外部記憶を用いた部分観測環境における教師なし強化学習

中本 光彦[1,a)]　鶴岡 慶雅[2,b)]

概要：部分観測環境における深層強化学習の適用は困難である．また，複雑なタスクにおいては適切な報酬関数を設計することも難しいとされている．本研究では，これらの課題を解決するために，部分観測環境における教師なし強化学習のアルゴリズムを提案する．部分観測性に対処するためにエージェントに外部の記憶機構を与え，外部報酬を用いる代わりに相互情報量に基づいた内発的報酬を提案する．提案する内発的報酬は，エージェントに観測情報が非常に限られている状態空間を優先的に探索しながら，有効な記憶を学習させることを可能にする．実験では，HalfCheetah エージェントに限られた観測だけで，外部報酬を一切使用せずに，前後に走ることを習得させることができた．

Unsupervised Reinforcement Learning for Partially Observable Environments Using External Memories

Mitsuhiko Nakamoto[1,a)]　Yoshimasa Tsuruoka[2,b)]

Abstract: Deep reinforcement learning (RL) is difficult when the environment is partially observable and has no reward function. In this paper, we propose an unsupervised RL algorithm to tackle these problems. We provide the agent with external memory to deal with partial observability, and propose a novel mutual information-based intrinsic reward for unsupervised exploration. The proposed intrinsic reward encourages the agent to explore the state space with strict partial observability, and at the same time, obtain an informative memory. In the experiments, our algorithm enables a HalfCheetah agent to run forward and backward with limited observations and without receiving any external rewards.

1. はじめに

近年，深層強化学習は複雑な意思決定問題を解く手法として盛んに研究されており，ビデオゲームや囲碁などにおいてコンピュータが人間を超える性能を発揮するようになった [1], [2]．一方，より一般的な環境に深層強化学習アルゴリズムを応用する上で，多くの重要な課題が浮上している [3], [4]．その中でも，部分観測環境や無報酬な環境における方策の学習は非常に重要な課題でありながら，まだ未解決な問題として知られている．

[1] 東京大学工学部電子情報工学科
Department of Information and Communication Engineering, The University of Tokyo
[2] 東京大学大学院情報理工学系研究科電子情報学専攻
Graduate School of Information Science and Technology, The University of Tokyo
[a)] mitsuhiko@logos.t.u-tokyo.ac.jp
[b)] tsuruoka@logos.t.u-tokyo.ac.jp

強化学習の問題設定では，エージェントは環境と相互作用しながら方策を学習する．その際，環境の次の状態は現在の状態と行動のみに依存するようなマルコフ決定過程を仮定している．しかし多くの環境では，完全な状態は観測できず，部分的な観測しか得られない．そのような部分観測環境では上記仮定が満たされず，方策の学習が困難である [3]．例えばロールプレイングゲーム（RPG）や一人称視点シューティングゲーム（FPS）は代表的な部分観測環境である．この問題を解決するための一つの手法として，エージェントに記憶を持たせる手法が挙げられる [5], [6]．また，強化学習では一般的にタスク固有の報酬関数をうまく設計する必要がある．しかし，報酬関数の設計には多大な労力が必要であり，複雑なタスクにおいては，適切な報酬関数の設計は非常に困難である [3], [4]．そこで近年，外部からの報酬を必要とせずに方策を学習する「教師なし強化学習」が多く研究されている [7], [8], [9], [10]．

上記 2 つの問題が合わさると，強化学習の応用はさらに難しくなる．本研究では，そのような部分観測かつ無報酬な環境において，有効に方策を学習できるような教師なし強化学習のアルゴリズムを提案する．部分観測性に対処するためにエージェントに外部の記憶機構を与え，さらに外部報酬を用いる代わりに相互情報量に基づいた内発的報酬を提案する．提案する内発的報酬は，エージェントに観測情報が非常に限られている状態空間を優先的に探索しながら，有効な記憶を学習させることを可能にする．実験では，提案手法は HalfCheetah エージェントに限られた観測だけで，外部報酬を一切使用せずに，前後に走ることを習得させることができた．

2. 関連研究

2.1 部分観測環境における強化学習

部分観測性に対する対処法として，主に 3 つのアプローチが提案されている．1 つは一定期間の過去の観測を保存して最新の観測とともに方策の入力として用いる手法である．Deep Q-Network (DQN) を用いて Atari 社のゲームをプレイする方策を学習した先行研究 [1] では，ゲーム画面を画像として観測し，直近 4 フレームのゲーム画面を行動選択時に方策に入力している．2 つ目のアプローチは再帰型ニューラルネットワーク (RNN) や長短期記憶 (LSTM) を用いて，エージェントに過去の隠れ状態を保持させる手法である．Hausknecht らの手法 [5] では，DQN のネットワークに LSTM を組み込んだ Deep Recurrent Q-Network (DRQN) を提案し，Atari 社のゲームにおいて，単一画像のみの入力で先述の研究 [1] と同等なパフォーマンスを達成した．3 つ目のアプローチとしては，エージェントに外部の記憶機構と，それを制御できる行動を持たせる手法である．Icarte ら [6] はこのような手法で前述の 2 つのアプローチより高いパフォーマンスを達成している．また，このアプローチは，RNN や LSTM ベースの手法と比べて計算コストが低く，データのサンプル効率が良い．そのため，本研究でも外部の記憶機構を用いる．

2.2 相互情報量に基づく内発的報酬

外部からの報酬を用いずに方策を学習する教師なし強化学習では，内発的報酬をどのように設計するかが重要であり，相互情報量を用いた手法が数多く提案されている．DIAYN [8] は観測状態 s とスキル z の相互情報量 $\mathcal{I}(s;z)$ を報酬として用いて様々なスキルを習得している．DISCERN [9] は観測状態 s と目標状態 g の相互情報量 $\mathcal{I}(s;g)$ を報酬として用い，与えられた目標状態に到達することに成功している．DADS [10] は，環境のダイナミクスを考慮した報酬 $\mathcal{I}(s_{t+1};z \mid s_t)$ を用いてスキルを学習している．本研究でも，このような相互情報量を用いた内発的報酬を考案する．

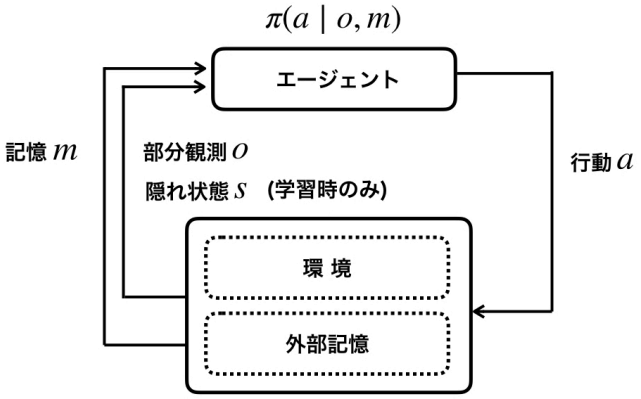

図 1: 本研究の想定する環境の枠組み.

3. 提案手法

3.1 想定する環境

多くの環境では，シミュレータによる方策の学習時は全ての状態を観測できる一方，実環境で運用する際は一部の観測しか得られず，学習時に得られた方策が上手く適用できない．本稿でもそのような設定に基づいて，図 1 のような環境を想定する．エージェントは，学習時には隠れ状態 s を得ることができるが，検証時は，部分観測 o しか得られない．また，エージェントは外部の記憶機構 m を持ち，行動 a によって環境に作用するだけでなく，記憶 m_{old} を m_{new} に書き換えることができる．このような環境において，有効な方策 $\pi(a \mid o, m)$ を学習することを目的とする．ここで，記憶機構 m は様々な構造が考えられるが，本稿では K ビットのバッファを用いる．記憶を編集する際の具体的な流れは付録 A.1 に示す．

3.2 内発的報酬

上記の環境で効率的に方策を学習するために，方策の学習時に，部分観測 o により条件づけられた隠れ状態 s と記憶 m 間の相互情報量 $\mathcal{I}(s;m \mid o)$ を最大化することを提案する．ここで o, s, m は同一時刻のものである．これは式 1 のように 2 つのエントロピーの項に分解でき，同相互情報量を最大化することは，$\mathcal{H}(s \mid o)$ を最大化，つまり部分観測 o から隠れ状態 s が予測し難い状態空間を探索しつつ，$\mathcal{H}(s \mid o, m)$ を最小化，つまり隠れ状態を予測するために有効な記憶 m を獲得するように学習される．

$$\mathcal{I}(s;m \mid o) = \mathcal{H}(s \mid o) - \mathcal{H}(s \mid o, m) \tag{1}$$

続いて，方策 $\pi(a \mid o, m)$ の学習方法を述べる．エントロピーの定義より，式 1 は次のように変形できる．

$$\mathcal{I}(s;m \mid o) = \iiint p(m, o, s) \log \frac{p(s \mid o, m)}{p(s \mid o)} ds\, do\, dm \tag{2}$$

しかし，分布 $p(s \mid o, m)$ は計算できない．そこで，

表 1: 部分観測 o として扱うパラメータ

部位	rootz	rooty	bthigh	bshin	bfoot	fthigh	fshin	ffoot
パラメータ	位置	角度	角度	角度	角度	角度	角度	角度

表 2: 隠れ状態 s として扱うパラメータ

部位	rootx	rootz	rooty	bthigh	bshin	bfoot	fthigh	fshin	ffoot
パラメータ	速度	速度	角速度	角速度	角速度	角速度	角速度	角速度	角速度

$p(s \mid o, m)$ を $q_\phi(s \mid o, m)$ を用いて変分近似することにより，式 3 のような下界を導くことができる．

$$\begin{aligned} \mathcal{I}(s; m \mid o) &= \mathbb{E}_{m,o,s\sim p}\left[\log \frac{p(s \mid o, m)}{p(s \mid o)}\right] \\ &\geq \mathbb{E}_{m,o,s\sim p}\left[\log \frac{q_\phi(s \mid o, m)}{p(s \mid o)}\right] \end{aligned} \tag{3}$$

従って，$\mathcal{I}(s; m \mid o)$ を最大化する代わりに，上記の下界を最大化することを目標とし，それは以下の 2 ステップを繰り返すことにより実現できる．

ステップ (1): 分布 q_ϕ のパラメータ ϕ を最尤推定で更新.
ステップ (2): 更新された q_ϕ を用いて式 3 の下界を最大化するよう方策 π を更新.

ステップ (1) は，現在の方策をロールアウトして (s, m, o) をサンプルして最尤推定すればよい．ステップ (2) は，$\log q_\phi(s \mid o, m) - \log p(s \mid o)$ を報酬として強化学習すれば良いが，$\log p(s \mid o)$ は計算不可能であるため，先行研究 [10] に倣い式 4 のように近似することにより，報酬関数を式 5 のように導くことができる．ただし，L は記憶の事前分布 $p(m)$ からサンプルする m_i の数である．

$$\begin{aligned} p(s \mid o) &= \int p(s \mid o, m)\, p(m \mid o) dm \\ &\approx \int q_\phi(s \mid o, m)\, p(m) dm \\ &\approx \frac{1}{L}\sum_{i=1}^{L} q_\phi(s \mid o, m_i), \quad m_i \sim p(m) \end{aligned} \tag{4}$$

$$r_m(o, a, s) = \log \frac{q_\phi(s \mid o, m)}{\sum_{i=1}^{L} q_\phi(s \mid o, m_i)} + \log L \tag{5}$$

全体のアルゴリズムを Algorithm 1 に示す．このような手続きを用いれば，相互情報量 $\mathcal{I}(s; m \mid o)$ を最大化するような方策を学習でき，得られた方策は部分観測環境下において有効な記憶を獲得し，部分観測性に対してロバストになると期待される．

4. 実験

4.1 実験設定

提案手法を検証するため，OpenAI Gym [11] の

Algorithm 1 提案手法のアルゴリズム

Initialize: π, q_ϕ
  while not converged **do**
    記憶 m を初期化;
    方策をロールアウトしてデータをサンプル;
    q_ϕ を最尤推定で更新;
    各状態遷移の $r_m(o, a, s)$ を計算;
    方策 π を更新;
  end while

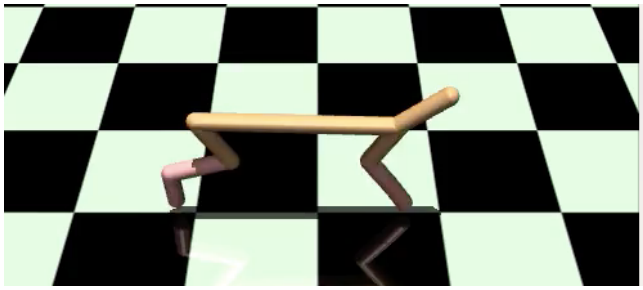

図 2: HalfCheetah の二足歩行エージェント

HalfCheetah-v1 という環境で実験を行なった．このタスクでは，図 2 のような二足歩行エージェントを走らせるように制御する．通常では，走った距離と加えたトルクの大きさに基づいた外部報酬が与えられるが，本実験の学習ではその外部報酬を用いない．また，通常では表 1 と表 2 に示されている全てのパラメータを観測でき，これらを観測情報として方策の入力とする．一方，本実験では，表 1 に示されている位置に関するパラメータを部分観測 o として方策の入力として，表 2 に示されている速度に関するパラメータは隠れ状態 s として，シミュレータでの学習時にのみ与えられるものとした．ただし，部分観測性を増やすため，*rootx* の位置は o と s のいずれにも使用していない．また，外部記憶の構造としては K ビットのバッファを用いた．実装の詳細は付録 A.2 に示す．

4.2 実験結果

K ビットの外部記憶の有効性を比較検討するために，本実験では K=12 と K=1 で実験を行った．学習時の内発的報酬の推移を図 3 に示す．図を見ると，12 ビットの外部記憶を用いた場合は，学習を進めるにつれて内発的報酬が増加しているのに対して，1 ビットの外部記憶を用いた場合は内発的報酬が小さい値にとどまっている．また，実際に

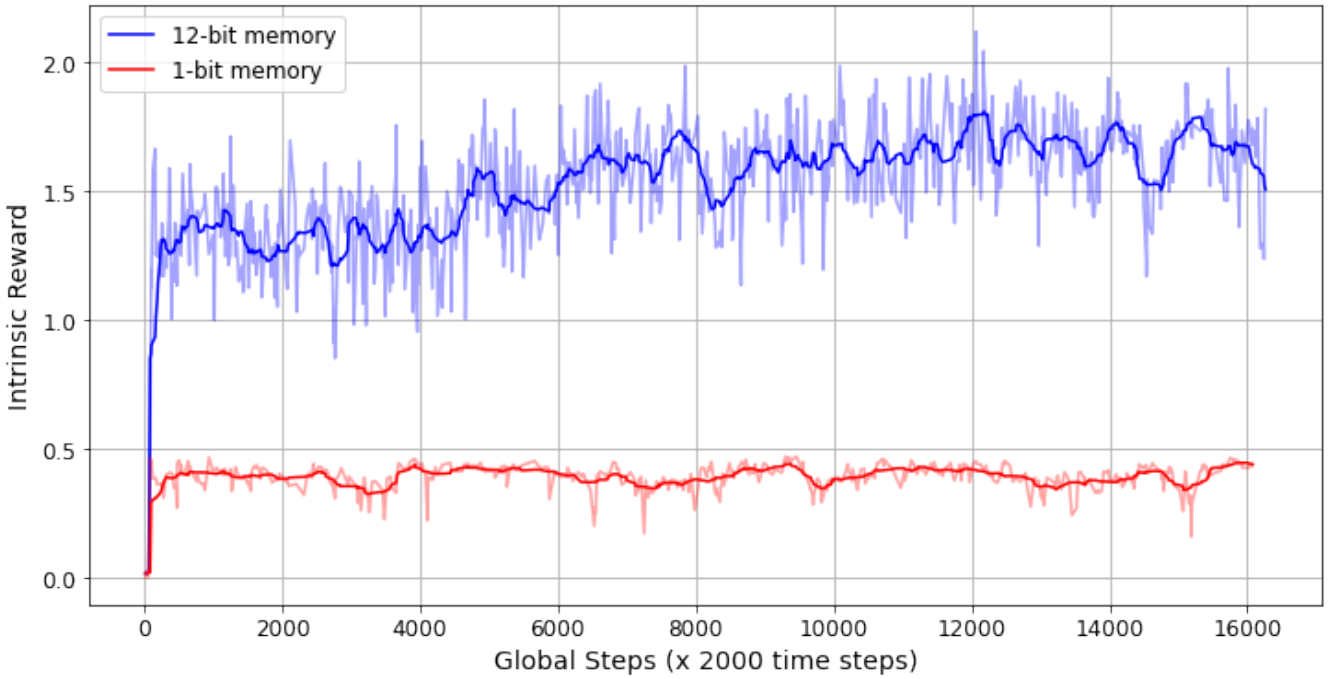

図 3: 学習時に獲得した内発的報酬の推移．図中の薄い線が 1 回の学習ループ（=2,000 タイムステップ）で得た 1 タイムステップあたりの内発的報酬であり，濃い線はその移動平均を表す．

得られた方策を用いて，エージェントを制御した際の動きのフレームを付録 A.3 に示す．12 ビットの外部記憶を用いた場合は，エージェントは前方や後方に向かって上手く走っているのに対して，1 ビットの外部記憶を用いた場合は，エージェントは走らずに画面中心にとどまっている．

4.2.1 考察

上記結果により，1 ビットの記憶は部分観測性を補うためには表現力が不足しており，上手く学習できていないのに対して，12 ビットの外部記憶は部分観測性を十分に補えるような記憶を学習できていると考えられる．注目すべきは，12 ビットの外部記憶を用いた場合，走ることを明示的に教える外部報酬を用いずに，本研究で提案した相互情報量を用いた内発的報酬のみで，エージェントに走ることを習得させている．これは，提案手法の有効性を示す結果となった．

5. おわりに

本稿では，部分観測環境における教師なし強化学習のアルゴリズムを提案した．部分観測性を補うために，エージェントに外部の記憶機構とそれを制御する行動を与えた．また，教師なしで方策を学習するために，相互情報量に基づいた内発的報酬を提案した．提案した内発的報酬は，エージェントに観測情報が非常に限られている状態空間を優先的に探索しながら，有効な記憶を学習させることを可能にする．提案手法を用いた実験では，HalfCheetah エージェントに限られた観測だけで，外部報酬を一切使用せずに，前後に走ることを習得させることができた．今後の展望として，より多くの部分観測の設定や記憶の構造を用いて提案手法の有効性を確認したい．また，得られた方策は，部分観測環境における下流タスクを学習するための初期値として有効であるかも検証したい．

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, Vol. 518, No. 7540, pp. 529–533, 2015.

[2] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, Vol. 529, No. 7587, pp. 484–489, 2016.

[3] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, pp. 1–50, 2021.

[4] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, Vol. 40, No. 4-5, pp. 698–721, 2021.

[5] Matthew Hausknecht and Peter Stone. Deep Recurrent Q-Learning for Partially Observable MDPs. In *2015 AAAI Fall Symposium Series*, 2015.

[6] Rodrigo Toro Icarte, Richard Valenzano, Toryn Q. Klassen, Phillip Christoffersen, Amir massoud Farahmand, and Sheila A. McIlraith. The act of remembering: a study in partially observable reinforcement learning. *arXiv*, 2020.

[7] Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational Empowerment as Representation Learning for Goal-Conditioned reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139 of *Proceedings of Machine Learning Research*, pp. 1953–1963, 2021.

[8] Benjamin Eysenbach, Julian Ibarz, Abhishek Gupta, and Sergey Levine. Diversity is all you need. *International Conference on Learning Representations (ICLR)*, 2019.

[9] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *International Conference on Learning Representations (ICLR)*, 2019.

[10] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *International Conference on Learning Representations (ICLR)*, 2020.

[11] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv*, 2016.

[12] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic Algorithms and Applications. *arXiv*, 2018.

付　　録

A.1　外部記憶を編集する際の流れ

提案手法では，エージェントに外部の記憶機構とそれを制御する行動を与え，図 1 のような環境を想定した．方策 $\pi(a \mid o, m)$ は部分観測 o と記憶 m を入力として，行動 a を出力とする．行動 a は，従来の環境に作用する行動 a_e と，記憶を編集するための行動 a_m に分けて $a = \langle a_e, a_m \rangle$ と表すことができる．外部記憶 m が K ビットであれば，それを編集する行動 a_m も K ビットであり，記憶の編集は下式のように，m と a_m の排他的論理和をとることにより行う．

$$m_{new} = m_{old} \oplus a_m$$

例えば，あるタイムステップにおいて，$m_{old} = 010$，$a_m = 110$ であれば，$m_{new} = 010 \oplus 110 = 100$ となる．

A.2　実装の詳細

本研究では，Soft Actor-Critic (SAC) [12] を用いて off-policy の学習を行った．エージェントは 1 回の学習ループ（Global Step）において，2,000 タイムステップのサンプルを集めて Replay Buffer にデータを追加する．Replay Buffer のサイズは 10,000 とした．次に，Replay Buffer 内のデータを用いて，$q_\phi(s \mid o, m)$ を 16 エポック繰り返し更新する．その後，更新された q_ϕ を用いて，式 5 に従い内発的報酬の計算を行うが，その際に記憶の事前分布 $p(m)$ からデータ m_i をサンプルする必要があり，今回は離散一様分布 $p(m) = \frac{1}{2^K}$ から 100 個の m_i をサンプルした．ただし，K は記憶のビット数である．最後に，計算された内発的報酬を用いて SAC の Actor と Critic ネットワークをそれぞれ 128 エポック繰り返し更新し，次の Global Step に移る．

また，q_ϕ，Actor，Critic の 3 つのニューラルネットワークは，入出力層に加え，1,024 次元の隠れ層を 2 つ用いて構成した．SAC の方策は連続値を出力するため，記憶を編集する行動 a_m は，最初は $-1 \sim 1$ の連続値が出力されるが，のちに 0/1 のビットに正規化して用いた．

A.3　学習した方策によるエージェントの動き

図 A·1 と図 A·2 では，12 ビットの外部記憶を用いて，エージェントは前方と後方に走ることを習得している．一方，図 A·3 では，1 ビットの外部記憶を用いたエージェントが画面中央に留まっており，方策が上手く学習されていない．

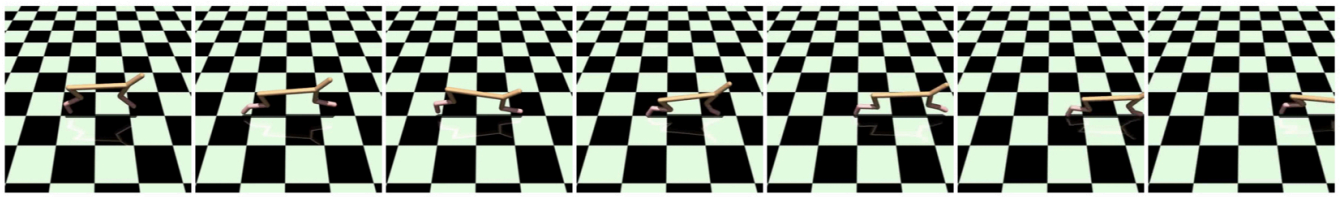

図 A·1: 12 ビットの外部記憶を用いて学習した方策によるエージェントの動き（前方）.

図 A·2: 12 ビットの外部記憶を用いて学習した方策によるエージェントの動き（後方）.

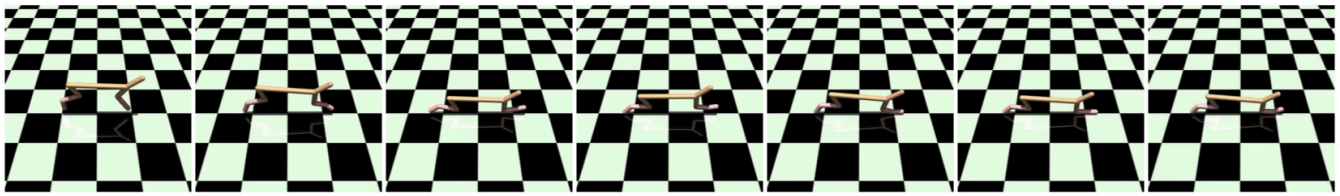

図 A·3: 1 ビットの外部記憶を用いて学習した方策によるエージェントの動き.