781	1,143,037	**2,059,424**
	(2,895,306)	(1,939,021)	(3,292,936)

B.2 Pseudocode for SARSA-SRR.

We provide the pseudocode for SARSA-SRR in Algorithm 1. We note that SARSA, SARSA-SR and SARSA-FR utilise the similar algorithm, but only replacing the intrinsic bonus.

Algorithm 1 Pseudocode for SARSA-SRR

Require: $\alpha, \eta, \gamma, \gamma_{\text{SR}}, \beta, \epsilon$
  $s = env.reset()$;
  $\mathbf{M} = \mathbf{0} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$; ▷ Initialise the SR matrix as zero matrix
  $\mathbf{Q} = \mathbf{0} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$;
  while not $done$ **do**
    $\theta \sim \mathcal{U}(0, 1)$;
    if $\theta < \epsilon$ **then** ▷ ϵ-greedy policy
      $a \sim \mathcal{U}(\mathcal{A})$;
    else
      $a = \text{argmax}_{a\in\mathcal{A}}\, Q[s, a]$;
    end if
    $s', r, done = env.step(a)$;
    $\mathbf{M}[s, :] = \mathbf{M}[s, :] + \eta\,(\mathbb{1}(s) + \gamma_{\text{SR}}(1 - done)\mathbf{M}[s', :] - \mathbf{M}[s, :])$; ▷ TD-learning of the SR
    $r = r + \beta(\mathbf{M}[s, s'] - ||\mathbf{M}[:, s']||_1)$; ▷ Constructing intrinsic reward
    $\theta' \sim \mathcal{U}(0, 1)$;
    if $\theta' < \epsilon$ **then**
      $a' \sim \mathcal{U}(\mathcal{A})$;
    else
      $a' = \text{argmax}_{a\in\mathcal{A}}\, Q[s', a]$;
    end if
    $\mathbf{Q}[s, a] = \mathbf{Q}[s, a] + \alpha\,(r + \gamma(1 - done)\mathbf{Q}[s', a'] - \mathbf{Q}[s, a])$;
    $s = s'$;
  end while

B.3 Evaluations given the fixed SR.

Conforming to our analysis of $r_{\text{SR-R}}$ with fixed SR (Section 3), we additionally evaluate SARSA-SR/FR/SRR with the corresponding intrinsic rewards constructed based on fixed SR/FR matrix on RiverSwim and SixArms (Table 3. Similar to what we found in the grid worlds (Figure 1c), both SARSA-SR and SARSA-FR perform worse than their online-SR counterparts (note one exception being SARSA-FR on SixArms). However, in contrast to the decrease in exploration efficiency of SARSA-SRR in grid worlds, we found that fixing the SR actually improves the performance of SARSA-SRR. Hence, in accord with our analysis in Section 3, the cause for the improved empirical performance of $r_{\text{SR-R}}$ does not lie solely in the online learning process of SR, but might stems from the inherent "bottleneck-seeking" property of $r_{\text{SR-R}}$.

B.4 Ablation studies of SPIE in discrete tasks

We perform ablation studies on SARSA-SRR for further demonstration of the utility of the SPIE objective of combining both the prospective and retrospective information. We firstly show that prospective

Table 4: Ablation studies of SARSA-SRR on RiverSwim and SixArms.

	SARSA-SRR	SARSA-SRR(a)	SARSA-SRR(b)	SARSA-SRR(c)
RiverSwim	$\mathbf{2,547,156}$	127,703	$\mathbf{2,629,947}$	95,691
	(479,655)	(530,564)	(930,170)	(181,216)
SixArms	$\mathbf{2,199,291}$	893,530	$\mathbf{1,902,553}$	562,346
	(1,024,726)	(2,601,324)	(2,211,960)	(1,748,455)

information alone cannot yield strong exploration, whereas utilising solely the retrospective information maintains the strong explorative performance. We consider two variants of SARSA-SRR, SARSA-SRR(a) and SARSA-SRR(b), with the respective intrinsic rewards as following.

$$\mathcal{R}_{\text{SR-R(a)}}(s,a,s') = \hat{M}[s,s']\,, \quad \mathcal{R}_{\text{SR-R(b)}}(s,a,s') = -||\hat{M}[:,s']||_1\,, \tag{22}$$

From Table 4, we observe that utilising the prospective information alone for exploration yields suboptimal performance, hence empirically justifying the utility of the SPIE framework. However, we do observe that utilising the retrospective information alone yields near- or supra-optimal performance. Together, the results indicate that the global topological information contained in the retrospective information is essential for intrinsic exploration purposes.

We argue that the dynamic balancing between exploring states with high uncertainty and bottleneck states is a key factor driving the empirical success of SPIE. In order to test this hypothesis, we devise a variant of the $\mathcal{R}_{\text{SR-R}}$.

$$\mathcal{R}_{\text{SR-R(c)}} = ||\hat{M}[s,:]||_1 - \hat{M}[s,s']\,, \tag{23}$$

Intuitively, $\mathcal{R}_{\text{SR-R(c)}}$ provides an intrinsic motivation for taking transitions that lead to states that are less reachable from s, which only yields exploration towards states of high uncertainty, but does not provide any motivation towards bottleneck states. Indeed, as we observe from Table 4 that SARSA-SRR(c) also yields suboptimal performance, providing empirical evidence supporting the benefits of SPIE in driving the agents towards bottleneck states.

C Further results on exploration in grid worlds

C.1 Transient dynamics of exploration.

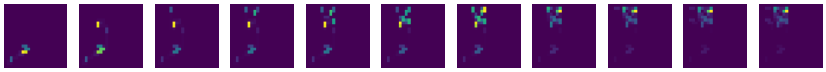
We look more closely at the transient dynamics of the considered agents during pure exploration in *Cluster-simple-large* (where *Cluster-simple-large* denotes the 20×20 grid world with two clusters). We observe that in the absence of external reinforcement, SARSA-SR, regardless of based on intrinsic rewards given either online-learned or fixed SR matrix, exhibits minimal exploration (Figure 6a, 6b). This is largely due to its local exploration behaviour. For SARSA-FR, we observe significant difference between using online-trained and fixed FR matrix, where exploration with intrinsic rewards based on fixed FR completed disrupts exploration, only exploring a small proportion of the environment. In contrast, we observe that SARSA-SRR consistently fully explores both clusters (repeatedly) under both conditions. Additionally, by closely examining the transient dynamics during the exploration phase, we observe the "cycling" behaviour[5].

C.2 Effect of optimistic initialisation.

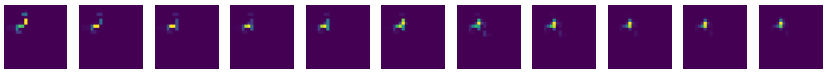
We note that across all considered SARSA agents, the Q values were initialised to be 0 for all state action pairs. Given that all SR entries are non-negative, we know that $r_{\text{SR-R}}$ only admits negative rewards, hence the zero-initialisation yields optimistic initialisation, which encourages the agent to explore [40, 29]. To disentangle the effect of SPIE from optimistic initialisation, we perform the ablation study on pure exploration with augmented SARSA-SR and SARSA-FR agents with optimistic initialisation. Specifically, we note that the maximum value the SR entries can take is $\frac{1}{1-\gamma}$, and additionally since the FR entries, by definition, are always less than or equal to the corresponding

[5] see the attached videos in supplementary materials for the full exploration dynamics for the considered agents

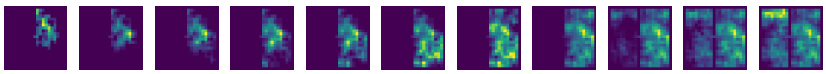
(a)



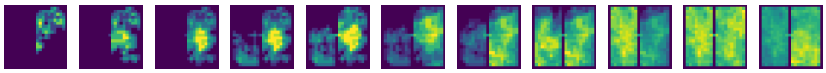
(b)



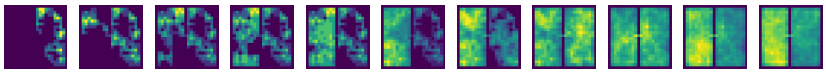
(c)



(d)



(e)



(f)

Figure 6: **Pure exploration given fixed SR / FR measures.** Temporal evoluation of state coverage heatmaps over 6000 training steps of (a) SRASA-SR; (c) SARSA-FR; (e) SARSA-SRA agents with intrinsic rewards based on fixed SR/FR measures in *OF-small*; and (b), (d), (f) for the counterparts with online-trained SR/FR measures in the 20×20 *Cluster-simple* grid world. From left to right: $200, 400, 600, 800, 1000, 1500, 2000, 3000, 4000, 5000, 6000$ steps.

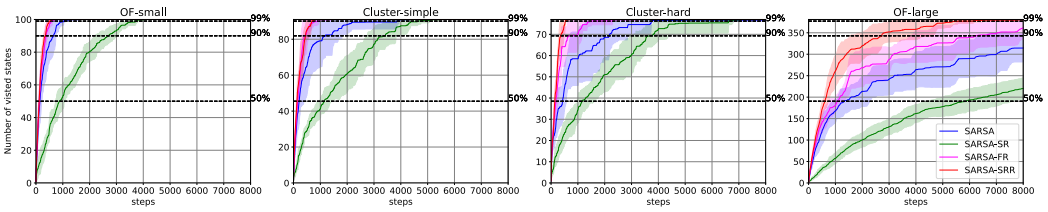


Figure 7: **Ablation study on optimistic initialisation on exploration efficiency.** We evaluate SARSA, SARSA-SRR, and optimistically augmented SARSA-SR and SARSA-FR on the considered grid worlds (Figure 1a).

SR entries, we initialise the Q values for all state-action pairs for both SARSA-SR and SARSA-FR to be $\frac{1}{1-\gamma}$. We evaluate the exploration efficiency for the optimistically augmented agents on the grid worlds (Figure 7), and we observe that despite the optimistic initialisation improves the performance of both SARSA-SR and SARSA-FR relative to their corresponding naive counterparts, the performance differences in terms of exploration efficiency between the augmented agents and SARSA-SRR are significant, hence justifying the utility of the SPIE framework independent of the optimistic initialisation.

D Further results on deep RL implementation of SPIE in Atari games

D.1 Ablation study on the effect of predictive reconstruction auxiliary task

In our implementation of DQN-SF-PF, by following relevant literature [27, 12], we include an additional sub-module in the neural architecture for predicting action-dependent future observation, which is trained via minimising the predictive reconstruction error. The purpose of including this sub-module is purely for learning better latent representations underlying the visual observation. We validate the utility of such predictive reconstruction auxiliary supervision by performing ablation study. We implemented an alternative version of DQN-SF-PF, removing the visual reconstruction sub-module, and test on Montezuma's Revenge. The resulting model achieves 551.5 points (averaged over 5 random seeds, s.e. equals 618.4). We observe that there is a significant decrease from standard DQN-SF-PF (Table 2), indicating the importance of stronger representation learning given the predictive reconstruction auxiliary task. Moreover, given the reported performance of 398.5 points (s.e., equals 230.1) of DQN-SF in the absence of predictive reconstruction auxiliary task from Machado et al. [12], we observe that the SPIE objective still yields improved performance over exploration with SF alone, justifying the utility of SPIE irrespective of the specific neural architecture we choose.

E Experiment Details

Here we provide further details of the experiments presented in the main paper.

Tabular tasks. We run hyperparameter sweeps for all considered agents (SARSA, SARSA-SR, SARSA-FR, SARSA-SRR) on the following hyperparameters: $\{0.005, 0.05, 0.1, 0.25, 0.5\}$ for learning rate of TD learning for the Q values (α); $\{0.005, 0.05, 0.1, 0.25, 0.5\}$ for learning rate of TD learning for the SR/FR matrices (η); $\{0.5, 0.8, 0.9, 0.95, 0.99\}$ for the discounting factor defining the SR/FR formulation ($\gamma_{\text{SR/FR}}$); $\{1, 10, 50, 100, 1000, 10000\}$ for the multiplicative scaling factor controlling the scale of the intrinsic rewards (β); $\{0.01, 0.05, 0.1\}$ for the degree of randomness in ϵ-greedy exploration (ϵ). The complete sets of optimal hyperparameters for the reported performance of the considered agents in Table 1 (and for the corresponding agents with intrinsic rewards based on fixed SR/FR matrix; Table 3) in shown in Table 5.

Exploration in grid worlds. For all presented results in the grid worlds, we use the hyperparameters $(0.1, 0.1, 0.95, 0.95, 1.0, 0.1)$ for $(\alpha, \eta, \gamma, \gamma_{\text{SR/FR}}, \beta, \epsilon)$.

MountainCar experiment. We use the 128-dimensional random Fourier features, defined over the two-dimensional state space (location$\times$speed), as the state representation. We use the hyperparameters

Table 5: Hyperparameters for the considered agents in the tabular hard-exploration tasks (the values in parentheses are the corresponding hyperparameter values for the learning of the PR).

	agent	α	η	γ	$\gamma_{\text{SR/FR}}$	β	ϵ
RiverSwim	SARSA	0.005	-	0.95	-	-	0.01
	SARSA-SR	0.25	0.1	0.95	0.95	10	0.1
	SARSA-FR	0.25	0.01	0.95	0.95	50	0.1
	SARSA-SRR	0.1	0.25	0.95	0.95	10	0.01
	SARSA-SR-PR	0.25	0.25(0.1)	0.95	0.95(0.99)	1	0.01
	SARSA-SR (fixed)	0.01	-	0.95	0.95	10	0.05
	SARSA-FR (fixed)	0.1	-	0.95	0.95	10	0.1
	SARSA-SRR (fixed)	0.25	-	0.95	0.95	10	0.01
SixArms	SARSA	0.5	-	0.95	-	-	0.01
	SARSA-SR	0.1	0.01	0.95	0.99	100	0.01
	SARSA-FR	0.1	0.01	0.95	0.99	100	0.01
	SARSA-SRR	0.01	0.01	0.95	0.99	10000	0.01
	SARSA-SR-PR	0.05	0.25(0.25)	0.95	0.95(0.99)	10	0.01
	SARSA-SR (fixed)	0.5	-	0.95	0.95	1	0.01
	SARSA-FR (fixed)	0.5	-	0.95	0.95	1	0.01
	SARSA-SRR (fixed)	0.5	-	0.95	0.95	10	0.01

$(0.1, 0.2, 0.2, 0.99, 0.95, 0.95, 1000, 0.3)$ for $(\alpha, \eta, \eta_{\text{PR}}, \gamma, \gamma_{\text{SR}}, \gamma_{\text{PR}}, \beta, \epsilon)$, where η_{PR} and γ_{PR} are the learning rate and discounting factor values for the PR, respectively.

Atari experiments. The neural architecture of the deep RL implementation shown in Figure 2, here we provide the specific hyperparameters of the architecture. The *Conv* block is a convolutional network with the configuration $(4, 84, 84, 0, 2) - ReLU - (64, 40, 40, 2, 2) - ReLU - (64, 6, 6, 2, 2) - ReLU - (64, 10, 10, 0, 0) - FC(1024)$, where the tuple represents a 2-dimensional convolutional layer with the architecture (num_filters, kernel_width, kernel_height, padding_size, stride), and $FC(1024)$ represents a fully connected layer with 1024 hidden units. We take the output of the *Conv* block as the 1024-dimensional state representation given the observation, which is then subsequently used for computing the SF and the PF. The action input is transformed into a high-dimensional embedding through a linear transformation, $FC(2048)$. The MLP for the predictive reconstruction block is $FC(2048) - ReLU$, for the Q-value estimation block is $FC(|\mathcal{A}|)$, for the SF head block is $FC(2048) - ReLU - FC(1024)$, for the PF head block is $FC(2048) - ReLU - FC(1024)$. The *Deconv* block is $FC(2048) - FC(1024) - ReLU - FC(6400) - Reshape((64, 10, 10)) - \langle 64, 6, 6, 2, 2 \rangle - \langle 64, 6, 6, 2, 2 \rangle - \langle 1, 6, 6, 0, 2 \rangle - Flatten$, where the tuple represents a 2-dimensional deconvolutional layer with parameters $\langle$ num_filters, kernel_width, kernel_height, padding_size, stride $\rangle$.