Thesis for Bachelor's Degree

Understanding Stitching Ability in State Space Model

Jaegyun Im

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2025

학사학위논문

상태 공간 모델의 조합적 일반화 능력 이해

임재균

전기전자컴퓨터공학부

광주과학기술원

2025

Understanding Stitching Ability in State Space Model

Advisor: Sundong Kim

by

Jaegyun Im

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in the Electrical Engineering and Computer Science Concentration

Gwangju, Republic of Korea

December 06, 2024

Approved by

Professor Sundong Kim

Committee Chair

Understanding Stitching Ability in State Space Model

Jaegyun Im

Accepted in partial fulfillment of the requirements for

the degree of Bachelor of Science

December 06, 2024

Committee Chair _____

Prof. Sundong Kim

Committee Member _____

Prof. Kangil Kim

Dedicated to my family.

BS/EC 20195151 Jaegyun Im. Understanding Stitching Ability in State Space Model. School of Electrical Engineering and Computer Science. 2025. 20p. Advisor: Prof. Sundong Kim.

Abstract

State Space Models (SSM) have gained attention for their effectiveness in long-sequence modeling and reinforcement learning (RL), demonstrating potential to enhance agents' stitching ability and model-based planning by accurately capturing environmental dynamics. Experiments in the Frozen Lake environment revealed that SSM successfully generated new optimal trajectories from limited data. However, limitations include insufficient evaluation metrics, lack of comparative experiments, and restricted experimental settings. Future studies will explore more complex environments and systematically evaluate SSM's performance in comparison to traditional RL models.

BS/EC 20195151 임재균. 상태 공간 모델의 조합적 일반화 능력 이해. 전기전자컴퓨터공학부. 2025. 20p. 지도교수: 김선동 교수님.

국 문 요 약

State Space Model(SSM)은 장기 시퀀스 모델링 및 강화학습(RL)에서 주목받고 있으며, 특히 환경의 동적 특성을 학습하여 에이전트의 stitching 능력과 모델 기반 계획 효율성을 향상시킬 수 있는 잠재력을 지닌다. Frozen Lake 환경에서 SSM의 stitching 능력을 평가한 결과, 제한된 데이터에서 새로운 최적 궤적을 형성하는 데 성공했다. 그러나 평가 지표 부족, 비교 실험 부재, 제한된 실험 환경 등 몇 가지 한계가 존재한다. 향후 연구는 보다 복잡한 환경에서의 실험 및 기존 RL 모델과의 비교를 포함하여 SSM의 성능을 보다 체계적으로 검증할 예정이다.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

최근 State Space Sequence(SSM) 기반 모델은 장기 시퀀스 모델링(long sequence modeling) 과제에서 최고 수준의 성능을 달성하며 주목받고 있다. 빠른 추론 속도와 병렬 학습이 가능하다는 장점을 갖추고 있어, 다양한 강화학습(Reinforcement Learning, RL) 환경에서도 적용 가능성이 높다. 실제로, OpenAI Gym 표준 강화학습 벤치마크에서 우수한 성능을 보이며 그 잠재력을 입증했다 [1]. 그러나 아직까지 SSM이 강화학습에서 효과적인 이유를 이론적으로 설명하는 연구는 충분하지 않은 상태다.

이와 관련해 한 가지 유력한 가설은, SSM이 환경의 동적 특성을 더욱 정확하게 학습함으로써 에이전트의 stitching 능력과 모델 기반 계획(model-based planning)의 효율성을 높 수 있다는 것이다. stitching이란 다양한 상태나 시점에서 얻은 부분적인 정보나 부분 정책을 하나로 이어 붙여, 에이전트가 더 나은 결정을 내릴 수 있도록 통합하는 과정을 의미한다 [2].

동적 계획법(dynamic programming)은 가치 함수(value function) 추정을 통해 stitching을 가능하게 하는 것으로 알려져 있다. 기존의 강화학습 접근법에 비해 SSM은 높은 복잡도의 동적 모델링을 수행할 수 있으므로, 보다 정확한 가치 함수 추정을 할 수 있을 것으로 예상된다. 이는 결과적으로 에이전트의 stitching 성능 향상으로 이어질 수 있다.

본 연구는 실험을 통해, SSM의 동적 모델링 능력이 실제로 stitching 및 모델 기반

계획 성능을 높이는지 검증하고자 한다. 이를 통해 SSM의 동적 모델링과 강화학습 성능 간의 관계를 규명하고, 강화학습에서 SSM 활용의 이론적 토대를 마련하고자 한다.

Chapter 2

Preliminaries

2.1 Reinforcement Learning

강화학습(Reinforcement Learning, RL)은 에이전트가 환경과의 상호작용을 통해 최적의 행동 전략(정책, policy)을 학습하는 기계 학습의 한 분야다 [3]. 매 시간 단계마다 에이전트는 상태를 관찰하고, 행동을 취하며, 환경으로부터 보상을 받는다. 이러한 상호작용 과정은 시간의 흐름에 따라 순차적으로 전개되기에, 마르코프 결정 프로세스(Markov Decision Process, MDP)으로 표현된다.

MDP는 $\langle S, A, R, P, \gamma \rangle$로 구성되며, 각각 상태 집합 S, 행동 집합 A, 보상 함수 R, 상태 전이 확률 P, 할인율 γ를 나타낸다.

- 상태 집합 S: 환경의 모든 가능한 상태를 정의한다.
- 행동 집합 A: 에이전트가 각 상태에서 선택할 수 있는 모든 행동을 포함한다.
- 보상 함수 $R(s, a)$: 특정 상태 s에서 행동 a를 수행했을 때 에이전트가 받는 즉각적인 보상을 나타낸다.
- 상태 전이 확률 $P(s'|s, a)$: 현재 상태 s에서 행동 a를 선택했을 때 다음 상태 s'로 전이될 확률 분포를 정의하며, 이는 환경의 동적 특성을 반영한다.
- 할인율 γ: 미래 보상의 중요도를 결정하며, $0 \leq \gamma \leq 1$ 범위 내의 값을 가진다. γ가 1에 가까울수록 미래 보상의 비중이 커지고, 0에 가까울수록 즉각적인 보상에만

집중하게 된다.

MDP를 해결하기 위해 가치 함수(value function)가 사용된다. 상태 가치 함수(state value function) $V^{\pi}(s)$는 특정 정책 π를 따를 때 상태 s에서 시작하여 얻을 수 있는 기대 누적 보상을 나타낸다. 이는 다음과 같이 정의된다.

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s\right], \text{ for all } s \in S, \tag{2.1}$$

반면, 행동 가치 함수(action value function) $Q^{\pi}(s, a)$는 상태 s에서 행동 a를 취하고 이후 정책 π를 따를 때 기대 누적 보상을 나타낸다. 이는 다음과 같이 정의된다.

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a\right], \text{ for all } s \in S, \tag{2.2}$$

벨만 방정식(Bellman Equation)은 가치 함수 간의 관계를 재귀적으로 정의하며, MDP를 해결하기 위한 기초를 제공한다. 상태 가치 함수에 대한 벨만 방정식은 다음과 같이 표현된다.

$$V^{\pi}(s) = \sum_{a} \pi(a|s) \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^{\pi}(s')\right] \tag{2.3}$$

행동 가치 함수에 대한 벨만 방정식은 다음과 같이 표현된다.

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \sum_{a'} \pi(a'|s') Q^{\pi}(s', a') \tag{2.4}$$

최적 상태 가치 함수(optimal state value function) $V^{*}(s)$는 상태 s에서 시작하여 최적

정책 π^*을 따랐을 때 기대되는 최대 누적 보상을 나타낸다. 이는 다음과 같이 표현된다.

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right] \tag{2.5}$$

최적 행동 가치 함수(optimal action value function) $Q^*(s,a)$는 상태 s에서 행동 a를 선택한 후 최적 정책 π^*을 따랐을 때 기대되는 최대 누적 보상을 나타낸다. 이는 다음과 같이 표현된다.

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a') \tag{2.6}$$

최적 정책(optimal policy)은 최적 행동 가치 함수를 기준으로 행동을 선택한다.

$$\pi^*(a|s) = \arg\max_a Q^*(s,a) \tag{2.7}$$

강화 학습에서 에이전트는 이러한 MDP의 수학적 구조를 활용하여 환경의 동적 특성을 학습하고, 누적 보상을 극대화하는 정책을 탐구한다.

2.2 Dynamic Programming

동적 계획법(Dynamic Programming, DP)은 MDP를 해결하기 위한 강력한 방법론 중 하나로, 가치 함수를 반복적으로 갱신하여 최적 정책을 학습하는 데 사용된다. DP는 MDP의 상태 전이 확률 $P(s'|s,a)$과 보상 함수 $R(s,a)$에 대한 완전한 정보를 필요로 하며, 이를 활용해 상태 가치 함수 또는 행동 가치 함수의 최적 값을 계산하고 이를 기반으로 최적 정책을 도출한다. 특히, DP에서 가장 널리 사용되는 알고리즘 중 하나

인 가치 함수 반복(Value Function Iteration)은 벨만 최적 방정식(Bellman Optimality Equation)을 기반으로 상태 가치 함수를 반복적으로 갱신하여 최적 상태 가치 함수에 수렴하도록 한다.

가치 함수 반복은 초기 단계에서 상태 가치 함수 $V(s)$를 초기화하는 것으로 시작된다. 초기값은 보통 모든 상태에 대해 0으로 설정되거나 랜덤 값으로 할당된다. 이후, 벨만 최적 방정식을 활용해 상태 가치 함수를 반복적으로 갱신한다. 벨만 최적 방정식은 특정 상태에서의 가치가 해당 상태에서 선택할 수 있는 모든 행동에 대한 보상의 기대값과 이후 상태의 가치의 합 중 최대값으로 정의된다. 이 수식은 다음과 같다.

$$V_{k+1}(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right].$$

여기서 k는 반복 단계를 나타낸다.

이 반복 과정은 상태 가치 함수 $V_k(s)$가 최적 상태 가치 함수 $V^*(s)$에 수렴할 때까지 계속된다. 수렴 조건은 보통 연속된 반복 간의 값 변화가 미리 정의된 작은 값 ϵ보다 작아질 때 충족된다. 최적 상태 가치 함수에 도달한 후에는, 이를 활용해 최적 정책을 도출할 수 있다. 최적 정책은 상태 가치 함수에서 가장 높은 기대 보상을 제공하는 행동을 선택하는 방식으로 정의되며, 이는 다음과 같은 수식으로 표현된다:

$$\pi^*(a|s) = \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s') \right].$$

2.3 Stitching

Stitching은 상태 공간에서 서로 다른 궤적(trajectory)을 결합하여 새로운 궤적 생성하는 능력을 의미하며, 이는 강화 학습과 같은 분야에서 학습 효율성과 일반화 능력을

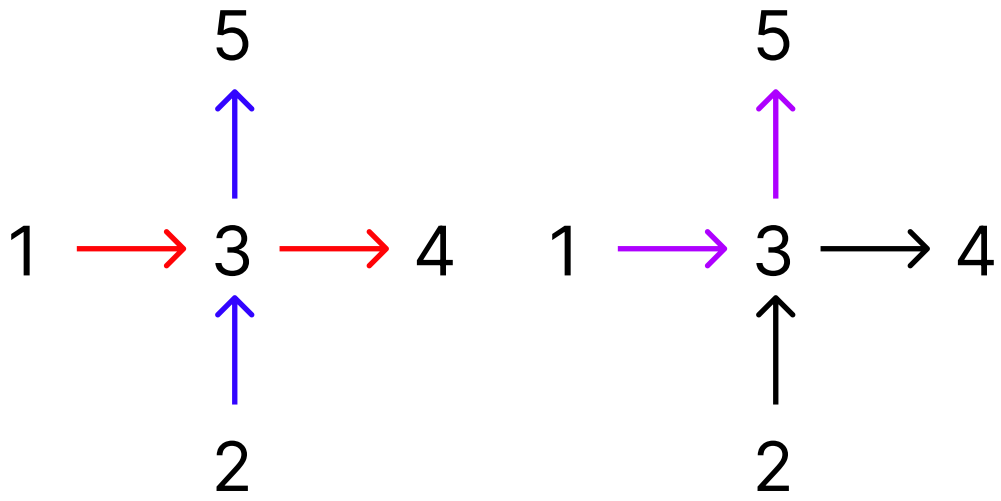

Figure 2.1: Stitching은 부분 궤적을 결합하여 최적 정책을 학습하는 과정으로, 보상을 전파하며 새로운 궤적을 생성하는 능력을 시각화한 예시입니다.

높이는 데 핵심적인 역할을 한다 [2, 4, 5, 6, 7, 8]. 동적 계획법 기반 알고리즘들은 이와 같은 특성을 활용하여 부분적인 궤적을 연결함으로써 최적 정책을 생성할 수 있다.

예를 들어, 5개의 상태로 구성된 MDP에서 상태 1에서 5로 직접 이동하는 데이터가 없더라도, $5 \to 3 \to 1$과 같은 역방향 경로를 통해 보상을 전파하여 올바른 행동을 도출할 수 있다 [8]. 이러한 Stitching의 원리는 벨만 방정식에 의해 작동한다. 벨만 방정식은 각 상태의 가치를 다음 상태들의 가치를 기반으로 재귀적으로 계산하며, 이를 통해 서로 다른 궤적에서 얻은 정보를 자연스럽게 결합한다.

Stitching은 다음 세 가지 측면에서 강화 학습에서 중요한 가치를 가진다. 첫째, 기존의 최적이 아닌(sub-optimal) 데이터를 효과적으로 활용하여 데이터 효율성을 높인다. 둘째, 오프폴리시(Off-policy) 학습을 가능하게 한다. 오프폴리시 학습은 현재 에이전트가 따르고 있는 정책과는 다른 정책에서 수집된 데이터를 활용하는 방법으로, 데이터를 재활용하여 학습 효율을 극대화할 수 있게 한다. 예를 들어, 이전 정책에서 생성된 궤적 데이터나 임의적 행동 데이터를 재활용하여 학습에 활용할 수 있다. 이는 Stitching을 통해 궤적 간의 정보를 결합함으로써 더욱 강력하게 작동한다. 셋째, 학습 중에 나타나지

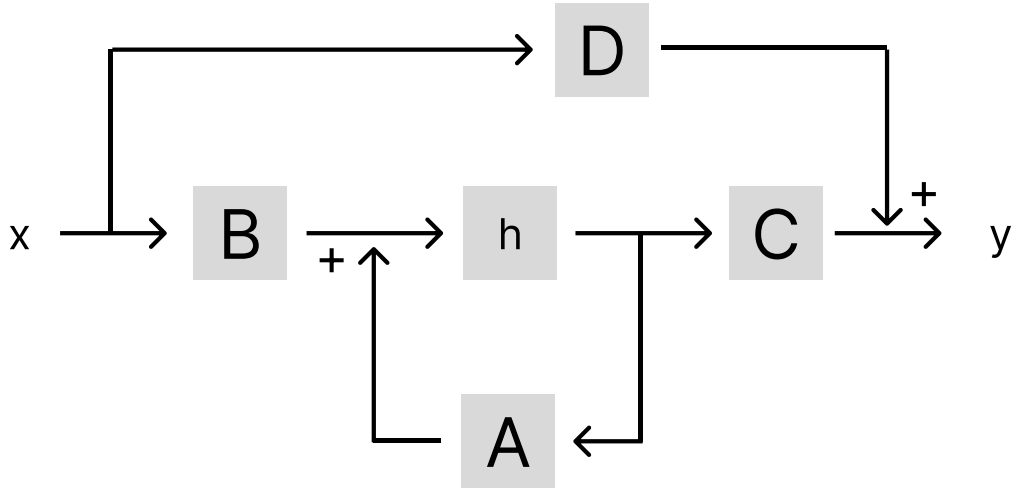

Figure 2.2: 상태 공간 방정식에서 변수 간의 흐름과 상호 작용을 나타내는 블록 다이어그램

않았던 상태-목표(state-goal) 쌍에 대해서도 추론이 가능하게 함으로써 학습의 일반화 능력을 확장한다. 이는 에이전트가 이전에 탐색하지 않은 상태에서도 적절한 행동을 도출할 수 있도록 돕는다 [2, 4, 5, 6, 7, 8].

2.4 State Space Model

State Space Model(SSM)은 전통적으로 제어 이론에서 상태 변수를 통해 동적 시스템을 모델링하는 데 사용되며, 시간이 지남에 따라 변화하는 시스템을 설명하는 수학적 모델의 한 종류다. 동역학을 포착하는 데 특히 효과적인 이 모델은 제어 시스템, 신호 처리, 시계열 분석, 강화 학습 등 다양한 분야에서 활용된다. SSM은 다음과 같은 두 방정식으로 구성된다.

$$\mathbf{h}'(\mathbf{t}) = \mathbf{A}\mathbf{h}(\mathbf{t}) + \mathbf{B}\mathbf{x}(\mathbf{t}) \tag{2.8}$$

$$\mathbf{y}(\mathbf{t}) = \mathbf{C}\mathbf{h}(\mathbf{t}) + \mathbf{D}\mathbf{x}(\mathbf{t}) \tag{2.9}$$

- h_k : 상태 변수(state vector), 시스템 내부의 상태를 나타내는 벡터로 시스템이

동적으로 어떻게 변화하는 지를 나타낸다.

- x_k : 입력 변수(input vector), 외부에서 시스템으로 주어지는 입력을 나타낸다.
- y_k : 출력 변수(output vector), 시스템에서 측정되는 출력을 나타낸다.
- A : 상태 행렬(state matrix), 상태가 다음 시점으로 전이(transition)할 때, 이전 상태에 어떤 영향을 받는지를 나타내는 행렬이다.
- B : 입력 행렬(input matrix), 시스템의 상태가 다음 시점으로 전이할 때, 입력 벡터가 어떻게 작용하는지를 나타내는 행렬이다.
- C : 출력 행렬 (output matrix), 상태 벡터가 출력으로 어떻게 매핑되는지를 나타내는 행렬이다.
- D : 피드포워드 행렬 (feedforward matrix), 입력 벡터가 출력에 직접적으로 미치는 영향을 나타내는 행렬이다.

이는 이산화를 통해 아래와 같이 표현할 수 있다.

$$\mathbf{h_k} = \overline{\mathbf{A}}\mathbf{h_{k-1}} + \overline{\mathbf{B}}\mathbf{x_k} \tag{2.10}$$

$$\mathbf{y_k} = \overline{\mathbf{C}}\mathbf{h_k} + \overline{\mathbf{D}}\mathbf{x_k} \tag{2.11}$$

S4(Structured State Space Model)는 SSM(State Space Model)을 사용하여 긴 시퀀스를 모델링하는 방법과 이를 딥러닝과 결합했을 때 안정성, 성능, 학습 속도를 향상시키는 다양한 기술을 제안했다 [9]. S4 모델은 HiPPO라고 불리는 특별한 행렬 초기화를 사용하여 시퀀스의 히스토리를 더 잘 보존한다 [10].

S4 모델의 주요 강점 중 하나는 이를 순환(Recurrent) 모델과 컨볼루션(Convolutional) 모델로 변환할 수 있다는 점이다. 순환 모델로 변환하면 추론 시 빠르고 메모리 효율적인 계산이 가능하며, 컨볼루션 모델로 변환하면 시간 단계(Timestep)를 병렬로 처리하여 학습 효율을 높일 수 있다.

최근에는 S4를 단순화한 모델인 S5를 제안되었는데, S5의 주요 기여 중 하나는 컨볼루션 대신 병렬 스캔(Parallel Scan)을 사용하는 것이다. 이 방법은 S4의 복잡성을 크게 단순화하고 더 유연한 수정이 가능하게 한다 [11].

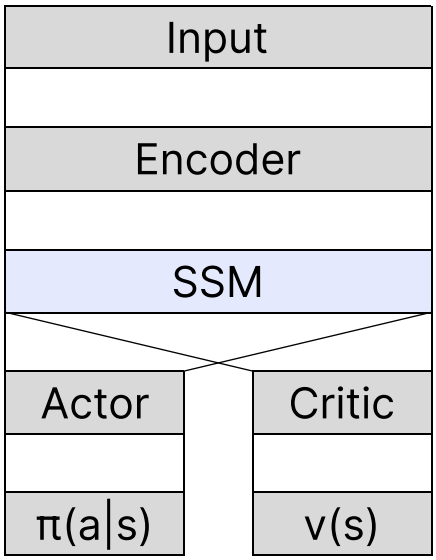

Figure 2.3: 모델 구조에 대한 블록다이어그램

S5의 상태 변수(hidden state)를 훈련 단계에서 궤적(trajectory) 내에서 초기화할 수 있도록 수정하는 방안이 제안되어 기존 프레임워크에서 RNN을 S5 레이어로 간단히 대체할 수 있었다 [1]. 액터-크리틱(actor-critic) 기준 알고리즘을 사용하여 실험을 진행했으며, LSTM(Long Short-Term Memory)을 S5 블록으로 교체했다 [12].

Actor는 현재 상태에서 어떤 행동을 취해야 하는지 결정하며, Critic은 해당 행동이 얼마나 적절한지 평가한다. 이 구조는 상태 공간 모델이 학습한 정보를 강화 학습의

의사결정 과정에 통합하는 역할을 한다.

Chapter 3

Method

3.1 Frozen Lake

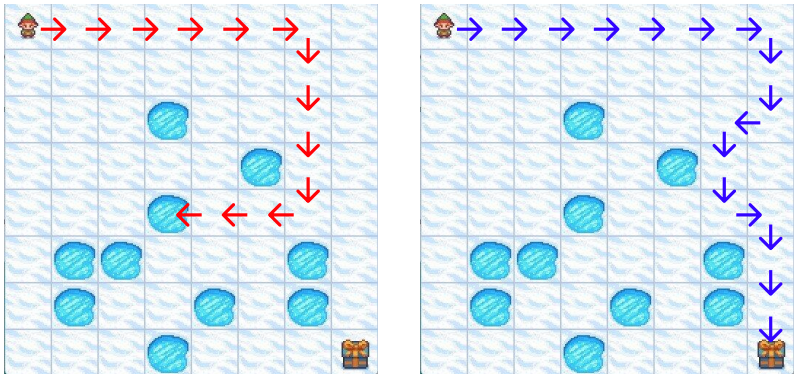

Figure 3.1: 학습 과정에서 제공된 두 개의 궤적를 나타낸 그림이다. 왼쪽 그림는 구멍에 빠지는 궤적를 나타내며, 오른쪽 그림은 목표 지점에 도달하는 경로를 보여주지만, 최적 궤적는 아니다.

본 실험은 Frozen Lake 환경에서 에이전트가 제한된 학습 데이터를 바탕으로 목표 지점까지의 최적 경로를 stitching을 통해 형성할 수 있는지를 평가하는 데 초점을 맞추고 있다. Frozen Lake는 8×8 크기의 격자로 구성되어 있으며, 각 타일은 에이전트가 이동할 수 있는 얼음 타일(Frozen, F), 목표 지점(Goal, G), 그리고 실패로 간주되는 구멍(Hole, H)으로 구성된다. 에이전트는 시작 (Start, S)지점에서 시작해 구멍에 빠지지 않고 목표 지점에 도달하는 최적의 경로를 학습해야 한다.

보상 구조는 에이전트가 목표 지점에 도달하면 보상 1.0을 받으며, 구멍에 빠질 경우 보상 0과 함께 에피소드가 종료된다. 그 외의 일반적인 이동에는 추가적인 보상이 주어지지 않는다.

Frozen Lake의 상태 공간은 8×8 격자의 각 타일로 이루어져 총 64개의 상태로 구성되며, 행동 공간은 상(Up), 하(Down), 좌(Left), 우(Right)의 네 가지 방향으로 구성된다. 에이전트는 이 행동을 통해 상태 전환을 수행하며, 목표 지점이나 구멍에 도달할 때까지 탐색을 계속한다. 에피소드는 에이전트가 목표 지점에 도달하거나 구멍에 빠졌을 때 종료된다.

학습 단계에서 두 개의 궤적이 제공된다. 첫 번째 궤적은 빨간색 화살표로 표시된 경로를 따르며, 이 경로를 따라가다 보면 결국 구멍에 빠지게 된다. 따라서, 이 궤적에서는 모든 단계에서 가치 함수 값이 0으로 설정된다. 두 번째 궤적은 파란색 화살표로 표시된 경로를 따른다. 이 경로를 통해 목표 지점에 도달할 수는 있지만, 해당 경로는 최적의 궤적은 아니다. 이 궤적에서는 가치 함수 값을 목표 지점에서 시작하여, γ=0.9를 적용해 계산된 결과를 가치함수의 값으로 사용했다.

이후, 에이전트가 학습 데이터에 포함되지 않은 새로운 궤적(최적의 궤적)를 stitching 능력을 통해 생성할 수 있는지를 평가했다.

Chapter 4

Result

4.1 Result

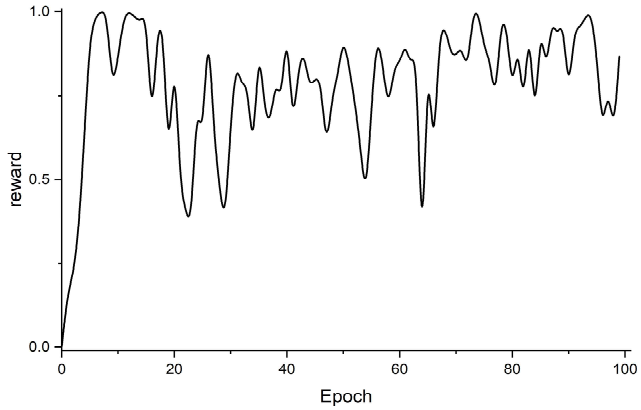

Figure 4.1: 각 에포크당 얻은 평균 보상에 대한 그래프이다. 100 에포크동안 측정했으며, 1 에포크는 동일한 8개의 환경, 128step으로 이루어져있다.

본 실험에서는 8개의 동일한 환경에서 에이전트의 성능을 평가했으며, 각 에포크는 128 스텝으로 구성되었다. 보상은 에포크 내 평균 보상을 기준으로 계산하여, 100 에포크동안 모델 성능을 측정하였다.

실제 에이전트가 수행한 행동들을 분석한 결과, 에이전트는 최적의 궤적을 찾아 이를 활용하는 데 성공했다. 이는 에이전트가 학습 데이터를 바탕으로 최적의 궤적를 찾아내고 활용할 수 있음을 보여준다.

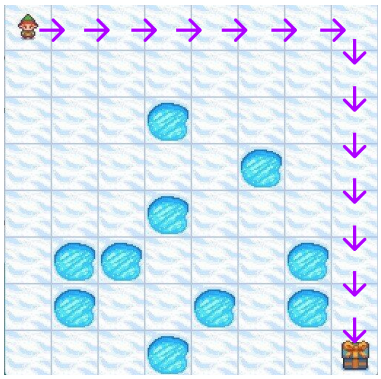

Figure 4.2: 해당 그림들은 평가 과정에서 에이전트가 인식하고 따라간 전문가 궤적들을 보여준다.

Chapter 5

Limitation and Future works

5.1 Limitation

본 연구에서는 학습에 사용된 궤적이 아닌 새로운 궤적을 통해 에이전트가 목표 지점에 성공적으로 도달하는 모습을 확인하였다. 이는 SSM(State Space Model)이 stitching 능력, 즉 기존에 학습된 경로를 조합하여 새로운 궤적를 형성하는 능력을 갖추고 있음을 보여준다. 그러나 이러한 결과만으로 SSM의 stitching 능력이 뛰어나다고 단언하기에는 몇 가지 한계가 존재한다.

첫째, SSM의 stitching 능력을 정량적으로 평가하기 위해 사용된 지표는 각 에포크에서 얻어진 보상의 평균 계산에 그쳤다. 하지만 이는 생성된 궤적이 최적 궤적에 얼마나 근접했는지, 혹은 학습된 하위 궤적을 얼마나 효과적으로 결합했는지 구체적으로 측정하기에 충분하지 않다. 따라서 SSM의 stitching 능력을 보다 명확히 평가하기 위해 추가적인 지표와 분석이 필요하다.

둘째, 연구의 범위가 제한적이었다. stitching 능력을 평가하는 방법 중 하나로, 목표 지점에 도달하는 궤적이 포함되지 않은 데이터만 제공한 상태에서 SSM이 학습된 하위 궤적을 조합하여 목표 지점까지 도달하는 능력을 검증하는 실험이 이루어지지 않았다.

셋째, SSM의 stitching 능력을 평가하기 위해 다른 모델과의 비교 실험이 이루어지지 않았다. 기존 강화 학습 모델과 SSM의 성능을 비교함으로써 SSM이 환경의 구조적 특성을 얼마나 효율적으로 학습하고 활용하는지, 그리고 제한된 데이터로 새로운 궤적을

생성하는 데 있어 얼마나 뛰어난지를 입증할 필요가 있다.

결론적으로, 본 연구는 SSM이 제한된 궤적 데이터를 활용하여 새로운 궤적을 형성할 수 있는 stitching ability를 갖추고 있음을 확인한 중요한 결과를 제시한다. 그러나 stitching 능력을 더욱 깊이 이해하고 평가하기 위해서는 정량적 평가, 다양한 실험 세팅, 및 모델 간의 비교를 포함한 추가적인 연구가 필요하다.

5.2 Future works

본 연구에서는 상태 공간 모델의 조합적 일반화 능력을 분석하기 위해 Frozen Lake 환경에서 실험을 수행했다. 그러나, 제한된 환경과 실험 설정으로 인해 모델의 일반화 가능성을 다양한 상황에서 충분히 검증하지 못한 한계가 있다. 향후 연구에서는 보다 복잡한 환경인 D4RL의 Maze2D 에서 상태 공간 모델의 성능을 평가하고, 기존 강화학습 모델과 비교 분석을 진행할 계획이다. 이러한 확장은 본 연구의 결과를 일반화하는 데 기여할 것으로 기대된다.

Summary

Understanding Stitching Ability in State Space Model

stitching 능력을 활용하여 새로운 최적 경로를 형성하는 데 효과적임을 검증하고자 했다. Frozen Lake 환경에서 진행된 실험에서는 제한된 데이터(두 가지 궤적)만 제공된 상황에서도 SSM이 학습 데이터를 조합하여 최적의 궤적을 생성할 수 있음을 확인했다. 이는 SSM이 기존 강화학습 모델 대비 더 높은 수준의 동적 모델링과 가치 함수 추정을 통해 에이전트의 계획 능력을 향상시킬 가능성을 보여준다.

그러나 연구에는 몇 가지 한계가 존재했다. 첫째, stitching 능력을 정량적으로 평가할 지표가 평균 보상 계산에 그쳐 생성된 궤적의 최적성이나 하위 궤적의 효과적 결합 수준을 구체적으로 측정하기 어려웠다. 둘째, 실험 범위가 제한적이어서 목표 지점에 도달하는 궤적을 포함하지 않은 데이터를 활용한 stitching 성능을 충분히 검증하지 못했다. 셋째, 기존 강화학습 모델과의 비교 실험이 이루어지지 않아 SSM의 우수성을 명확히 입증하는 데 한계가 있었다.

향후 연구에서는 보다 복잡한 환경, 예를 들어 Maze2D(D4RL)에서 SSM의 성능을 분석하고, 기존 강화학습 모델과 비교 실험을 통해 제한된 데이터에서 새로운 궤적을 생성하는 능력을 정밀하게 평가할 계획이다. 이를 통해 SSM의 stitching 능력과 환경의 구조적 특성을 학습 및 활용하는 능력을 보다 체계적으로 검증하고, SSM의 이론적 토대를 강화할 수 있을 것으로 기대된다.

References

1. C. Lu, Y. Schroecker, A. Gu, E. Parisotto, J. Foerster, S. Singh, and F. Behbahani, "Structured state space models for in-context reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

2. C. A. Hepburn and G. Montana, "Model-based trajectory stitching for improved offline reinforcement learning," *arXiv preprint arXiv:2211.11603*, 2022.

3. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

4. I. Char, V. Mehta, A. Villaflor, J. M. Dolan, and J. Schneider, "Bats: Best action trajectory stitching," *arXiv preprint arXiv:2204.12026*, 2022.

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

6. X. Lei, X. Zhang, Z. Zhuang, and D. Wang, "Q-wsl: Optimizing goal-conditioned rl with weighted supervised learning via dynamic programming," *arXiv preprint arXiv:2410.06648*, 2024.

7. Z. Zhou, C. Zhu, R. Zhou, Q. Cui, A. Gupta, and S. S. Du, "Free from bellman completeness: Trajectory stitching via model-based return-conditioned supervised learning," *arXiv preprint arXiv:2310.19308*, 2023.

8. R. Ghugare, M. Geist, G. Berseth, and B. Eysenbach, "Closing the gap between td

learning and supervised learning–a generalisation point of view," *arXiv preprint arXiv:2401.11237*, 2024.

9. A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

10. A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, vol. 33, pp. 1474–1487, 2020.

11. J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," *arXiv preprint arXiv:2208.04933*, 2022.

12. I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvari, S. Singh, *et al.*, "Behaviour suite for reinforcement learning," *arXiv preprint arXiv:1908.03568*, 2019.