

Article

Tuning the Weights: The Impact of Initial Matrix Configurations on Successor Features' Learning Efficacy

Hyunsu Lee 

Department of Physiology, School of Medicine, Pusan National University, Yangsan 50612, Republic of Korea; hyunsu.lee@pusan.ac.kr

Abstract: The focus of this study is to investigate the impact of different initialization strategies for the weight matrix of Successor Features (SF) on the learning efficiency and convergence in Reinforcement Learning (RL) agents. Using a grid-world paradigm, we compare the performance of RL agents, whose SF weight matrix is initialized with either an identity matrix, zero matrix, or a randomly generated matrix (using the Xavier, He, or uniform distribution method). Our analysis revolves around evaluating metrics such as the value error, step length, PCA of Successor Representation (SR) place field, and the distance of the SR matrices between different agents. The results demonstrate that the RL agents initialized with random matrices reach the optimal SR place field faster and showcase a quicker reduction in value error, pointing to more efficient learning. Furthermore, these random agents also exhibit a faster decrease in step length across larger grid-world environments. The study provides insights into the neurobiological interpretations of these results, their implications for understanding intelligence, and potential future research directions. These findings could have profound implications for the field of artificial intelligence, particularly in the design of learning algorithms.

Keywords: successor feature; hippocampus; synaptic weight initialization; grid-world maze



Citation: Lee, H. Tuning the Weights: The Impact of Initial Matrix Configurations on Successor Features' Learning Efficacy. *Electronics* **2023**, *12*, 4212. <https://doi.org/10.3390/electronics12204212>

Academic Editor: Rashid Mehmood

Received: 12 September 2023

Revised: 2 October 2023

Accepted: 10 October 2023

Published: 11 October 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

For survival, animals are compelled to explore and interact with their environments. This interaction is underpinned by the ability to remember details about the environment, which enables animals to form expectations about future events or states based on their decisions. This capacity to predict outcomes based on past experiences is a cornerstone of intelligence. In animal cognition, the hippocampal system governs these predictive capabilities and memory functions [1].

The activity of place cells in the hippocampus has long been of interest in the context of learning and memory. These specialized neurons, integral to the brain's limbic system, activate when an animal finds itself in a particular location [2,3]. They essentially form a cognitive map or a neural embodiment of the spatial environment, which is pivotal for memory formation and learning. The discovery of place cells has led to numerous theories attempting to elucidate their role and the overarching function of the hippocampus in comprehending and learning spatial information.

Among a myriad of theoretical models, the successor representation (SR) has proven to be an influential explanation for the role of the hippocampus in spatial representation [4]. SR posits that the hippocampus forms a cognitive map that is not a static spatial representation but rather a dynamic anticipatory map that predicts future locations based on the current state [4,5]. The predictive map theory, interpreting place cell activity through the lens of SR learning, has shown considerable explanatory power for in vivo place cell activity. When an animal is first exploring an environment, its movements are random, and the expectation pattern appears symmetrical in all directions. As an animal becomes familiar with its environment, the activity of its place cells changes. While place cell activity

exhibits a geodesic pattern during initial exploration, it transitions to an asymmetrical firing pattern as the animal becomes accustomed to the environment [6].

According to the predictive map theory, the change in the firing pattern can be attributed to a shift in response from the act of visiting a specific location to the expectation of visiting that location. This shift in place cell activity results in a pattern that leans toward the animal's direction of movement because the expectation increases as the animal nears the location. The SR theory goes beyond predicting the immediate subsequent state, suggesting that the hippocampus forecasts all future states. This ability to construct a predictive map of the environment encompasses the animal's anticipations of future states, given its current state and behavior. This model elegantly bridges the gap between spatial navigation and reinforcement learning [7–10].

While SR provides a compelling explanation for place cell activity patterns, it presumes the animal has comprehensive knowledge of the environment's size and fully observable location information. To overcome this limitation and extend the predictive theory to partially observable environments, recent studies have proposed a feature-based SR model as a representation of hippocampal activity [5,11].

In contrast to the traditional SR learning, which employs tabular methods and views each state as a separate entity, feature-based SR—also known as successor feature (SF)—employs a neural network as a function approximator to learn SR [11]. This adaptation equips SR learning with the capacity to manage high-dimensional state spaces, making SF a more plausible neurobiological model than its naive counterpart. Nevertheless, the approach to initialize the weight matrix of the neural network varies significantly across the literature [5,11]. Synaptic weights play a crucial role in neural networks, affecting the speed and success of learning. Despite their crucial role, the influence of different weight matrix initialization methods on overall learning remains largely unexplored.

In this research, our objective is to investigate the influence of varying synaptic weight initialization patterns on SF learning. By subjecting SF learners to a basic maze environment under differing weight initialization patterns, we aim to illuminate the role of weight initialization in the SF learning process. For the evaluation of these impacts, we conducted an experiment utilizing identity, zero, and random matrices for weight initialization. With an ϵ -greedy policy, the performances of the SR agent and the non-random SF agents were observed to be comparable in a one-dimensional (1D) maze. However, SF agents with randomly initialized weight matrices exhibited superior performance compared to their non-random counterparts. In the results section, we delve into the changes in the SR matrix throughout the learning process. Furthermore, in the discussion section, we reflect upon the neurobiological implication of weight matrix initialization. This investigation contributes to the continuous pursuit of understanding intelligence from both neuroscientific and artificial intelligence viewpoints.

2. Model and Methods

2.1. Successor Representation (SR)

In this study, we assume that an RL agent interacts with the environment through Markov decision processes (MDP, [12,13]). An MDP is a tuple $M := (\mathcal{S}, \mathcal{A}, R, \gamma)$ comprising the following elements. Sets \mathcal{S} and \mathcal{A} are the state (e.g., spatial locations) and action spaces, respectively. The function $R(s)$ specifies the immediate reward received in state s , which can be expressed as $R : \mathcal{S} \rightarrow \mathbb{R}$. Here, the discount factor $\gamma \in [0, 1)$ is a weight that reduces the reward in the distant future.

In RL, the agent's objective is to discover a policy function $\pi : \mathcal{S} \mapsto \mathcal{A}$ that maximizes the cumulative discounted reward, often referred to as the return $G_t = \sum_{i=t}^{\infty} \gamma^{i-t} R_{i+1}$, where $R_t = R(S_t)$. The return essentially represents the sum of all future rewards that an agent can expect to accumulate, discounted by the factor γ . In order to solve this

optimization problem, a common approach is to employ dynamic programming, which defines and computes the value function of a policy π as follows:

$$V^\pi(s) := \mathbb{E}^\pi[G_t | S_t = s], \quad (1)$$

where $\mathbb{E}^\pi[\cdot]$ denotes the expected value when the agent follows policy π . After determining $V^\pi(s)$, also known as policy evaluation, the policy π can be improved in a greedy manner. This process, referred to as greedy policy improvement, is defined as follows: $\pi'(s) \in \operatorname{argmax}_a Q^\pi(s, a)$, where $Q^\pi(s, a) := \mathbb{E}[R_{t+1} + \gamma V^\pi(S_{t+1}) | S_t = s, A_t = a]$ [13]. Here, $Q^\pi(s, a)$ represents the expected return from taking action a in state s and following policy π thereafter.

As proposed in the literature [14], the central premise of SR learning lies in the decomposition of the value function (Equation (1)). It suggests that the value function can be decomposed into an expected visiting occupancy and reward of the successor states s' as follows:

$$\begin{aligned} V^\pi(s) &= \sum_{s'} \mathbb{E}^\pi \left[\sum_{i=t}^{\infty} \gamma^{i-t} \mathbb{I}(S_i = s') R(s') | S_t = s \right] \\ &= \sum_{s'} M(s, s') R(s'), \end{aligned} \quad (2)$$

where $\mathbb{I}(S_i = s')$ yields a value of 1 when an agent visits the successor state s' at time t ; otherwise, it returns 0. Consequently, $M(s, s')$ represents the discounted expectation of visitation to the successor state s' from the state s . $M(s, s')$ can be perceived as a comprehensive representation that integrates not only the immediate transition probabilities from state s to state s' but also the cumulative impact of the agent's policies and the array of potential future trajectories. This interpretation underscores the dynamism and predictive capacity of the SR, as it encapsulates the influence of the agent's decisions and environmental dynamics on future state visitations [14].

The SR matrix M can be incrementally learned by the agent through the use of the temporal difference (TD) learning algorithm. This approach allows the agent to continually update its understanding of the environment based on the difference between the predicted and actual visitation. The specific TD learning equation for the SR matrix M is derived as follows [4,14]:

$$\Delta M(s_t, s') = \alpha_M [\mathbb{I}(s_t = s') + \gamma M(s_{t+1}, s') - M(s_t, s')]. \quad (3)$$

2.2. Feature-Based SR

The classical form of SR learning is constrained to tabular environments, limiting its applicability to more complex high-dimensional settings [15]. An effective means of circumventing this limitation is the application of a set of feature functions, denoted as $\psi(s)$, which allows for the generalization of SR learning [11].

By assuming that the expected reward of state s can be represented as a product of the feature vector and its corresponding reward weights, denoted as $R(s) = \phi(s)^T \mathbf{w}_{rew}$, we can reframe the value function (Equation (1)) in a way that accommodates these feature functions. The revised value function is given as follows:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}^\pi \left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi_{i+1} | S_t = s \right]^T \mathbf{w}_{rew} \\ &:= \psi^\pi(s)^T \mathbf{w}_{rew}, \end{aligned} \quad (4)$$

where ϕ_t denotes $\phi(S_t)$. By incorporating a one-hot vector in $\mathbb{R}^{|S|}$ for a tabular environment, $\psi(s)$ essentially mirrors the $M(s, :)$ vector of SR learning. This is because it represents the discounted sum of occurrences of $\phi(s')$ when a transition unfolds under policy π .

For clarity, we henceforth refer to $\psi^\pi(s)$ as the SF associated with state s under policy π .

The introduction of SF marks a significant broadening of the SR learning framework, facilitating its application across a wider spectrum of MDP environments, such as partially observable MDPs and those characterized by continuous states [16].

In our approach, the SFs are approximated utilizing a linear function represented as follows:

$$\hat{\psi}(s) = W^T \phi(s). \quad (5)$$

This estimation leans on the presumption that $\phi(s)$ operates as a population vector of neurons that responds to the state s observed by an agent. The utilization of a linear function aligns with neurobiological models of hippocampal place cells and finds support in the literature, reinforcing its relevance and applicability in our research [4,5,15].

To estimate $\psi(s)$, we apply the TD learning to update the weight matrix W . This procedure parallels the matrix M updating method observed in successor representation (SR) learning, thereby offering a streamlined approach to SF estimation in reinforcement learning contexts.

$$\Delta W = \alpha_W [\phi(s_t) + \gamma \hat{\psi}(s_{t+1}) - \hat{\psi}(s_t)] \phi(s_t)^T \quad (6)$$

It is worth highlighting that Equation (6) corresponds to Equation (3) when $\phi(s)$ is presented as a one-hot vector. Alongside this, the expectation weight vector associated with rewards, denoted as \mathbf{w}_{rew} , is updated using a simple delta rule [5,13] as follows:

$$\Delta \mathbf{w}_{rew} = \alpha_r (R_t - \phi(s)^T \mathbf{w}_{rew}) \phi(s). \quad (7)$$

With the established update rules, we are now ready to investigate the SFs' learning with different initialization methods of the weight matrix W . Notably, the initial values of weight matrix W are assumed to play a critical role in the learning performance and efficiency.

2.3. SF Learners and Their Weight Initialization Patterns

To explore the influence of different weight initialization methods on the learning dynamics of SFs, we initialized the weight matrix W using three different methods: identity, zero, and small random matrices.

2.3.1. Identity Matrix Initialization

The identity matrix initialization method sets the initial weight matrix W as an identity matrix, $W = I$. This means that the initial estimates of the SFs are equivalent to the one-hot encoded state representations. This initialization strategy can be regarded as a “knowledgeable initialization”, endowing the agents with preliminary information about the environment [5].

2.3.2. Zero Matrix Initialization

In contrast, the zero matrix initialization method sets all elements of the initial weight matrix W to zero. This means the SFs initially predict no future state visitations, assuming no knowledge of the world at the initial state. This initialization method can be seen as a “naive initialization”, where agents start learning from scratch without any prior knowledge about the environment.

2.3.3. Small Random Matrix Initialization

Small random matrix initialization, a commonly employed method in machine learning, sets the initial weight matrix W with small random values drawn from specific distributions. This technique infuses randomness into the preliminary estimates of SFs, conjecturing

a mixture of accurate and imprecise understanding of the world at the onset [11]. We employ a single layer for the successor feature. Given that the expected future visitation cannot be negative, we ensure that the weights are initialized randomly within the positive domain by applying an absolute value function. For this investigation, we utilized three prevalent techniques to initialize small random matrices: the Xavier method, the He method, and a uniform distribution.

Xavier Method

The Xavier method, also known as Glorot initialization [17], is a popular method for weight initialization in deep neural networks. This method determines initial weights by drawing a random number from a uniform probability distribution (U) within the range of $-\frac{1}{\sqrt{n}}$ to $\frac{1}{\sqrt{n}}$. In our study, ‘ n ’ corresponds to the number of input neurons, thereby representing the size of the one-dimensional (1D) grid world.

He Method

The He initialization technique [18], another approach utilized in this research, derives initial weights from a Gaussian probability distribution characterized by a mean of zero and a standard deviation given by $\sqrt{2/n}$, where ‘ n ’ symbolizes the number of input neurons.

Uniform Distribution

The uniform distribution method represents the most straightforward approach to initializing small random matrices. In our study, this method involved distributing weights uniformly across an interval ranging from 0 to 0.1. This choice of distribution infers that we hold the expectation of future visitation to each successor state as uniformly probable.

2.4. Experimental Setup

To investigate the learning process of each RL agent, we used a simple one-dimensional (1D) grid world of size $N \in \mathbb{N}$ spanning from 3 to 100 cells. In this environment, the agent navigates the grid world using left and right actions (Figure 1). In every episode, the agent starts at the leftmost position in the grid world. The ultimate goal is to reach the rightmost end of the world (also known as the terminal state). Upon reaching this terminal state, the agent receives a reward of 1 point. In contrast, all other states receive a score of 0, which means no reward. In our investigation, the discount factor γ was set to 0.95.

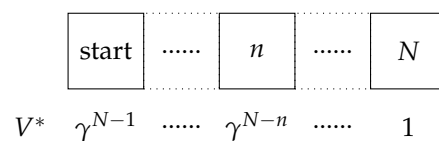


Figure 1. Schematic of the 1D grid world following the MDP. V^* represents the expected true value of each cell according to discount factor (γ) when the reward of the terminal state is one.

To maximize the overall discounted reward, the agent selects actions predicated upon the estimated Q value utilizing an ϵ -greedy policy. This policy prescribes a uniform random action selection with probability ϵ , while at other times, with a probability of $1 - \epsilon$, the agent chooses the action associated with the highest Q-value estimate. To foster adequate exploration and promote learning stabilization, the probability ϵ undergoes a decay according to the rule: $\epsilon_k = 0.9 \cdot 0.95^k + 0.1$, where k signifies the episode index [19].

The learning rates assigned to each learner—the matrix M for the SR agent and the matrix W for the SF agents—were uniformly set at $\alpha_M = \alpha_W = 0.1$. The learning rate allocated to the reward position vector, for both the SR agent and the SF agents, was fixed at $\alpha_r = 0.1$. To equitably compare SR and SF agents, maze environment observations were utilized as state indices for the SR agent and one-hot coding vectors for the SF agents.

2.5. Performance Evaluation Metrics

We evaluated the learning performance of SF agents with different weight initialization methods based on several metrics, including learning speed, final performance, and stability of learning. In addition to these performance metrics, we also analyzed the changes in the SR matrix and the weight matrix throughout the learning process. These analyses allowed us to better understand the dynamics and mechanisms underlying the influence of the weight initialization on SF learning.

In this evaluation, each agent simulation test was run 10 times, and the mean and standard deviation of the results are presented in the experimental results section.

2.5.1. Evaluating the Evolution of SR Place Field Matrix

In an effort to elucidate the intricacies of the SR matrix's evolution and convergence patterns over the progression of episodes, we utilized Principal Component Analysis (PCA)—a powerful dimensionality reduction tool [20]. This process was complemented by calculating the L1 distance between matrices at various stages throughout the learning episodes. This measurement helped in detailing the patterns of convergence inherent to the SR matrix as the agent gained expertise within the simple maze environment. This measurement was computed as follows:

$$d_1(A, B) = \sum_{i=1}^N \sum_{j=1}^N |a_{ij} - b_{ij}|. \quad (8)$$

In this formula, $a_{i,j}$ and $b_{i,j}$ denote individual elements within the SR place field matrices A and B , respectively. This methodological approach offers a comprehensive portrayal of the SR matrix's conversion throughout the unfolding learning episodes.

2.5.2. Value Error

Within a one-dimensional maze that begins from the leftmost position, the optimal policy would invariably guide movements towards the right. Accordingly, the legitimate value of the n -th grid cell, denoted as $V(s_n)$, amounts to γ^{N-n} , with s_n representing the n -th grid cell. This investigation entailed a comparison of the learning efficiency among diverse agents, relying on the mean square error (MSE) as a metric. The MSE is the discrepancy between the true value function, V^* , and the value function under the current policy, V^π . It is mathematically represented as follows:

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N-1} (V^*(s_n) - V^\pi(s_n))^2. \quad (9)$$

Apart from the aforementioned metric, we also incorporated an alternative measure, defined as $-\frac{\Delta \text{MSE}}{\Delta \text{episode}}$. This metric specifically provides insight into the rate at which the value error diminishes over time, hence offering an additional perspective on learning efficiency.

2.5.3. Step Length

In our analysis, we utilized additional metrics to understand the agent's learning progression in a comprehensive manner. The step length, representing the number of steps the agent takes to reach the goal, was one such measure. As the agent improves its policy with learning, this step length is expected to decrease. We assessed the rate of change of the step length over episodes, defined as $\frac{\Delta(\text{step length})}{\Delta \text{episode}}$, to gain insights into the speed of the agent's learning and the pace of policy improvement.

Further, we also evaluated the variability in the rate of step length decrease to understand the stability of learning. This was accomplished by calculating the standard deviation of $\frac{\Delta(\text{step length})}{\Delta \text{episode}}$ over the course of learning episodes. This metric allows us to gauge the

consistency of the agent's learning progress, providing a more complete picture of the learning dynamics.

3. Experimental Results

3.1. Accelerated Convergence of Random SF Agents to Asymmetrical SR Place Fields

To elucidate the impact of disparate initial matrix forms on SF agents, we analyzed the transformational learning history of the weight variables of SF agents, juxtaposing the findings with those of the SR agent.

3.1.1. Learning History of SR Place Field

In Figure 2, we present characteristic results simulated in a grid world comprising 100 cells. Upon comparison of the learning pattern of the 50th cell's SR place field across agents, we noticed that the SF agents equipped with random weights (Xavier, He, and uniform) exhibited an expedited shift towards asymmetrical SR place fields relative to their non-random counterparts (SR, identity matrix, and zero matrix).

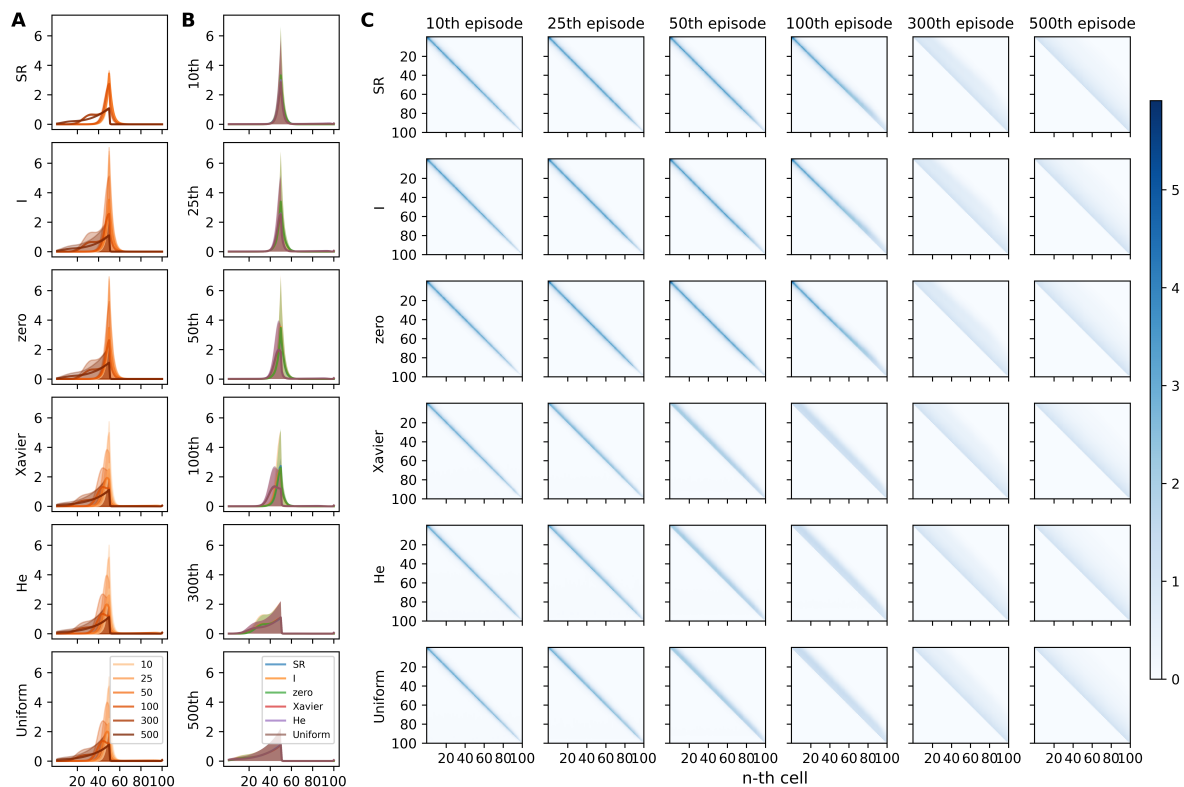


Figure 2. The simulated learning histories of the SR place field show that the SF agents, with the initial weight set with random weights, rapidly converge to the asymmetric SR place field. (A) Line plots of the learned SR place field of 50th cell after the end of the 10th, 50th, 100th, and 300th episode are shown. Each line and shade show the averaged result with the standard deviation from 10 simulations in a grid world with 100 cells. Each row panel displays the SR agent (first row) or SF agents with different weight initialization methods (five rows below). (B) Rearranged line plots from (A) comparing the agents (SR, blue; SF weight initialization with identity matrix, orange; zero matrix, green; the Xavier method, red; the He method, purple; the uniform distribution, brown). Each row panel displays the simulated results after the end of the 10th, 25th, 50th, 100th, 300th, and 500th episode. Note that the SF agents with random weights show skewed SR place fields at the 50th episode, but other agents show symmetrical SR place fields. (C) Learning histories of the whole SR place field matrix in a grid world with 100 cells are shown according to episodes (column panels) and learning agents (row panels).

In detail, the SR place fields of the 50th cell for non-random agents retained a symmetrical pattern even at the 50th and 100th episodes (Figure 2B). In contrast, when we inspected the learned pattern of the comprehensive SR matrix at the 50th episode (Figure 2C), it became evident that the SR place fields of random agents already displayed an asymmetrical pattern. This held true even for cells located proximally to the first cell. On the other hand, non-random agents, with the exception of those in the vicinity of the goal location, continued to exhibit a symmetrical pattern in their SR place fields.

These observations underscore the intriguing finding that SF agents with random initial weights converge more rapidly towards asymmetrical SR place fields compared to non-random agents. This facet is especially pronounced in the early stages of learning, which can have implications on the temporal dynamics of learning and overall task performance.

3.1.2. Analyzing SR Matrix Changes with PCA

The comprehensive SR matrix embodies the combined responses of all place cells, thereby collectively representing the entirety of the grid world. Consequently, to analyze and monitor the alterations in the learning pattern across episodes, a dimensionality reduction of the SR matrix is crucial. Borrowing methods from neuroscience research that are employed to examine large-scale neuronal recordings [21], we utilized PCA for this purpose.

As hypothesized, and in alignment with the earlier observed transformations in the SR place field, our findings revealed that agents initialized with random weights followed a more direct path towards convergence (Figure 3).

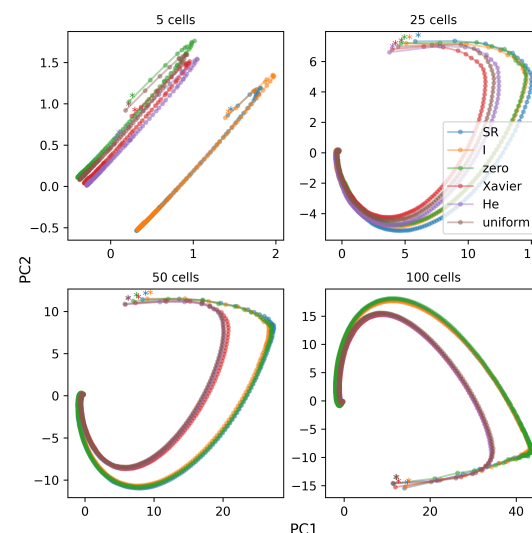


Figure 3. Principal component analysis (PCA) of the SR place field matrix learning history shows that SF agents with random weights take shorter routes to converging optima. The simulated results from four different sizes of grid worlds ($N = 5, 25, 50, 100$) are shown. Each dot shows the PCA results of SR place fields after each episode (*, first episode). Each line shows the historical route of the SR place field learning from SR or SF agents (SR, blue; SF weight initialization with identity matrix, orange; zero matrix, green; the Xavier method, red; the He method, purple; the uniform distribution, brown). The average of the SR place field matrices from 10 simulations was used for PCA. The distinct trend in the fourth image arises from a PCA of a large weight matrix. Its unique pattern, differing from the other images, may reflect specific variances or features within the matrix. The exact cause remains an area for further exploration.

In the case of a relatively compact grid world ($N = 5$), a similar progression pattern in the SR place matrix was noticeable between the SR agents and SF agents initialized with identity matrices (the upper left panel of Figure 3). On the same note, agents initialized

with random weights and SF agents with zero matrix initialization demonstrated analogous navigation patterns. These patterns, however, start to diverge with the expansion of the grid world ($N = 25$). Here, random agents appear to emulate each other's trajectories, just as non-random agents do (the upper right panel of Figure 3). As we delve into larger grid worlds, for instance, $N = 50$ and $N = 100$, the distinctions between the random and non-random agents become increasingly apparent (the lower panels of Figure 3). It is in these settings that agents initialized randomly show a propensity for shorter convergence paths.

It is important to note that there are differences in the scale of the axes, and for the same axis scale, readers are referred to Figure A1.

3.1.3. Inter-Agent SR Matrix Distance

The PCA results suggest an intriguing possibility: if the randomly initialized agents achieve faster convergence towards the optimal SR place field, the distance between their SR place matrices and those of non-random agents should escalate during the learning process, eventually plateauing upon convergence. Conversely, the distance between the SR place matrices of non-random agents would remain relatively constant.

To investigate this possibility, we computed the L1 distances between the SR matrices of the agents, presenting the results as a function of learning episodes (Figure 4). The trends in the L1 distances over episodes among the six agents support our prediction, with the distance between random and non-random agents initially increasing before decreasing once again. Conversely, there is no discernible increase in the L1 distance when comparing either within the group of random agents or the group of non-random agents. Owing to the larger SR matrices and consequently larger distances found in larger grid worlds, all y-axes in Figure 4 are plotted on a logarithmic scale.

To mitigate the influence of the SR matrix's size on the L1 distance, we can divide the L1 distance by the total number of elements in the matrix ($N \times N$). This normalization procedure brings the metric down to the level of a single SR matrix element and further emphasizes that the distance between the randomly initialized agents and the non-random ones tends to increase (Figure A2).

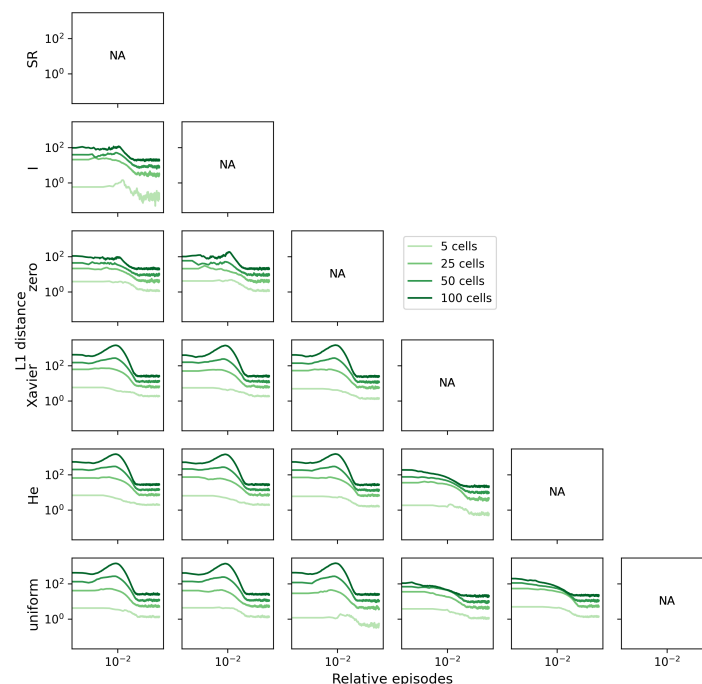


Figure 4. The L1 distance between SR place fields of agents shows that the SF agents with random weights differ from other non-random agents. Line plots show the change in the L1 distance according to episode in four different sizes of grid worlds ($N = 5, 25, 50, 100$). Since the total number of episodes depends on the grid world size, the relative episodes ($\frac{\text{episode}}{\text{total number of episodes}}$) are shown on the x-axis.

3.2. Enhanced Value Error and Step Length Reduction in Random Agents

Drawing on Equations (2) and (4), a direct correlation can be established between the variance in the SR matrix learning and the RL agent's performance. Herein, we analyze the anticipated value and step length to highlight the performance differences in the learning process.

3.2.1. Examination of Mean Square Error Decline Rate

Taking into consideration the ground truth value (V^*), the MSE of the estimated value (V^π) was calculated (please refer to Equation (9) in Section 2.5.2). Consistent with our expectations, we observed that the MSE of V^π for the random agents diminished at a faster pace than for the non-random agents (the upper panel of Figure 5A).

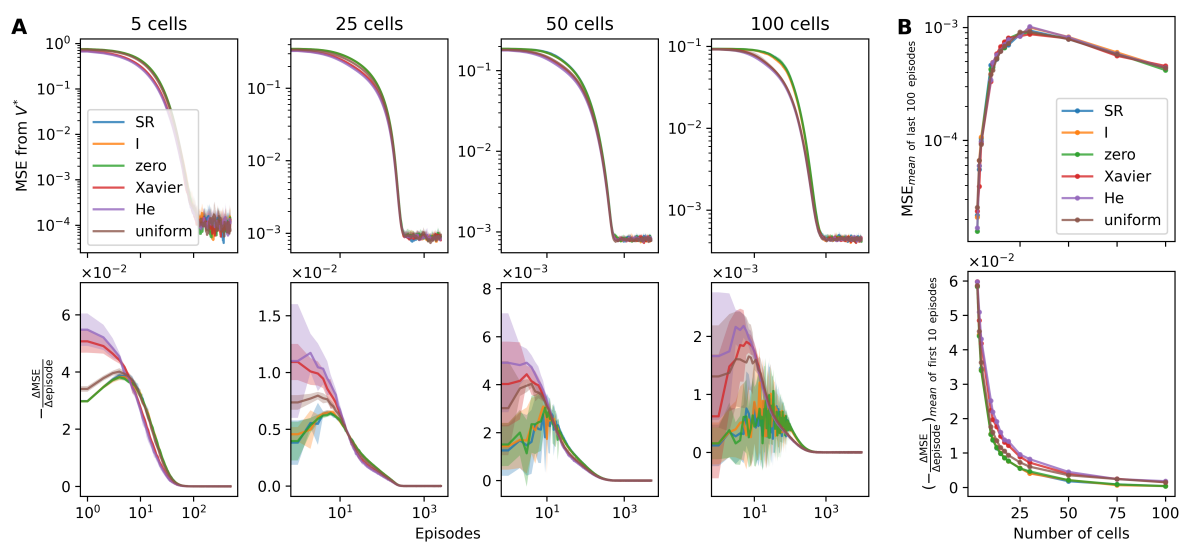


Figure 5. The mean square error of values shows that the estimated value of the SF agents with random weights decreases to the true value faster than the non-random agents. (A) The upper panel shows the mean square error (MSE) of the estimated values (V^π) decreases as the episode progresses. The lower panel shows the decrease in the MSE per single episode ($-\frac{\Delta \text{MSE}}{\Delta \text{episode}}$). The results are from 10 simulations in four grid worlds of different sizes (arranged by columns). The averages (lines) and standard deviation (shades) of the SR or SF agents (SR, blue; SF weight initialization with identity matrix, orange; zero matrix, green; the Xavier method, red; the He method, purple; the uniform distribution, brown) are shown. (B) The upper panel shows that the averages of the MSE from last 100 episodes are similar across the SR or SF agents. The lower panel shows that the average of $-\frac{\Delta \text{MSE}}{\Delta \text{episode}}$ from first 10 episodes of the SF agents with random weights are larger than the non-random agents. Each circle marker indicates the size of grid worlds, which were simulated.

We examined the rate of MSE decrement, represented as $-\frac{\Delta \text{MSE}}{\Delta \text{episode}}$ (the lower panel of Figure 5A). Early episodes, up to the 10th, illustrated a higher decrement rate in the randomized agents, indicating a more rapid reduction in the MSE value. It is noteworthy that among the random agents, those initialized utilizing the He and Xavier methods depicted a steeper reduction relative to those initialized uniformly (the lower panel of Figure 5A,B).

Nevertheless, in the latter episodes, the rate of MSE reduction exhibited minimal variation across the agents, underlining the comparable efficiency of the initialization methods in the long run.

3.2.2. Step Length Reduction

Given the quick reduction observed in the MSE of random agents, we can anticipate a corresponding accelerated decline in the step length to the goal cell within each episode of the grid world exploration. Due to their initially high ϵ probability, all RL agents undertake

an exploration of the grid world that mimics a random walk, which naturally results in longer step lengths during the early episodes. As shown in Figure 6, as the exploration episodes advance, the step length predictably shrinks to the size of the grid world.

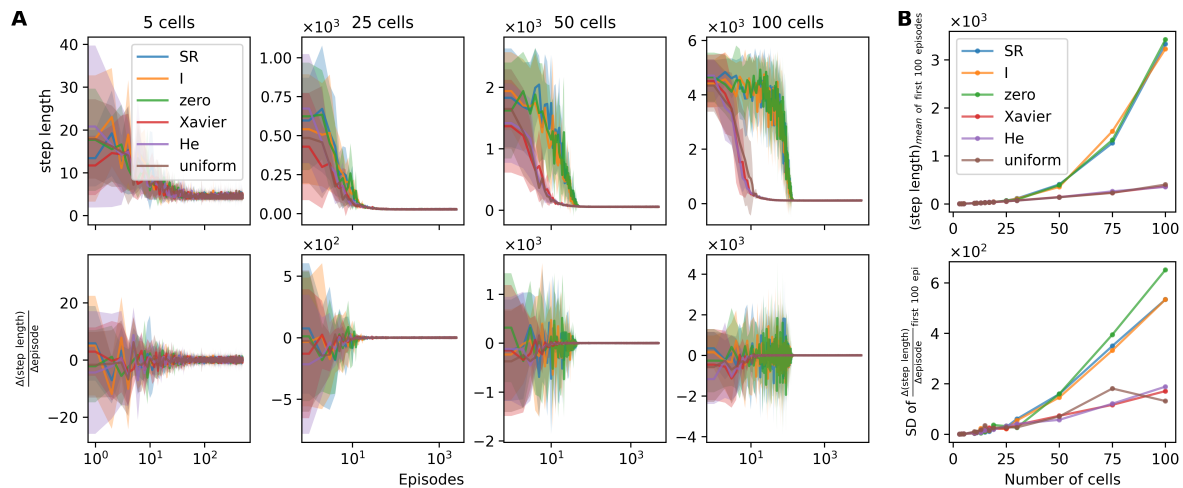


Figure 6. The SF agents with random weights converge to the optimal step length more rapidly and stably than the non-random agents. **(A)** The upper panel displays the total step length taken to reach the target state for each episode. In the simulation results from the large grid world ($N = 100$), the step length of the SF agents with random weights decreases to the ideal step length for the first 10 episodes, while the non-random agent decreases to the ideal step length after hundreds of episodes. The lower panel shows the decrease in total step length with each episode ($\frac{\Delta(\text{step length})}{\Delta\text{episode}}$). In the simulation results of the large-scale grid world ($N \geq 50$), the jittering of $\frac{\Delta(\text{step length})}{\Delta\text{episode}}$ of the SF agents with random weights disappears after 10 episodes, whereas its jittering of the non-random agents persists. The results are from 10 simulations in four grid worlds of different sizes (arranged by columns). The averages (lines) and standard deviation (shades) of the SR or SF agents (SR, blue; SF weight initialization with identity matrix, orange; zero matrix, green; the Xavier method, red; the He method, purple; the uniform distribution, brown) are shown. **(B)** The top panel shows the average step length of the first 100 episodes according to grid world size. The lower panel shows the standard deviation of $\frac{\Delta(\text{step length})}{\Delta\text{episode}}$ of the first 100 episodes according to grid world size.

In smaller grid worlds (where $N < 30$), no significant differences in the reduction of step lengths amongst the RL agents were observed. However, as the grid world's size expands ($N \geq 50$), it was noted that the step length of random agents diminished at a faster rate (see Figure 6B, upper).

The trajectory of step length reduction manifested clear distinctions between the two groups. Non-random agents demonstrated significant fluctuations in the decrement of step length, while such variance was less prevalent in random agents. This disparity was further illustrated by calculating the rate of step length reduction, $\frac{\Delta(\text{step length})}{\Delta\text{episode}}$, and evaluating its standard deviation (the lower panel of Figure 6A,B).

For non-random agents, the fluctuations in the rate of step length reduction inflated exponentially with the increase in grid world size. Conversely, for the random agents, the fluctuations displayed a linear growth pattern despite the expanding grid world size, indicating a more stable decrease in step length as the learning process progressed.

4. Related Works

Effective initialization methods, contingent upon the activation function, have been well documented in the study of Artificial Neural Networks (ANN) utilizing backpropagation algorithms. The normalized Xavier initialization method [17] is typically employed with sigmoid and tanh functions, while the He initialization method [18] sees frequent usage with ReLU functions.

In this study, the absolute values derived from the Xavier and He methods were employed to initialize the random agent. It is worth noting that the inclusion of negative numbers in the weight matrix can result in a negative SR value corresponding to future occupancy, as we make use of a single-layer function approximator devoid of an activation function. To address this issue, we can utilize multilayer ANNs as a function approximator. When a deep neural network is employed as an SF approximator [22,23], it begs the question as to which activation function in the hidden layer is optimal and, consequently, the most effective weight initialization method.

Though this paper focused on exploring MDP-based agent learning of environmental characteristics via the SF algorithm, numerous other algorithms are available that describe animal–environment interactions and learning mechanisms, for instance, Particle Swarm Optimization (PSO) that models avian foraging behavior [24,25]. Among the latest advancements to the PSO algorithm, multi-swarm PSO has been successfully implemented in feature learning for sentiment analysis of Massive Online Open Course lecture reviews [26].

5. Discussion

In this study, we investigated the role of the initial weight matrix configurations in the efficiency of SF learning. We scrutinized three initialization methods: the identity matrix, zero matrix, and random matrix (using Xavier, He, and uniform distribution). Our results demonstrated that the randomized agents, regardless of the specific initialization method, outperformed the identity and zero matrix agents. Specifically, we found that the random agents learned faster, which was evident from the decrease in the MSE of the estimated value and step length to the goal cell in a grid world environment. Further, PCA analysis illuminated the distinct patterns of learning in randomized agents compared to non-randomized ones, which provided additional insight into the evolution of SR place matrix. Thus, our findings underscore the significant influence of the initial weight configurations on the effectiveness and speed of SF learning.

5.1. Interpretation of SF Weight Matrix Initialization

Initiating the SF weight matrix as an identity matrix provides the agent with a unique starting position in its learning journey about the environment. As learning progresses, each element in the identity matrix corresponds to a particular state, thereby facilitating the updating of knowledge regarding state transitions. However, this initialization method could restrict the agent's versatility in exploring and learning diverse environmental patterns, potentially resulting in slower learning as observed in previous research. This limitation could be particularly consequential for an agent's adaptability in increasingly complex or dynamic environments.

Alternatively, initializing the SF weight matrix with zero establishes a “tabula rasa” situation for the agent. Devoid of any prior knowledge, these agents are heavily influenced by their environmental interactions and the inherent learning algorithm. Although this approach broadens the exploration scope, it may decelerate learning due to the absence of initial guidance. This downside was apparent in studies where agents initiated with a zero matrix took a longer time to converge compared to their randomly initialized counterparts.

Contrastingly, random matrix initialization strikes a balance between exploration and exploitation. Incorporating random elements into the SF weight matrix equips the agent with a degree of “innate knowledge” guiding its initial steps while preserving a vast spectrum for exploration and learning. Consequently, this initialization method may enhance the learning efficiency, offering a promising avenue for improving SF learning algorithms.

5.2. Neurobiological Considerations

In the context of RL, the feature vector of the input layer offers a snapshot of the agent's current position. Subsequently, this information is transformed by the SF weight matrix into a population vector, effectively encoding the anticipated future occupancy

given the policy at hand. This sequence of operations bears striking resemblance to the neurobiological mechanisms believed to underpin spatial learning.

A collection of studies [4,5,7] suggests that hippocampal CA1 place cells encode SR through population codes. Viewed through this neurobiological lens, the SF weight matrix may be interpreted as a close analog to the synaptic weights connecting CA1 place cells to preceding layers of neurons in the neural hierarchy, such as those in the CA3 and entorhinal cortex. This parallel between the functioning of RL algorithms and the neuronal processes that facilitate spatial learning lends support to the use of such algorithms in the investigation of cognition and its underlying biological substrates.

While the exact mechanisms by which the brain might implement the synaptic update rule used in our study remain elusive, a body of research has found substantial evidence that TD learning parallels the activity of dopaminergic neurons in response to reward prediction errors [27,28]. This aligns with the hypothesis that the neural instantiation of TD learning might be facilitated through neuroplasticity rules, such as spike timing-dependent plasticity and heterosynaptic plasticity [15,29]. This conjecture, if further corroborated, could add an extra layer of understanding to our exploration of the intersections between artificial intelligence and neurobiology.

From a biological standpoint, it seems reasonable to posit that place coding and reward prediction coding might be processed in tandem within the brain, which subsequently synthesizes these elements into anticipated values for a given state. This line of thought supports the perception of the brain as a device engaging in parallel distributed processing, as suggested by [30]. Underpinning this proposition, the backpropagation algorithm has exhibited exceptional capability in tasks such as image recognition [31,32]. Moreover, a convolutional neural network (CNN) trained with this algorithm has exhibited activation patterns that bear resemblance to those observed in the visual cortex and the inferior temporal cortex of the brain [33]. Notably, when the activation pattern of a trained CNN was used to manipulate an image, it was found to predict neuronal responses in the V4 visual cortex of macaque monkeys [34]. Nevertheless, it is still a matter of ongoing debate and remains unconfirmed whether the backpropagation algorithm is genuinely operative within the brain [35,36].

While the biological embodiment of SR learning remains elusive, particularly regarding the brain's processing location and method for the inner product of the feature vector and reward vector, there is a notable correlation between the outcomes of SR learning and the behavior of hippocampal place cells [4,37,38]. However, it warrants further exploration to fully understand how the brain learns and signifies the sequences of state transitions, rewards, and state values. Experimental findings have associated the representation of the reward signal with the orbitofrontal cortex (OFC) [39,40], suggesting the anterior cingulate gyrus as a probable area for the integration of the OFC's reward signal and the HPC's SF signal [41,42]. In contrast, a study by [43] postulated that the HPC directly encodes the position of the reward.

Transitioning our focus to the question of 'how', we are confronted with the challenge of extracting a scalar value from the successor feature vector and reward vector [44]. Although this issue extends beyond the scope of the current study, we can glean some neurobiological insights. Specifically, if the synaptic weights in a neural network at the developmental stage are randomly initialized, they demonstrate faster convergence to an optimal state.

5.3. Limitations of the Study

While our study offers valuable insights into the impact of SF weight matrix initialization on learning efficiency and convergence, it is important to note that these findings are based on a one-dimensional grid world. The 1D grid world was chosen for its simplicity, allowing for clear analysis and interpretation. However, the learning patterns and efficiency we observed may vary with different types of environments. For instance, we noticed more

distinct learning trajectories in larger grid worlds, while smaller grid worlds exhibited similar patterns.

In more complex environments, such as two-dimensional or three-dimensional grid worlds, the state space grows, and agents may face intricate navigation challenges influenced by factors like obstacles, multiple goals, or varying reward structures. Therefore, the grid-world size is a potential limitation of our study, and our conclusions might be more applicable to larger grid worlds.

Recognizing this, we believe it is crucial to investigate the proposed methods in varied scenarios to understand their robustness and generalizability better. Future studies could broaden the scope by investigating a wider range of MDP environments, such as two-dimensional grid worlds, to enhance the applicability of our findings and provide a comprehensive understanding of the learning dynamics in different contexts.

Matrix initialization plays a pivotal role in the behavior and performance of optimization algorithms, especially in the context of neural network models [17,18]. Various initialization methods have been explored because different strategies can lead to variations in convergence rates, sensitivity to local minima, and overall model performance. However, our research does not delve into methods such as orthogonal initialization and sparse initialization.

While we opted for initialization methods that are well-established in supervised learning and deep learning literature, it is evident that the landscape of matrix initialization is vast and multifaceted. Future research could provide a comparative analysis of these methods in different contexts, shedding light on their strengths, limitations, and optimal use cases.

Our study relied on certain evaluation metrics, including the MSE of the value error, step length, and the PCA of the SR place matrix, to analyze the learning efficiency and convergence. While these metrics provided significant insights, they might not encapsulate all aspects of an agent's learning trajectory. For instance, MSE and step length predominantly focus on the speed of learning, potentially overlooking other critical dimensions such as the stability and adaptability of learning. Additionally, PCA, while effective in dimensionality reduction and visualizing high-dimensional data, may oversimplify complex learning patterns.

6. Conclusions

This study embarked on an exploratory journey into the role of matrix initialization in SF learning within the framework of RL. We discovered notable differences in the learning trajectories of agents with different matrix initialization forms—identity, zero, and random (Xavier, He, and uniform distribution). Our findings suggest that random matrix initialization, particularly using the Xavier and He methods, led to more efficient learning and faster convergence to the optimal state, as evidenced by a quicker decrease in the value error and step length. The PCA further revealed distinct patterns of SR place matrix evolution among different agents, reinforcing the importance of matrix initialization in shaping the learning dynamics.

The study highlights the significance of weight initialization in the learning process. Our observations demonstrate that the choice of initialization method significantly influences the learning trajectory and efficiency of the agents. Specifically, agents initialized with random matrices demonstrated accelerated learning and quicker convergence to the optimal state. These findings underline the value of exploring diverse initialization techniques to enhance the effectiveness of SF learning.

The implications of this research extend beyond SF learning and RL, contributing to our broader understanding of intelligence from both a neuroscientific and artificial intelligence perspective. By drawing parallels between SF learning and the functioning of place cells in the brain, the study offers intriguing insights into the neurobiological processes underlying learning. An intriguing direction for future research is to delve deeper into the parallels and disparities between biological learning and AI learning algorithms.

The results of this study shed light on the learning efficiency of agents, mirroring the learning process of place cells in the brain. Continued efforts to bridge this gap could lead to the development of more biologically-inspired AI models, possibly leading to breakthroughs in our understanding of both artificial and natural intelligence.

Funding: This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; Ministry of Science and ICT) (No. NRF-2017R1C1B507279).

Data Availability Statement: The manuscript does not involve experimental data; rather, it relies on simulations and analyses conducted using custom Python code developed by the author. The complete source code utilized in this study is accessible through the following dedicated repository: <https://github.com/HyunsuLee/Tuning-W-SF> (accessed on 11 September 2023). We encourage interested readers to explore this repository, as it offers a comprehensive resource for replicating and verifying the analytical procedures employed in this research.

Acknowledgments: The author would like to thank ChatGPT for their assistance in editing and improving the language of the paper, as well as for their helpful brainstorming sessions.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
SR	Successor Representation
SF	Successor Feature
PCA	Principal Component Analysis
MDP	Markov Decision Process
TD	Temporal Difference
MSE	Mean Square Error
PSO	Particle Swarm Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network

Appendix A

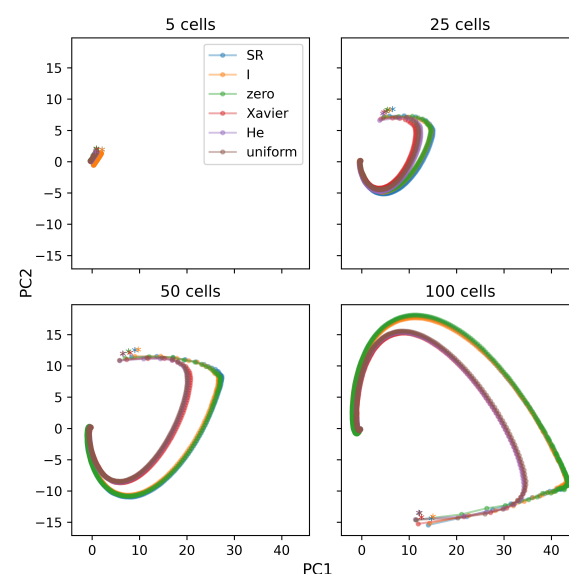


Figure A1. The same principal component analysis (PCA) of the SR place field matrix learning history as shown in Figure 3, but drawn to the same scale. Except for the scale, the details are the same as in Figure 3.

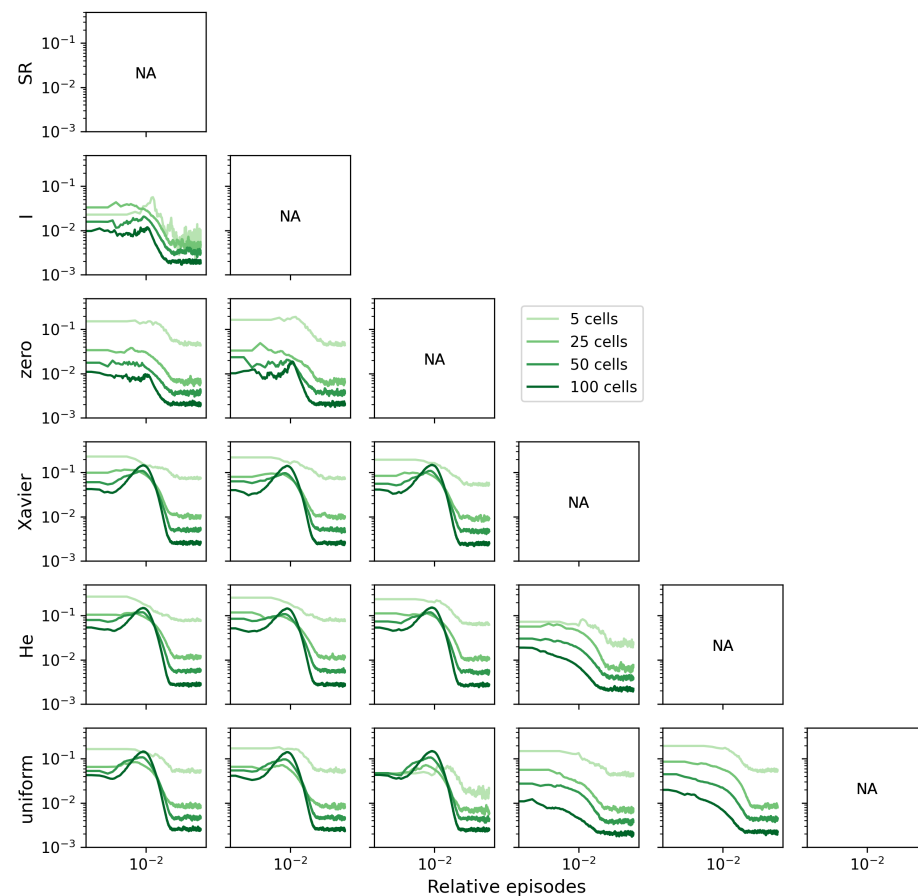


Figure A2. L1 distances divided by the size of the matrix ($N \times N$) are shown. This normalization shows the distance between one element of the SR matrix. The random agents are far from the non-random agents.

References

1. Andersen, P.; Morris, R.; Amaral, D.; Bliss, T.; O'Keefe, J. *The Hippocampus Book (Oxford Neuroscience Series)*; Oxford University Press: Oxford, UK, 2006; p. 872.
2. O'Keefe, J.; Dostrovsky, J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **1971**, *34*, 171–175. [[CrossRef](#)] [[PubMed](#)]
3. O'Keefe, J. Place units in the hippocampus of the freely moving rat. *Exp. Neurol.* **1976**, *51*, 78–109. [[CrossRef](#)] [[PubMed](#)]
4. Stachenfeld, K.L.; Botvinick, M.M.; Gershman, S.J. The hippocampus as a predictive map. *Nat. Neurosci.* **2017**, *7*, 1951. [[CrossRef](#)] [[PubMed](#)]
5. Geerts, J.P.; Chersi, F.; Stachenfeld, K.L.; Burgess, N. A general model of hippocampal and dorsal striatal learning and decision making. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 31427–31437. [[CrossRef](#)]
6. Mehta, M.R.; Quirk, M.C.; Wilson, M.A. Experience-dependent asymmetric shape of hippocampal receptive fields. *Neuron* **2000**, *25*, 707–715. [[CrossRef](#)] [[PubMed](#)]
7. de Cothi, W.; Barry, C. Neurobiological successor features for spatial navigation. *Hippocampus* **2020**, *30*, 1347–1355. [[CrossRef](#)] [[PubMed](#)]
8. George, T.; de Cothi, W.; Stachenfeld, K.; Barry, C. Rapid learning of predictive maps with STDP and theta phase precession. *Elife* **2023**, *12*, e80663. [[CrossRef](#)] [[PubMed](#)]
9. Fang, C.; Aronov, D.; Abbott, L.; Mackevicius, E. Neural learning rules for generating flexible predictions and computing the successor representation. *Elife* **2023**, *12*, e80680. [[CrossRef](#)]
10. Bono, J.; Zannone, S.; Pedrosa, V.; Clopath, C. Learning predictive cognitive maps with spiking neurons during behavior and replays. *Elife* **2023**, *12*, e80671. [[CrossRef](#)]
11. Barreto, A.; Dabney, W.; Munos, R.; Hunt, J.J.; Schaul, T.; Van Hasselt, H.; Silver, D. Successor features for transfer in reinforcement learning. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
12. Puterman, M.L. *Markov Decision Processes*; John Wiley & Sons: Hoboken, NJ, USA, 2014; p. 684.

13. Sutton, R.S.; Barto, A.G. *Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 2018; p. 552.
14. Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Comput.* **1993**, *5*, 613–624. [[CrossRef](#)]
15. Lee, H. Toward the biological model of the hippocampus as the successor representation agent. *Biosystems* **2022**, *213*, 104612. [[CrossRef](#)] [[PubMed](#)]
16. Vertes, E.; Sahani, M. A neurally plausible model learns successor representations in partially observable environments. *arXiv* **2019**, arXiv:1906.09480v1.
17. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv* **2015**, arXiv:1502.01852.
19. Lehnert, L.; Tellex, S.; Littman, M.L. Advantages and Limitations of using Successor Features for Transfer in Reinforcement Learning. *arXiv* **2017**, arXiv:1708.00102v1.
20. Jolliffe, I.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. A Math. Phys. Eng. Sci.* **2016**, *374*, 20150202. [[CrossRef](#)] [[PubMed](#)]
21. Cunningham, J.P.; Yu, B.M. Dimensionality reduction for large-scale neural recordings. *Nat. Neurosci.* **2014**, *17*, 1500–1509. [[CrossRef](#)] [[PubMed](#)]
22. Kulkarni, T.D.; Saeedi, A.; Gautam, S.; Gershman, S.J. Deep Successor Reinforcement Learning. *arXiv* **2016**, arXiv:1606.02396v1.
23. Zhang, J.; Springenberg, J.T.; Boedecker, J.; Burgard, W. Deep Reinforcement Learning with Successor Features for Navigation across Similar Environments. *arXiv* **2016**, arXiv:1612.05533v3.
24. Eberhart; Yuhui, S. Particle swarm optimization: Developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Republic of Korea, 27–30 May 2001; IEEE: Piscataway, NJ, USA, 2001.
25. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1996.
26. Liu, Z.; Liu, S.; Liu, L.; Sun, J.; Peng, X.; Wang, T. Sentiment recognition of online course reviews using multi-swarm optimization-based selected features. *Neurocomputing* **2016**, *185*, 11–20. [[CrossRef](#)]
27. Montague, P.R.; Dayan, P.; Sejnowski, T.J. A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *J. Neurosci.* **1996**, *16*, 1936–1947. [[CrossRef](#)]
28. Schultz, W. Predictive reward signal of dopamine neurons. *J. Neurophysiol.* **1998**, *80*, 1–27. [[CrossRef](#)] [[PubMed](#)]
29. Rao, R.P.; Sejnowski, T.J. Spike-timing-dependent Hebbian plasticity as temporal difference learning. *Neural Comput.* **2001**, *13*, 2221–2237. [[CrossRef](#)] [[PubMed](#)]
30. Rumelhart, D.E. *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*; MIT Press: Cambridge, MA, USA, 1986; Volume 1, p. 547.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
32. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
33. Yamins, D.L.K.; Hong, H.; Cadieu, C.F.; Solomon, E.A.; Seibert, D.; DiCarlo, J.J. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 8619–8624. [[CrossRef](#)] [[PubMed](#)]
34. Bashivan, P.; Kar, K.; DiCarlo, J. Neural population control via deep image synthesis. *Science* **2019**, *364*, 6439. [[CrossRef](#)] [[PubMed](#)]
35. Lillicrap, T.; Santoro, A.; Marris, L.; Akerman, C.; Hinton, G. Backpropagation and the brain. *Nat. Rev. Neurosci.* **2020**, *21*, 335–346. [[CrossRef](#)] [[PubMed](#)]
36. Whittington, J.C.; Bogacz, R. Theories of Error Back-Propagation in the Brain. *Trends Cog. Sci.* **2019**, *23*, 235–250. [[CrossRef](#)]
37. Gershman, S. The Successor Representation: Its Computational Logic and Neural Substrates. *J. Neurosci.* **2018**, *38*, 7193–7200. [[CrossRef](#)]
38. Momennejad, I.; Russek, E.M.; Cheong, J.H.; Botvinick, M.M.; Daw, N.D.; Gershman, S.J. The successor representation in human reinforcement learning. *Nat. Hum. Behav.* **2017**, *1*, 680–692. [[CrossRef](#)]
39. Gottfried, J.; O'Doherty, J.; Dolan, R. Encoding predictive reward value in human amygdala and orbitofrontal cortex. *Science* **2003**, *301*, 1104–1107. [[CrossRef](#)] [[PubMed](#)]
40. Sul, J.; Kim, H.; Huh, N.; Lee, D.; Jung, M. Distinct roles of rodent orbitofrontal and medial prefrontal cortex in decision making. *Neuron* **2010**, *66*, 449–460. [[CrossRef](#)]
41. Shenhav, A.; Botvinick, M.; Cohen, J. The expected value of control: An integrative theory of anterior cingulate cortex function. *Neuron* **2013**, *79*, 217–240. [[CrossRef](#)]
42. Kolling, N.; Wittmann, M.K.; Behrens, T.E.J.; Boorman, E.D.; Mars, R.B.; Rushworth, M.F.S. Value, search, persistence and model updating in anterior cingulate cortex. *Nat. Neurosci.* **2016**, *19*, 1280–1285. [[CrossRef](#)] [[PubMed](#)]

43. Gauthier, J.; Tank, D. A Dedicated Population for Reward Coding in the Hippocampus. *Neuron* **2018**, *99*, 179–193.e7. [[CrossRef](#)] [[PubMed](#)]
44. Meyniel, F.; Sigman, M.; Mainen, Z. Confidence as Bayesian Probability: From Neural Origins to Behavior. *Neuron* **2015**, *88*, 78–92. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.