

令和2年度卒業論文

自由エネルギー原理に基づく ロボットの好奇心アルゴリズム

指導教員印	提出印
	

指導教員 水内 郁夫 教授

東京農工大学
工学部 機械システム工学科

平成29年度入学

17255038

川原 大典

目 次

第1章 序論	9
1.1 研究の背景と目的	11
1.2 本論文の構成	12
第2章 自由エネルギー原理と好奇心の関係	13
2.1 はじめに	15
2.2 自由エネルギー原理	15
2.2.1 自由エネルギー	15
2.2.2 期待自由エネルギー	17
2.2.3 期待自由エネルギーと好奇心の関係	17
2.2.4 内発的動機づけと外発的動機づけ	19
2.3 おわりに	20
第3章 単純なタスク学習におけるロボット的好奇心アルゴリズムの提案と検証実験	21
3.1 はじめに	23
3.2 好奇心アルゴリズムに基づく行動選択	23
3.2.1 最尤推定とベイズ推定	23
3.2.2 自由エネルギーの最小化	24
3.2.3 期待自由エネルギーの計算	31
3.3 実験方法	33
3.4 実験結果	34
3.5 考察	35
3.6 おわりに	35
第4章 複雑なタスク学習におけるロボット的好奇心アルゴリズムの提案と検証実験	41
4.1 はじめに	43
4.2 好奇心アルゴリズム	43
4.2.1 Bayesian Neural Network (BNN)	43
4.2.2 提案手法：BNN + Curiosity	48
4.3 実験方法	49
4.4 実験結果	50

4.5 考察	50
4.6 おわりに	51
第5章 深層強化学習におけるロボットの好奇心アルゴリズムの提案と検証実験	53
5.1 はじめに	55
5.2 深層強化学習	55
5.2.1 Deep Deterministic Policy Gradient (DDPG)	56
5.2.2 Soft Actor Critic (SAC)	60
5.3 従来研究	62
5.4 好奇心アルゴリズム	64
5.5 カメラを用いた物体操作の学習	68
5.5.1 実験方法	68
5.5.2 実験結果	69
5.5.3 考察	70
5.6 触覚センサを用いた5本指ロボットハンドによる 物体操作タスクの学習	70
5.6.1 実験方法	71
5.6.2 実験結果	73
5.6.3 考察	73
5.7 触覚センサを用いた3本指ロボットハンドによる 物体操作タスクの学習	76
5.7.1 実験方法	76
5.7.2 実験結果	79
5.7.3 考察	79
5.8 おわりに	79
第6章 好奇心行動によるロボットハンドの物体操作実験	83
6.1 はじめに	85
6.2 実験方法	85
6.3 実験結果	86
6.4 考察	86

6.5 おわりに	87
第7章 結論と今後の展望	89
7.1 結論	91
7.2 今後の展望	91
謝辞	92
参考文献	96

第1章

序論

1.1 研究の背景と目的

ロボット制御などに深層強化学習 [1] が盛んに用いられている。深層強化学習によって、例えば、バラ積み部品のピッキング [2][3] やロボットの歩容動作 [4][5]、衣類の折り畳み [6][7]、ボトルの蓋の開け閉め [8] など様々なタスクをロボットが自律的に学習することが可能になっている。このような研究成果があるにもかかわらず、深層強化学習は教師あり学習のような機械学習と比べると実社会での応用例は限られている。その理由の一つとして、強化学習におけるサンプル効率の悪さが挙げられる。

強化学習ではエージェントは試行錯誤の中で環境から受け取ることができる報酬を最大化するような行動方策を自律的に学習する。したがって、教師あり学習のように入力に対する答えが直接与えられるわけではなく、エージェントは試行錯誤しながら各状態に対してどう行動すべきかを学習する必要があり、多くのサンプルを必要とする。一般的に、強化学習ではエージェントはランダムな行動によって環境を探索する [9][10]。しかし、エージェントが環境から受け取る報酬が疎である場合、つまりエージェントが 0 ではない報酬を受け取る機会が少ない場合や、ロボットの物体操作タスクのような状態・行動空間が高次元かつ連続的な場合には、ランダムな行動で全体の環境を見て回るのは難しく、サンプル効率は非常に悪くなる [11][12][13]。サンプル効率を改善するためのアプローチの一つに、人間がもつ好奇心をロボットに適用する手法がある。ロボットに好奇心を与えることで、これまで訪れた状態を繰り返すのではなく、まだ経験していない未知の状態へ探索を促すことができる。これにより、ランダムな行動よりも効率的に環境を探索することができ、サンプル効率の改善が期待できる。しかし、サンプル効率改善のために好奇心を取り入れた従来研究の多くはゲーム環境 [14][15] [16] で行われており、ロボットのタスク学習への応用は限られている。

本研究の目的は、ロボットのタスク学習において、サンプル効率の良いデータ収集のための好奇心アルゴリズムを提案し、その有効性をシミュレーションと実機を使った実験により検証することである。

1.2 本論文の構成

本論文は、全7章から構成される。以下に各章の概要を述べる。

- 第1章「序論」では、本研究の背景と目的について述べた。
- 第2章「自由エネルギー原理と好奇心の関係」では、自由エネルギー原理の概要と好奇心との関係について述べる。
- 第3章「単純なタスク学習におけるロボットの好奇心アルゴリズムの提案と検証実験」では、単純なタスク学習におけるロボットの好奇心アルゴリズムを提案し、提案手法の有効性をシミュレーション実験により検証する。
- 第4章「複雑なタスク学習におけるロボットの好奇心アルゴリズムの提案と検証実験」では、複雑なタスク学習におけるロボットの好奇心アルゴリズムを提案し、提案手法の有効性をシミュレーション実験により検証する。
- 第5章「深層強化学習におけるロボットの好奇心アルゴリズムの提案と検証実験」では、深層強化学習におけるロボットの好奇心アルゴリズムを提案し、提案手法の有効性をシミュレーション実験により検証する。
- 第6章「好奇心行動によるロボットハンドの物体操作実験」では、前章で述べた提案手法の妥当性を実機実験により検証する。
- 第7章「結論と今後の展望」では、結論及び今後の展望について述べる。

第2章

自由エネルギー原理と好奇心の関係

2.1 はじめに

本研究ではロボットに好奇心をもたせるために、自由エネルギー原理[17]に着目する。本章では、自由エネルギー原理について説明し、好奇心との関係について述べる。

2.2 自由エネルギー原理

自由エネルギー原理は脳の様々な認知機能を統一的に説明する理論として、University College London の神経科学者 Karl Friston によって 2006 年に提唱された[18]。自由エネルギー原理によって説明される脳機能は多岐にわたり、例えば、錯視[19]、眼球運動[20][21]、アフォーダンス[22]、注意[23]、模倣[24]、自閉症[25]、サヴァン症候群[26]、身体化による認知[27]、運動制御[28]、知覚推論[29]、意思決定[30]などがある。脳に関する理論には、シナプス可塑性についての法則であるヘブ則(Hebb's rule)[31]や脳における推論をベイズ統計学の原理になぞらえて説明するベイズ脳理論[32][33]、神経回路の結合は情報の伝達・保持能力が高くなるよう学習されるとする情報量最大化原理(Infomax principle)[34][35][36]、外界から知覚した信号と脳で予測した信号との差、つまり予測誤差を最小化するように情報を表現する予測符号化理論(Predictive coding)[37][38]、運動制御における最適化理論[39][40]などそれぞれのドメインに特化した理論が存在するが、自由エネルギー原理は脳についてのこれらの理論を統一的に説明する「原理」であり、これまでの理論を置き換えるためのものではない[41]。

theories[41]。

2.2.1 自由エネルギー

自由エネルギー原理では、生物は感覚入力の予測誤差を意味する自由エネルギーを最小化するように学習・推論すると定めている[42]。自由エネルギーはベイズ推定における近似法の1つである変分ベイズ法[43]を行うときに使われる値として定義される。ベイズ推定では、パラメータ θ の事前確率分布 $P(\theta)$ を設定した上で、データ D を観測した後の事後確率分布 $P(\theta|D)$ を推定する。事後確率分布 $P(\theta|D)$ はベイズの定理を用いて、次式のように表される。

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{\int P(D, \theta)d\theta} = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta} \quad (2.1)$$

一般的に、式(2.1)の分母の積分は解析的に解が求まらない場合が多い。ここでの目的は、確率分布 $Q(\theta)$ を真の事後確率分布 $P(\theta|D)$ になるべく近づけることである。2つの確率分布の差異を計る尺度として KL 情報量 (Kullback-Leibler Divergence) [44] がある。確率分布 $Q(\theta)$ と $P(\theta|D)$ の間の KL 情報量は次式のように表される。

$$D_{KL}[Q(\theta)||P(\theta|D)] = \int Q(\theta) \ln \frac{Q(\theta)}{P(\theta|D)} d\theta \quad (2.2)$$

しかし、式(2.2)中には $P(\theta|D)$ が入っているためこのままでは計算できない。そこで、変分ベイズ法ではベイズの定理 $P(\theta|D) = \frac{P(D, \theta)}{P(D)}$ を用いることで、式(2.2)を次のように変形する。

$$\begin{aligned} D_{KL}[Q(\theta)||P(\theta|D)] &= \int Q(\theta) \ln \frac{Q(\theta)P(D)}{P(D, \theta)} d\theta \\ &= \int Q(\theta) \left\{ \ln \frac{Q(\theta)}{P(D, \theta)} + \ln P(D) \right\} d\theta \\ &= \int Q(\theta) \ln \frac{Q(\theta)}{P(D, \theta)} d\theta + \ln P(D) \end{aligned} \quad (2.3)$$

ここで、

$$F = \int Q(\theta) \ln \frac{Q(\theta)}{P(D, \theta)} d\theta \quad (2.4)$$

$$= D_{KL}[Q(\theta)||P(D, \theta)] \quad (2.5)$$

とおくと、式(2.3)は次式のように表される。

$$D_{KL}[Q(\theta)||P(\theta|D)] = F + \ln P(D) \quad (2.6)$$

真の事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta)$ は $D_{KL}[Q(\theta)||P(\theta|D)]$ の最小化、つまり、2つ確率分布 $Q(\theta)$ 、 $P(\theta|D)$ の差異の最小化によって求まる。式(2.6)の $\ln P(D)$ はパラメータ θ に依存しないため、近似解 $Q^*(\theta)$ は以下のように求まる。

$$\begin{aligned} Q^*(\theta) &= \arg \min_{Q(\theta)} D_{KL}[Q(\theta)||P(\theta|D)] \\ &= \arg \min_{Q(\theta)} F \end{aligned} \quad (2.7)$$

式(2.7)の F は（変分）自由エネルギーと呼ばれる。変分ベイズ法ではこのように自由エネルギー F の最小化によって真の事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta)$ を求める。

2.2.2 期待自由エネルギー

自由エネルギー原理では、生物は自由エネルギー F の最小化によって学習・推論するだけでなく、将来期待される自由エネルギー（期待自由エネルギー） G を最小化するよう行動すると定めている[45]。行動を a 、時刻を τ としたときの期待自由エネルギー $G(a, \tau)$ は、自由エネルギー F に期待値として尤度（likelifood） $P(D_\tau | \theta_\tau)$ をかけることで次のように求まる[46]。

$$\begin{aligned}
G(a, \tau) &= E_{P(D_\tau | \theta_\tau)}[F] \\
&= E_{P(D_\tau | \theta_\tau)}\left[\int Q(\theta_\tau | a) \ln \frac{Q(\theta_\tau | a)}{P(D_\tau, \theta_\tau | a)} d\theta\right] \\
&= E_{P(D_\tau | \theta_\tau)}\left[\int Q(\theta_\tau | a)\{\ln Q(\theta_\tau | a) - \ln P(D_\tau, \theta_\tau | a)\} d\theta\right] \\
&= E_{P(D_\tau | \theta_\tau)}\left[\int Q(\theta_\tau | a)\{\ln Q(\theta_\tau | a) - \ln P(\theta_\tau | D_\tau, a) - \ln P(D_\tau)\} d\theta\right] \\
&= E_{P(D_\tau | \theta_\tau)}[E_{Q(\theta_\tau | a)}[\ln Q(\theta_\tau | a) - \ln P(\theta_\tau | D_\tau, a) - \ln P(D_\tau)]] \\
&\approx E_{Q(D_\tau | \theta_\tau)}[E_{Q(\theta_\tau | a)} \ln \frac{Q(\theta_\tau | a)}{Q(\theta_\tau | D_\tau, a)} - \ln P(D_\tau)] \quad (2.8) \\
&= -E_{Q(D_\tau, \theta_\tau | a)}[\ln \frac{Q(\theta_\tau | D_\tau, a)}{Q(\theta_\tau | a)}] - E_{Q(D_\tau, \theta_\tau | a)}[\ln P(D_\tau)] \\
&= -E_{Q(D_\tau | a)}[D_{KL}[Q(\theta_\tau | D_\tau, a) || Q(\theta_\tau | a)]] - E_{Q(D_\tau, \theta_\tau | a)}[\ln P(D_\tau)] \quad (2.9)
\end{aligned}$$

ただし、式(2.8)において、 $P(D_\tau | \theta_\tau) \approx Q(D_\tau | \theta_\tau)$, $P(\theta_\tau | D_\tau, a) \approx Q(\theta_\tau | D_\tau, a)$ と近似した。

将来に渡る期待自由エネルギーの和は

$$G(a) = \sum_{\tau=t+1}^T G(a, \tau) \quad (2.10)$$

となる。

したがって、行動 a は次のような。

$$a = \arg \min_a G(a) \quad (2.11)$$

2.2.3 期待自由エネルギーと好奇心の関係

式(2.9)の第1項中の $D_{KL}[Q(\theta_\tau | D_\tau, a) || Q(\theta_\tau | a)]$ は2つの確率分布 $Q(\theta_\tau | D_\tau, a), Q(\theta_\tau | a)$ 間の差異を表している。これは、行動 a によってデータ D を得ることでパラメータ θ に

についての確率分布 $Q(\theta)$ がどの程度更新されたかを表している。この値が大きいほど期待自由エネルギー G は小さくなる。したがって、生物はパラメータ θ についての確率分布 $Q(\theta)$ がより大きく更新されると期待されるデータ D を求めて行動することになる。また、 $D_{KL}[Q(\theta_\tau|D_\tau, a) || Q(\theta_\tau|a)]$ の値はベイジアン・サプライズ (Bayesian Surprise) [47] と呼ばれており、サプライズの大きさをベイズ的立場から定量化したものである。生物はパラメータ θ についての情報がより多く得られるデータ D を求めて行動を選択することになる。

ある出来事が起こる「不規則性」や「不確かさ」、「曖昧さ」を表す概念として、情報科学の分野におけるエントロピー（平均情報量、シャノン情報量） H が知られている。パラメータ θ のエントロピー $H(\theta)$ は次式で表される。

$$H(\theta) = E_{Q(\theta)}[-\ln Q(\theta)] = - \int Q(\theta) \ln Q(\theta) d\theta \quad (2.12)$$

式 (2.12) 中の $-\ln Q(\theta)$ は単に情報量、あるいは選択情報量、自己情報量などと呼ばれる。パラメータ θ の起こる確率 $Q(\theta)$ が小さい程、情報量は大きくなる。情報量の期待値がエントロピー $H(\theta)$ である。また、データ D を得たもののパラメータ θ のエントロピー $H(\theta|D)$ は条件付きエントロピーと呼ばれ、次式で表される。

$$H(\theta|D) = E_{Q(D)} E_{Q(\theta|D)}[-\ln Q(\theta|D)] = E_{Q(D,\theta)}[-\ln Q(\theta|D)] \quad (2.13)$$

相互情報量 $I(\theta; D)$ [48] は式 (2.12), 式 (2.13) を用いて次式で表される。

$$I(\theta; D) = H(\theta) - H(\theta|D) \quad (2.14)$$

これは、データ D を得る前のパラメータ θ についてのエントロピー $H(\theta)$ からデータ D を得た後のパラメータ θ のエントロピー $H(\theta|D)$ を引いた値になっている。つまり、相互情報量 $I(\theta; D)$ はデータ D を得ることでパラメータ θ についての曖昧さがどの程度減少したかを表している。

$E_{Q(D|\theta)}[-\ln Q(\theta)] = 1$ が成り立つことに注意すると、式 (2.12) は以下のように変形できる。

$$\begin{aligned} H(\theta) &= E_{Q(\theta)}[-\ln Q(\theta)] \\ &= E_{Q(D|\theta)} E_{Q(\theta)}[-\ln Q(\theta)] \\ &= E_{Q(D,\theta)}[-\ln Q(\theta)] \end{aligned} \quad (2.15)$$

このとき、式(2.14)は以下のようになる。

$$\begin{aligned}
 I(\theta; D) &= H(\theta) - H(\theta|D) \\
 &= E_{Q(D,\theta)}[-\ln Q(\theta)] - E_{Q(D,\theta)}[-\ln Q(\theta|D)] \\
 &= E_{Q(D,\theta)}[\ln \frac{Q(\theta|D)}{Q(\theta)}] \\
 &= E_{Q(D)}[D_{KL}[Q(\theta|D)||Q(\theta)]] \tag{2.16}
 \end{aligned}$$

式(2.16)は、式(2.9)の第1項にマイナスをつけた値になっている。逆に言えば、式(2.9)の第1項は負の相互情報量 $-I(\theta; D)$ であることがわかる。先に述べたように、自由エネルギー原理では生物は期待自由エネルギー G が小さくなる行動をとる。期待自由エネルギー G を小さくするには式(2.9)の第1項は小さいほどよく、それには相互情報量 $I(\theta; D)$ は大きいほうがよい。式(2.9)の第1項には行動 a が含まれているが、これは相互情報量 $I(\theta; D)$ が大きくなるような、つまり、パラメータ θ についての曖昧さができるだけ減るようなデータ D を行動 a によって変えられることを意味する。曖昧さを減らしていくというのは、言い換えると、自分が知らないことを知ろうとすることであり、好奇心の性質に近いものがある。

式(2.9)の第2項 $-E_{Q(D_\tau, \theta_\tau|a)}[\ln P(D_\tau)]$ 中の事前確率分布 $P(D_\tau)$ は自由エネルギー原理では、生物にとっての事前の選好(prior preference)とみなされる。事前確率分布 $P(D_\tau)$ が大きくなるような、つまり、事前の選好が大きくなるようなデータ D を得るほど第2項は全体として小さくなり、期待自由エネルギー G は小さくなる。ロボットの場合、事前の選好はタスクに応じて人間が与える必要がある。これは強化学習における報酬に対応する。

2.2.4 内発的動機づけと外発的動機づけ

心理学の考え方で、内発的動機づけ(Intrinsic Motivation)と外発的動機づけ(Extrinsic Motivation)というものがある[49]。

内発的動機づけとは、好奇心や興味・関心、探求心による自分の内側から起こる動機づけであり、他者からの評価や賞罰に依存しない行動である。これは特に子供は好奇心が極めて高いために幼児期によく見られる動機づけである。一方、外発的動機づけとは、評価や賞罰、報酬、強制などによってもたらされる動機づけであり、外発的動機づけに基づいた行動は外部から与えられた目標を達成するためのものである。例えば、「テストで良い点数を取ったら新しいゲームを買ってもらえるので勉強する」や「お手伝いをしなかったらお小遣いを減らされるのでお手伝いをする」などが外発的動機づけにあたる。

ここで式(2.9)に示す期待自由エネルギー G をもう一度見てみると、第1項はタスクとは独立しており、未知のパラメータに対する自身の認識の不確かさ・曖昧さを減らすことが行動の動機づけになっており、これは内発的動機づけに対応すると考えられる。一方、式(2.9)の第2項は外部から事前に与えられる情報が行動の動機づけになっており、これは外発的動機づけに対応すると考えられる。

データ D に関する事前の選好がない場合（強化学習では環境からの報酬がない場合）、データ D に関する事前確率分布は $P(D_\tau) = 0$ となり式(2.9)に示す期待自由エネルギー G は以下のようになる。

$$\begin{aligned} G(a, \tau) &= -E_{Q(D_\tau|a)}[D_{KL}[Q(\theta_\tau|D_\tau, a)||Q(\theta_\tau|a)]] \\ &= -(H(\theta_\tau) - H(\theta_\tau|D_\tau, a)) \end{aligned} \quad (2.17)$$

つまり、生物（本研究ではロボット）は、未知のパラメータ θ についての曖昧さをできるだけ減らす行動を選択することになる。

2.3 おわりに

本章では自由エネルギー原理について説明し、好奇心との関係について述べた。本研究では、自由エネルギー原理に基づいた好奇心をロボットにもたらすことによってタスク学習におけるサンプル効率の改善を期待する。第3章以降でいくつかのタスク学習を行い、本研究の提案手法の有効性を検証する。

第3章

単純なタスク学習におけるロボットの
好奇心アルゴリズムの提案と検証実験

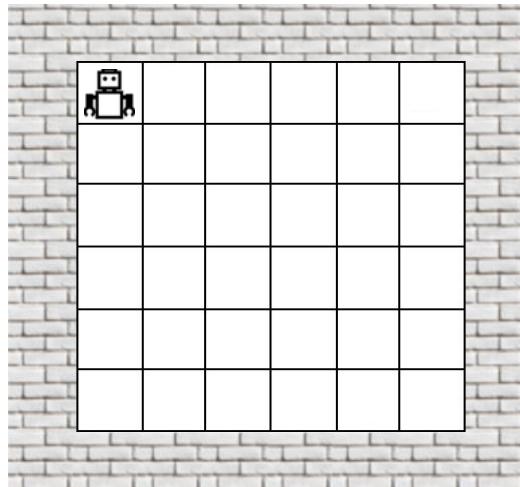


Fig. 3.1: Robot in 6×6 room

3.1 はじめに

本章では、単純なタスク学習において、サンプル効率の良いデータ収集のためのロボットの好奇心アルゴリズムの提案し、その有効性をシミュレーション実験により検証する。

本章では、図 3.1 に示す 6×6 マスの部屋において、ロボットが各マスにおける計測モデルを学習することをタスクとする。ここでいう計測モデルとは、ロボットのセンサで計測される値の確率分布である。ロボットは計測モデルの事前知識がない状態からスタートし、部屋の中を移動することで各マスで計測したデータを使って計測モデルを学習する。

3.2 好奇心アルゴリズムに基づく行動選択

本章では、図 3.1 の各マスに落ちている石ころの数を計測することを考える。各マスの計測モデルはガウス分布とする。

3.2.1 最尤推定とベイズ推定

今、計測モデルはガウス分布を想定しているので、学習する計測モデルのパラメータ θ はガウス分布の平均 μ と精度（分散の逆数） τ である。計測したデータ D からパラメータ θ を

学習する方法として、機械学習では最尤推定 (Maximum Likelihood Estimation) [50] が使われることがある。最尤推定により、ガウス分布の平均 μ と精度 τ は以下のように求まる。

$$\mu = \frac{x_1 + x_2 + \cdots + x_N}{N} \quad (3.1)$$

$$\frac{1}{\tau} = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (3.2)$$

ただし、 N はデータ $D = \{x_1, x_2, \dots, x_N\}$ に含まれる計測値 x の数である。

しかし、データ数が少い場合やモデルが複雑な場合はとくに、最尤推定は与えられたデータに対してモデルを過剰に当てはめてしまう過学習あるいは過剰適合 (overfitting) という現象が起こることが知られている [51][52]。また、現実世界では計測モデルがガウス分布になるとは限らず、パラメータ $\theta = \{\mu, \tau\}$ の不確実性も考慮する必要がある。最尤推定は点推定であり、パラメータ推定における確からしさを考慮していない。一方、ベイズ推定では 2.2.1 節で述べたように、パラメータ θ に関して事前確率分布 $P(\theta)$ を導入することでパラメータ推定における不確実性を定量的に扱える。また、モデルの複雑さに関する情報も事前にモデルに組み込むことができる。これにより、データ数が少い場合やモデルの自由度が大きく複雑な場合でも過剰適合を避けることができる。

ベイズ推定では、データ D が与えられたときの計測モデルのパラメータ $\theta = \{\mu, \tau\}$ の事後確率分布 $P(\theta|D)$ を求める。

3.2.2 自由エネルギーの最小化

式 (2.7) に示したように、事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta)$ は自由エネルギー F の最小化によって求まる。以下、具体的に近似解 $Q^*(\theta)$ を求める。

まず、確率分布 $Q(\theta)$ は平均場近似 [53] に基づき、次式のように分解する。

$$Q(\theta) = Q(\mu)Q(\tau) \quad (3.3)$$

式 (3.3) の形を持つ確率分布 $Q(\theta)$ の中で、自由エネルギー F が最小となるものを探したい。したがって、それぞれの確率分布 $Q(\mu), Q(\tau)$ について自由エネルギー F を最小化するため、各因子 $Q(\mu), Q(\tau)$ について順番に最適化を行っていく。このため、まず式 (3.3) を式 (2.4) に代入し、因子の一つ $Q(\mu)$ に対する依存項を取り出してみると、

$$\begin{aligned}
 F &= \iint Q(\mu)Q(\tau) \ln \frac{Q(\mu)Q(\tau)}{P(D, \theta)} d\mu d\tau \\
 &= \iint Q(\mu)Q(\tau) \{\ln Q(\mu)Q(\tau) - \ln P(D, \theta)\} d\mu d\tau \\
 &= \int Q(\mu) \ln Q(\mu) d\mu - \int Q(\mu) \int \ln P(D, \theta) Q(\tau) d\tau d\mu + const \\
 &= \int Q(\mu) \ln Q(\mu) d\mu - \int Q(\mu) E_\tau [\ln P(D, \theta)] d\mu + const \quad (3.4) \\
 &= D_{KL}[Q(\mu) || E_\tau [\ln P(D, \theta)]] + const \quad (3.5)
 \end{aligned}$$

となる。ただし、 E_τ は精度 τ の確率分布 $Q(\tau)$ での期待値を表す。したがって、

$$E_\tau [\ln P(D, \theta)] = \int \ln P(D, \theta) Q(\tau) d\tau \quad (3.6)$$

である。

自由エネルギー F が最小となるのは式 (3.4) より、 $\ln Q(\mu) = E_\tau [\ln P(D, \theta)]$ のときである。これは、因子 $Q(\mu)$ の最適解の対数は、データ D とパラメータ θ の同時確率分布の対数を考え、 $Q(\mu)$ 以外の因子（ここでは $Q(\tau)$ ）すべてについての期待値を取ったものに等しいことを意味する。同時確率分布 $P(D, \theta)$ は次式のように表される。

$$P(D, \theta) = P(D, \mu, \tau) = P(D|\mu, \tau)P(\mu|\tau)P(\tau) \quad (3.7)$$

また、尤度関数 $P(D|\mu, \tau)$ は次式で与えられる。

$$P(D|\mu, \tau) = \prod_{i=1}^N \sqrt{\frac{\tau}{2\pi}} \exp\left\{-\frac{\tau}{2}(x_i - \mu)^2\right\} \quad (3.8)$$

さらに、 μ と τ に関する共役事前分布を次式のように与える。

$$P(\mu|\tau) = \mathcal{N}(\mu|\mu_0, (\lambda_0\tau)^{-1}) \quad (3.9)$$

$$P(\tau) = \text{Gam}(\tau|a_0, b_0) \quad (3.10)$$

$\mathcal{N}(\mu|\mu_0, (\lambda_0\tau)^{-1})$ は平均が μ_0 、分散が $(\lambda_0\tau)^{-1}$ のガウス分布であり、 $\text{Gam}(\tau|a_0, b_0)$ は次式で定義されるガンマ分布である。

$$\text{Gam}(\tau|a, b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} \exp(-b\tau) \quad (3.11)$$

ここで、 Γ は次式で定義されるガンマ関数で、式 (3.11) が正規化されることを保証する。

$$\Gamma(a) = \int_0^\infty t^{a-1} \exp^{-t} dt \quad (3.12)$$

このとき、因子 $Q(\mu)$ の最適解 $Q^*(\mu)$ は以下のようになる。

$$\begin{aligned} \ln Q^*(\mu) &= E_\tau[\ln P(D, \theta)] \\ &= E_\tau[\ln P(D|\mu, \tau) + \ln P(\mu|\tau) + \ln P(\tau)] \\ &= E_\tau[\ln P(D|\mu, \tau) + \ln P(\mu|\tau)] + const \\ &= -\frac{E[\tau]}{2} \left\{ \sum_{i=1}^N (x_i - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right\} + const \end{aligned} \quad (3.13)$$

上式を μ に関して平方完成すると、 $Q(\mu)$ はガウス分布 $\mathcal{N}(\mu|\mu_N, \lambda_N^{-1})$ であり、平均 μ_N と精度 λ_N はそれぞれ以下で与えられる。

$$\mu_N = \frac{\lambda_0 \mu_0 + N \bar{x}}{\lambda_0 + N} \quad (3.14)$$

$$\lambda_N = (\lambda_0 + N)E[\tau] \quad (3.15)$$

ただし、 $E[\tau]$ はガンマ分布 $Gam(\tau|a, b)$ の期待値であり、次式で表される。

$$E[\tau] = \frac{a}{b} \quad (3.16)$$

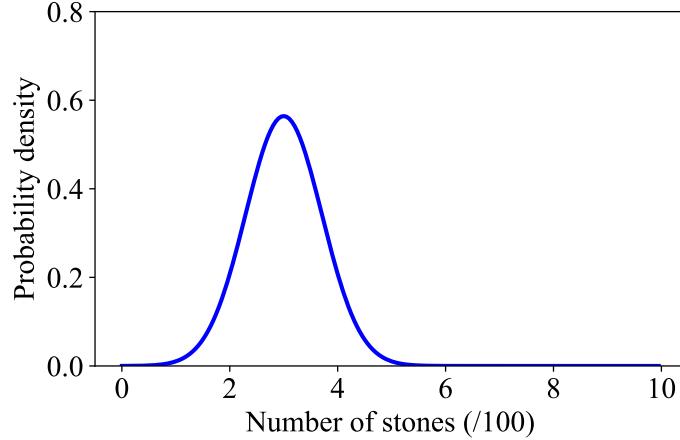
同様にして、因子 $Q(\tau)$ の最適解 $Q^*(\tau)$ は以下のようになる。

$$\begin{aligned} \ln Q^*(\tau) &= E_\mu[\ln P(D, \theta)] \\ &= E_\mu[\ln P(D|\mu, \tau) + \ln P(\mu|\tau) + \ln P(\tau)] \\ &= (a_0 - 1)\ln \tau - b_0 \tau + \frac{N+1}{2}\ln \tau \\ &\quad - \frac{\tau}{2}E_\mu \left[\sum_{i=1}^N (x_i - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right] + const \end{aligned} \quad (3.17)$$

よって、 $Q(\tau)$ はガンマ分布 $Gam(\tau|a_N, b_N)$ であり、そのパラメータは以下のようになる。

$$a_N = a_0 + \frac{N+1}{2} \quad (3.18)$$

$$b_N = b_0 + \frac{1}{2}E_\mu \left[\sum_{i=1}^N (x_i - \mu)^2 + \lambda_0(\mu - \mu_0)^2 \right] \quad (3.19)$$

Fig. 3.2: Gaussian distribution ($\mu = 3.0, \tau = 2.0$)

このようにして、それぞれの因子 $Q(\mu), Q(\tau)$ に関して自由エネルギー F を最小化する最適解の式が得られ、それらは互いの期待値に依存していることがわかった。したがって、これを解く方法の一つは、例えば $E[\tau]$ の初期値を推定し、これを用いて確率分布 $Q(\mu)$ を再計算することである。この改良された確率分布を使うと必要な期待値 $E[\mu] = \mu_N, E[\mu^2] = \mu_N^2 + \lambda_N^{-1}$ が求められ、確率分布 $Q(\tau)$ を再計算することができる。以上を繰り返すことで、事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta) = Q^*(\mu)Q^*(\tau)$ が得られる。

ここで例として、図 3.1 のロボットがいるマスにおける計測モデルの近似解 $Q^*(\theta)$ を求めてみる。このマスの真の計測モデルは図 3.2 に示すガウス分布（平均 $\mu = 3.0$, 精度 $\tau = 2.0$ ）とする。ここでは特定のマスにおける計測モデルのみを考えるため、ロボットは部屋の中を移動せず、毎ステップその場で停止するものとする。ロボットは 1 ステップごとに図 3.2 に示すガウス分布由来のデータ $D = \{x_1, x_2, x_3\} (N = 3)$ を計測するものとする。つまり、1 ステップごとにそのマスで 3 回計測することになる。

ステップ 1～ステップ 5 の間に計測したデータを表 3.1 に示す。各ステップで求めた真の事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta)$ を図 3.3 に示す。また、確率分布 $Q^*(\theta)$ の平均 μ と精度 τ それぞれの期待値を使った計測モデルを図 3.4 に示す。ただし、ステップ 0 における事前確率分布 $P(\theta)$ に必要なパラメータの初期値は $\mu_0 = 5.0, \lambda_0 = 10, a_0 = 2, b_0 = 2$ とした。このとき、 $E[\tau]$ の初期値は式 (3.16) より、次のようになる。

$$E[\tau] = \frac{a_0}{b_0} = \frac{2}{2} = 1 \quad (3.20)$$

Table 3.1: Measurement values included in the data at each step

	x_1	x_2	x_3
Step=1	2.96	1.99	3.05
Step=2	2.09	2.03	3.63
Step=3	3.13	3.19	2.61
Step=4	3.25	3.01	4.07
Step=5	3.29	3.70	3.62

図 3.3 より、そのマスにおけるデータのサンプル数が増えるほどパラメータ $\theta = \{\mu, \tau\}$ についての曖昧さが小さくなっていく様子が分かる。また、各ステップ毎の精度 τ の分散の変化と、平均 μ の分散の変化の様子に違いがあることが分かる。 τ の確率分布 $Q^*(\tau)$ は式 (3.17) より、ガンマ分布であり、分散 $var[\tau]$ は次式で表される。

$$var[\tau] = \frac{a_N}{b_N^2} = \frac{E[\tau]}{b_N} \quad (3.21)$$

ただし、 a_N, b_N は式 (3.18)、式 (3.19) から求めた値である。一方、 μ の確率分布 $Q^*(\mu)$ は式 (3.13) より、ガウス分布であり、分散 $var[\mu]$ は式 (3.15) より、次式となる。

$$var[\mu] = \frac{1}{\lambda_N} = \frac{1}{(\lambda_0 + N)E[\tau]} \quad (3.22)$$

τ, μ どちらの分散も τ の期待値 E_τ に依存しており、 τ の分散は期待値 E_τ に比例し、 μ の分散は反比例している。各ステップで得たデータ D がそれより前のステップで得たデータに近いとき、精度の期待値 $E[\tau]$ は大きくなることが考えられ、このとき、 $var[\tau]$ の値も比例して大きくなり、 $var[\mu]$ の値は反比例して小さくなることが考えられる。図 3.3 を見ると、各ステップにおいて $var[\mu]$ のほうが $var[\tau]$ より小さく変化しているが、これは表 3.1 に示す各ステップで得たデータがそれより前のステップで得たデータに近いためだと考えられる。

図 3.4 より計測モデルの学習が行えていることが分かる。

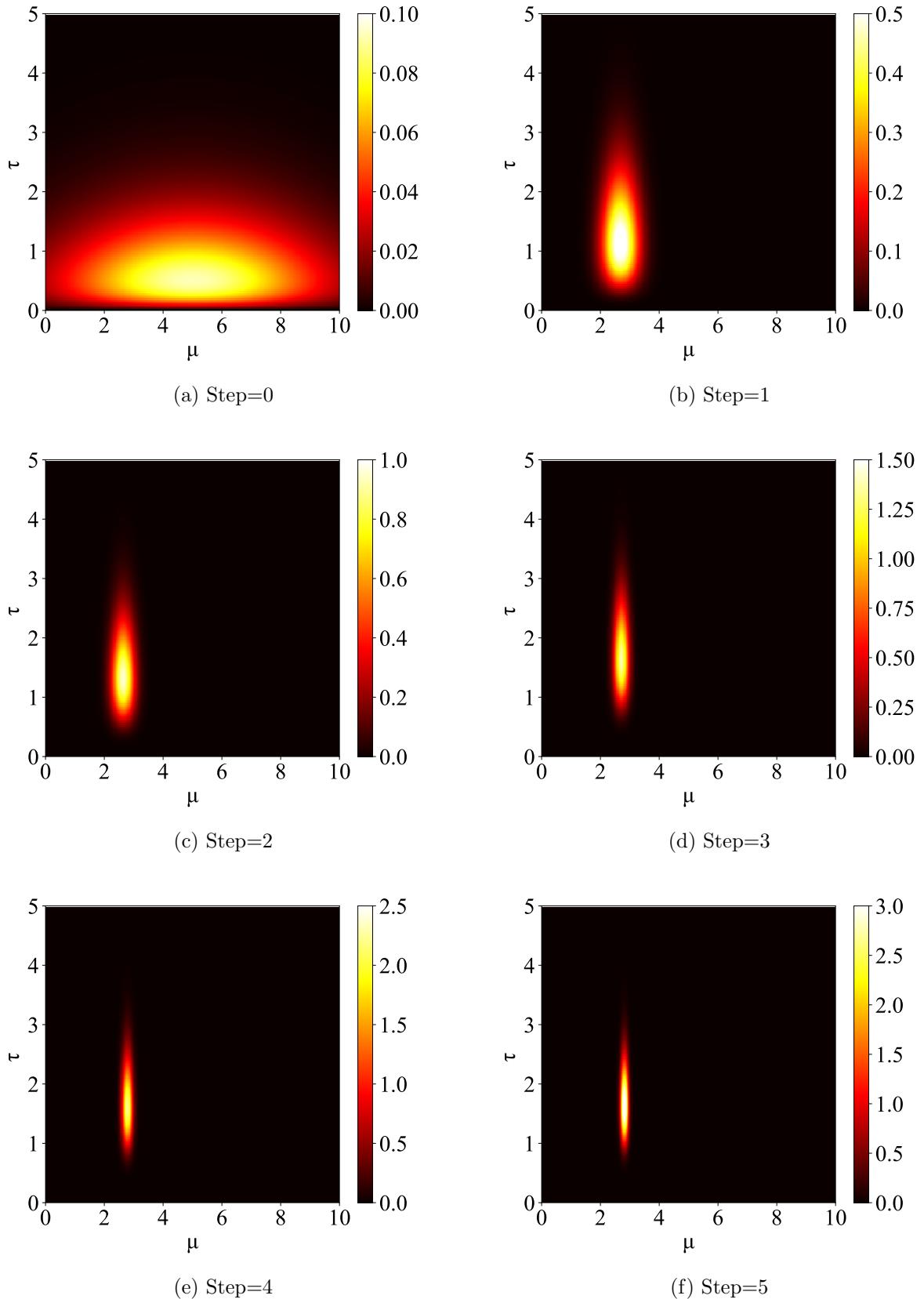


Fig. 3.3: $Q^*(\theta)$ at each step in the cell with the robot shown in Fig.3.1

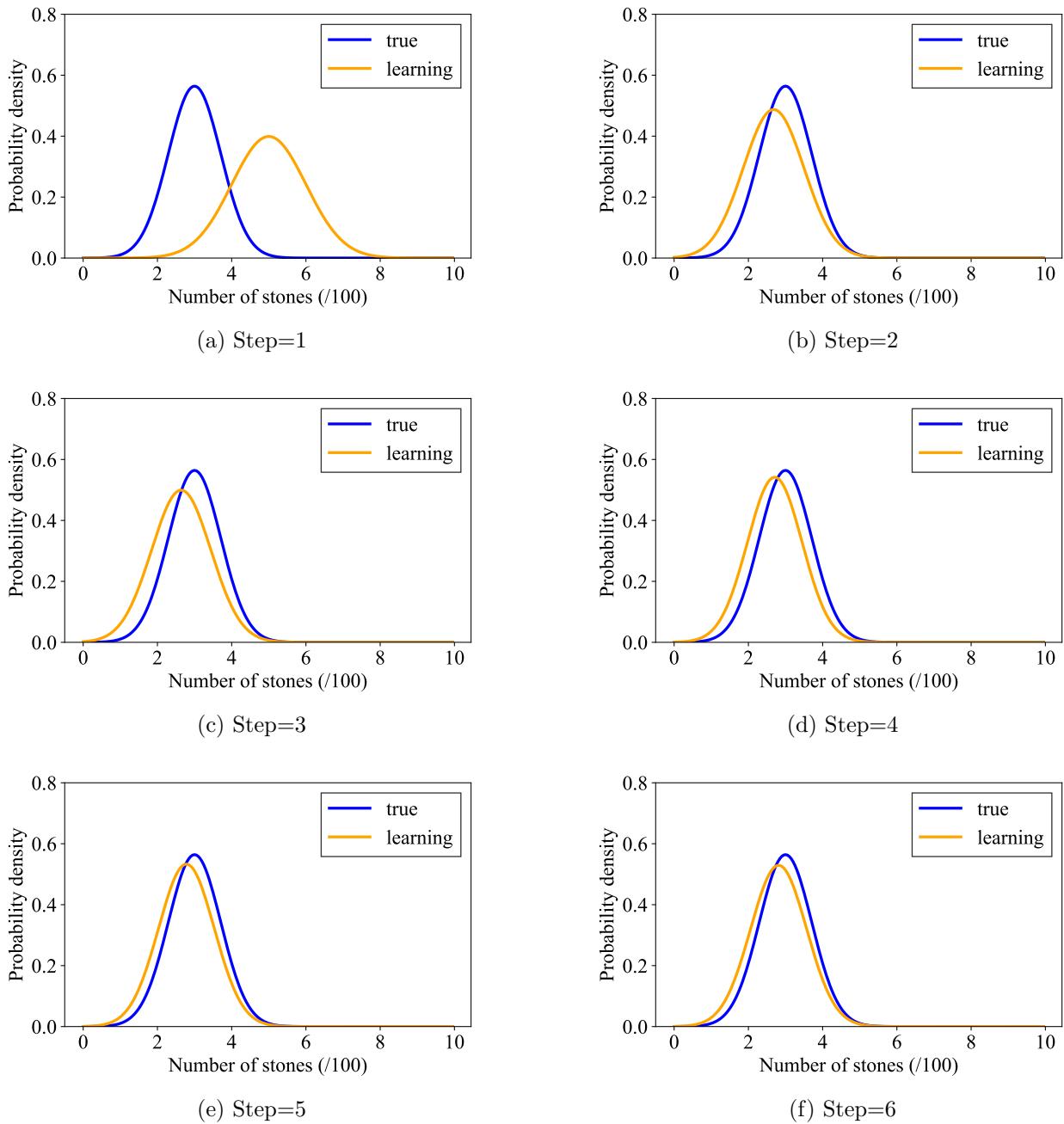


Fig. 3.4: Learning of measurement model at each step in the cell with the robot shown in Fig.3.1

3.2.3 期待自由エネルギーの計算

ロボットは式(2.11)に示すように、期待自由エネルギー G が最小となる行動 a をとる。式(2.17)に示すように、期待自由エネルギー G の計算にはパラメータ θ のエントロピー $H(\theta)$ と行動 a によってデータ D を得た後の条件付きエントロピー $H(\theta|D, a)$ を求める必要がある。

式(2.12)に式(3.3)を代入すると、エントロピー $H(\theta)$ は以下のようになる。

$$\begin{aligned} H(\theta) &= E_{Q(\theta)}[-\ln Q(\theta)] \\ &= E_{Q(\mu)Q(\tau)}[-\ln Q(\mu)Q(\tau)] \\ &= E_{Q(\mu)}E_{Q(\tau)}[-\ln Q(\mu)Q(\tau)] \end{aligned} \quad (3.23)$$

$$\begin{aligned} &= -E_{Q(\mu)}E_{Q(\tau)}[\ln Q(\mu) + \ln Q(\tau)] \\ &= -E_{Q(\mu)}[\ln Q(\mu)] - E_{Q(\tau)}[\ln Q(\tau)] \\ &= H(\mu) + H(\tau) \end{aligned} \quad (3.24)$$

ただし、式(3.23)では平均場近似による $Q(\mu)$ と $Q(\tau)$ の独立性を利用した。

式(3.24)において、ガウス分布 $Q(\mu)$ のエントロピー $H(\mu)$ は以下のようになる。

$$\begin{aligned} H(\mu) &= E_{Q(\mu)}[-\ln Q(\mu)] \\ &= E_{Q(\mu)}[-\ln \mathcal{N}(\mu|\mu_N, \lambda_N^{-1})] \\ &= -E_{Q(\mu)}[-\frac{1}{2}\{\frac{(\mu - \mu_N)^2}{\lambda_N^{-1}} + \ln \lambda_N^{-1} + \ln 2\pi\}] \\ &= E_{Q(\mu)}[\frac{1}{2}\{\frac{\mu^2 - 2\mu_N\mu + \mu_N^2}{\lambda_N^{-1}} + \ln \lambda_N^{-1} + \ln 2\pi\}] \\ &= \frac{1}{2}\{\frac{E_{Q(\mu)}[\mu^2] - 2\mu_N E_{Q(\mu)}[\mu] + \mu_N^2}{\lambda_N^{-1}} + \ln \lambda_N^{-1} + \ln 2\pi\} \\ &= \frac{1}{2}\{\frac{\mu_N^2 + \lambda_N^{-1} - 2\mu_N^2 + \mu_N^2}{\lambda_N^{-1}} + \ln \lambda_N^{-1} + \ln 2\pi\} \\ &= \frac{1}{2}(1 + \ln \lambda_N^{-1} + \ln 2\pi) \end{aligned} \quad (3.25)$$

式(3.25)を見ると、ガウス分布のエントロピーは精度の逆数 λ_N^{-1} (=分散) に依存している。このことからも、エントロピーが「曖昧さ」や「不確かさ」を表していることが分かる。

式(3.24)において、ガンマ分布 $Q(\tau)$ のエントロピー $H(\tau)$ は以下のようになる。

$$\begin{aligned}
 H(\tau) &= E_{Q(\tau)}[-\ln Q(\tau)] \\
 &= E_{Q(\tau)}[-\ln \text{Gam}(\tau|a_N, b_N)] \\
 &= -E_{Q(\mu)}[(a_N - 1)\ln \tau - b_N \tau - \ln \frac{b_N^{a_N}}{\Gamma(a_N)}] \\
 &= -(a_N - 1)E_{Q(\mu)}[\ln \tau] + b_N E_{Q(\mu)}[\tau] - \ln \frac{b_N^{a_N}}{\Gamma(a_N)} \\
 &= -(a_N - 1)(\psi(a_N - \ln b) + b_N \frac{a_N}{b_N} - \ln \frac{b_N^{a_N}}{\Gamma(a_N)}) \\
 &= -(a_N - 1)\psi(a_N) + (a_N - 1)\ln b_N + a_N - \ln b_N^{a_N} + \ln \Gamma(a_N) \\
 &= -(a_N - 1)\psi(a_N) - \ln b_N + a_N + \ln \Gamma(a_N)
 \end{aligned} \tag{3.26}$$

ただし、 $\psi(a_N)$ はディガンマ関数であり次式で表される。

$$\psi(a_N) = \frac{d}{da_N} \ln \Gamma(a_N) = \frac{\Gamma'(a_N)}{\Gamma(a_N)} \tag{3.27}$$

式(3.27)、式(3.25)を式(3.24)に代入すると

$$\begin{aligned}
 H(\theta) &= H(\mu) + H(\tau) \\
 &= \frac{1}{2}\{1 + \ln \lambda_N^{-1} + \ln 2\pi\} - (a_N - 1)\psi(a_N) - \ln b_N + a_N + \ln \Gamma(a_N)
 \end{aligned} \tag{3.28}$$

となる。

条件付きエントロピー $H(\theta|D, a)$ は、行動 a によってデータ D を得た後のパラメータ θ のエントロピーである。期待自由エネルギー G の計算において、条件付きエントロピー $H(\theta|D, a)$ は行動選択の段階で求める必要がある。そのため、 $H(\theta|D, a)$ 中のデータ D は実際に行動をとった後に得られるデータではなく、その行動 a を取ることによって得られるであろうデータになる。これには、行動 a を取ることによって得られるデータの期待値を考えればよく、その場合のパラメータ θ のエントロピーを計算することになる。しかし、これにはデータの期待値を求める際のモンテカルロ的なサンプリング、さらにはエントロピーの計算のために近似解 $Q^*(\theta)$ を再計算する必要があり計算量が多くなる問題がある。そこで本章では、式(2.17)から直接期待自由エネルギー G を求めるのではなく、行動先のエントロピー $H(\theta)$ が最も高い行動を選択することにした。これは図3.3からも分かるように、もともとのパラメータの曖昧さ、すなわちエントロピー $H(\theta)$ が大きいほど、データ D を得たときに曖昧さ

が大きく減少すると考えたためである。本章では、計測モデルのパラメータをロボットがサンプル効率の良いデータ収集によって学習することが目的である。したがって、計測モデルのパラメータ θ についての曖昧さ、すなわちエントロピー $H(\theta)$ が大きいマスでは、そのマスにおけるデータのサンプル数が少なく、計測モデルの学習が不十分である可能性が高い。逆に、エントロピー $H(\theta)$ が小さいマスでは、そのマスで十分にデータを得られて計測モデルの学習ができている可能性が高い。

以上をまとめたものを Algorithm1 に示す。

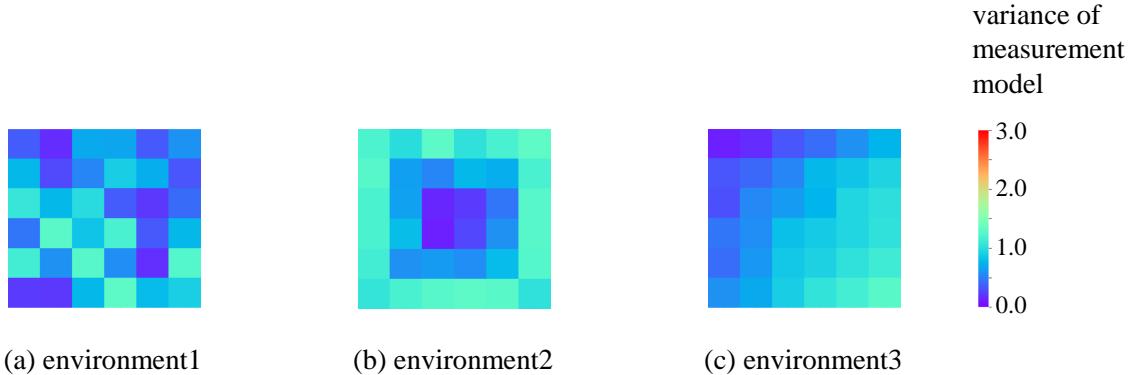
Algorithm 1

```
N ← number of observed value for each step
F ← free energy defined Eq.(2.4)
D = {x1, x2, ..., xN} ← data for each step
Initialize prior probability distribution P( $\theta$ )
Receive initial observation data D
for Step = 1, M do
    Select action  $a$  with the highest  $H(\theta)$  in the next state
    observe  $D$ 
     $Q^*(\theta) = \arg \min_{Q(\theta)} F$  following Eq.(2.7)
     $H(\theta) = E_{Q(\theta)}[-\ln Q(\theta)]$ 
end for
```

3.3 実験方法

提案手法が計測モデルの学習においてサンプル効率の良い好奇心アルゴリズムであるかを検証するために、図 3.1 に示す環境下で実験を行う。1 ステップにつきロボットは上下左右いずれかに 1 マス移動か停止を選択する。壁に向かって進む場合はその場に留まるが 1 ステップとして数える。移動先のマスに落ちている石ころの数を毎ステップ計測するものとする。計測モデルの学習は Algorithm1 に示す通りである。ただし、データ D に含まれる計測値 x は $N = 3$ とした。

図 3.5 に示すように各マスの計測モデルが異なる 3 つの環境で実験を行った。図 3.5 に示すマスの色は各マスにおける計測モデル（ガウス分布）の分散の大きさを表す。各マスにおける計測モデルの平均は 3 つの環境で同一であり、ランダムに設定した。図 3.5(a) は各マスの計測モデルの分散がランダムな環境、図 3.5(b) は部屋の壁に近づくにつれ計測モデルの

Fig. 3.5: Experiment environment in 6×6 room

分散が大きくなる環境、図 3.5(c) は部屋の右下に近づくほど計測モデルの分散が大きくなる環境を想定した。Algorithm1 に示した提案手法の有効性を確かめるため、ロボットがランダムに行動を選択した場合と訪問回数の少ないマスへの行動を選択した場合の 2 通りと比較した。図 3.5 に示す異なる 3 つの環境下でそれぞれ 30 回ずつ実験を行った。ただし、1 回の実験につきロボットは 200 ステップ行動する。

3.4 実験結果

図 3.5 に示す 3 つ環境で実験した結果をそれぞれ図 3.6 に示す。横軸はステップ数、縦軸は 1 マスあたりの真の計測モデルとロボットが学習した計測モデル間の KL 情報量である。random はランダム行動、less は訪問回数が少ないマスへの行動、1step, 2step は提案手法であり、それぞれ 1 ステップ先、2 ステップ先のエントロピー $H(\theta)$ が最も高いマスへ行動した結果である。

真の計測モデルとロボットが学習した計測モデル間の 1 マスあたりの KL 情報量が 0.25 以下になるのに必要なサンプル数を図 3.7 に示す。

図 3.5(a) の環境下において、1~200 ステップの間にロボットが学習した計測モデルの分散の期待値について、ランダム行動の場合を図 3.8(a)、提案手法 (1step) の場合を図 3.8(b) に示す。

図 3.5(b) の環境下において、1~200 ステップの間にロボットが学習した計測モデルの分散の期待値について、ランダム行動の場合を図 3.9(a)、提案手法 (1step) の場合を図 3.9(b) に示す。

図3.5(c)の環境下において、1~200ステップの間にロボットが学習した計測モデルの分散の期待値について、ランダム行動の場合を図3.10(a)、提案手法(1step)の場合を図3.10(b)に示す。

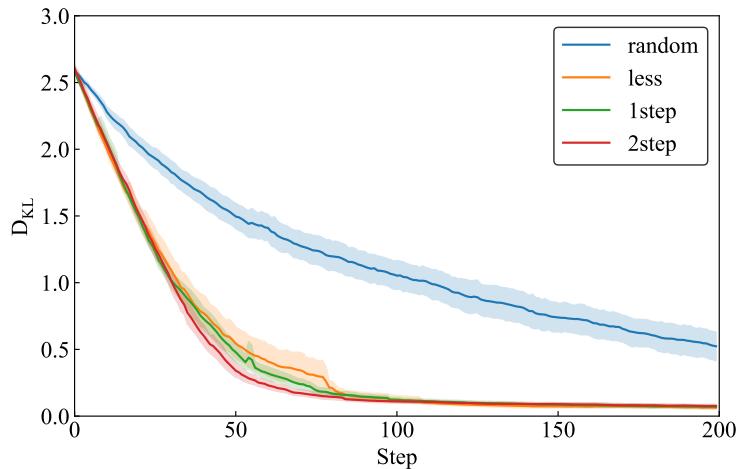
3.5 考察

図3.7より、図3.5に示す3つ環境すべてにおいて、提案手法がランダム行動および訪問回数が少ないマスへ行動した場合よりも少ないサンプルで計測モデルの学習ができていることがわかる。また、1ステップ先よりも2ステップ先の予測の曖昧さに基づいて行動した方がサンプル効率が良いことが分かる。

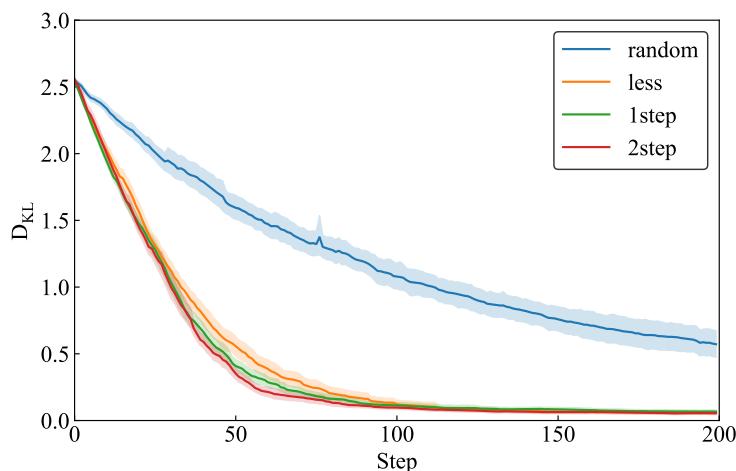
訪問回数に基づく行動よりも提案手法、すなわち予測の曖昧さに基づく行動のほうがサンプル効率がよい理由としては、マスごとに計測モデルが異なるため、マスごとに計測モデルの学習に必要なサンプル数が異なるためであると考えられる。もともとの計測モデルの分散が小さいマスでは、毎回似たようなデータが得られ、計測モデルの学習に必要なサンプル数は相対的に少なくなる。一方、もともとの計測モデルの分散が大きいマスでは、計測モデルの学習に必要なサンプル数は相対的に多くなる。計測モデルの学習が進むと、そのマスに対する予測の曖昧さは減少していき、ロボットはまだ学習が十分でなく予測の曖昧さが大きいマスへ移動することになる。しかし、訪問回数に基づく行動では、計測モデルの学習の進み具合に関係なく行動する。そのためすでに十分に学習ができるいるマスにおいて必要以上にデータを収集する場合があると考えられる。

3.6 おわりに

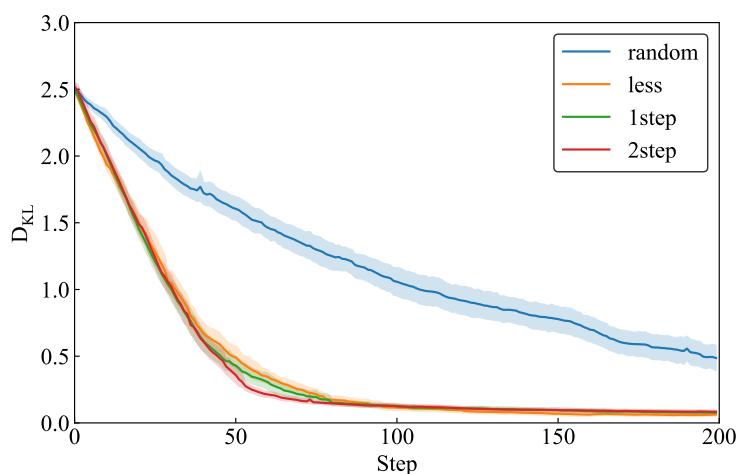
本章では単純なタスク学習におけるロボットの好奇心アルゴリズムを提案した。シミュレーション実験により、提案手法がランダム行動や訪問回数に基づく行動よりもサンプル効率良くデータ収集を行えていることを示した。



(a) Environment1

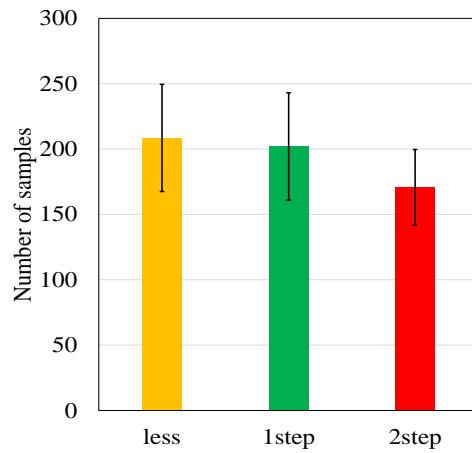


(b) Environment2

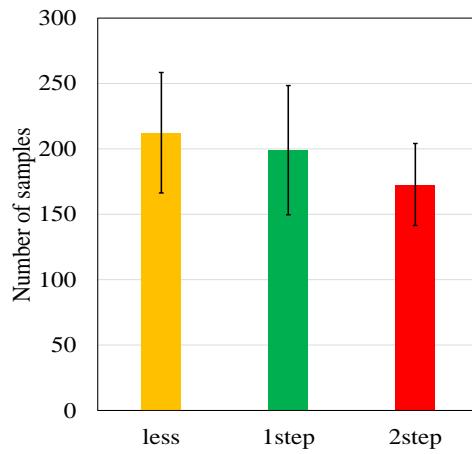


(c) Environment3

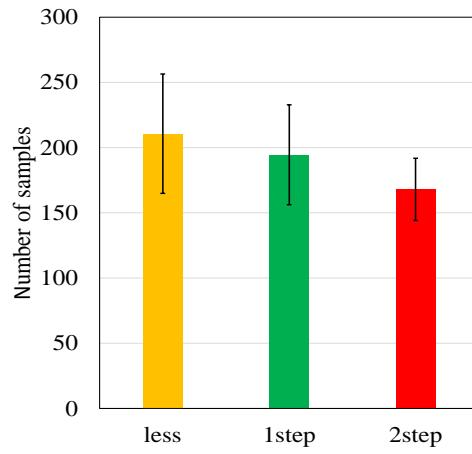
Fig. 3.6: Change of KL divergence per cell between the true measurement model and the measurement model under training



(a) Environment1



(b) Environment2



(c) Environment3

Fig. 3.7: Number of samples required for KL divergence per cell between the true measurement model and the measurement model under training to be less than 0.25

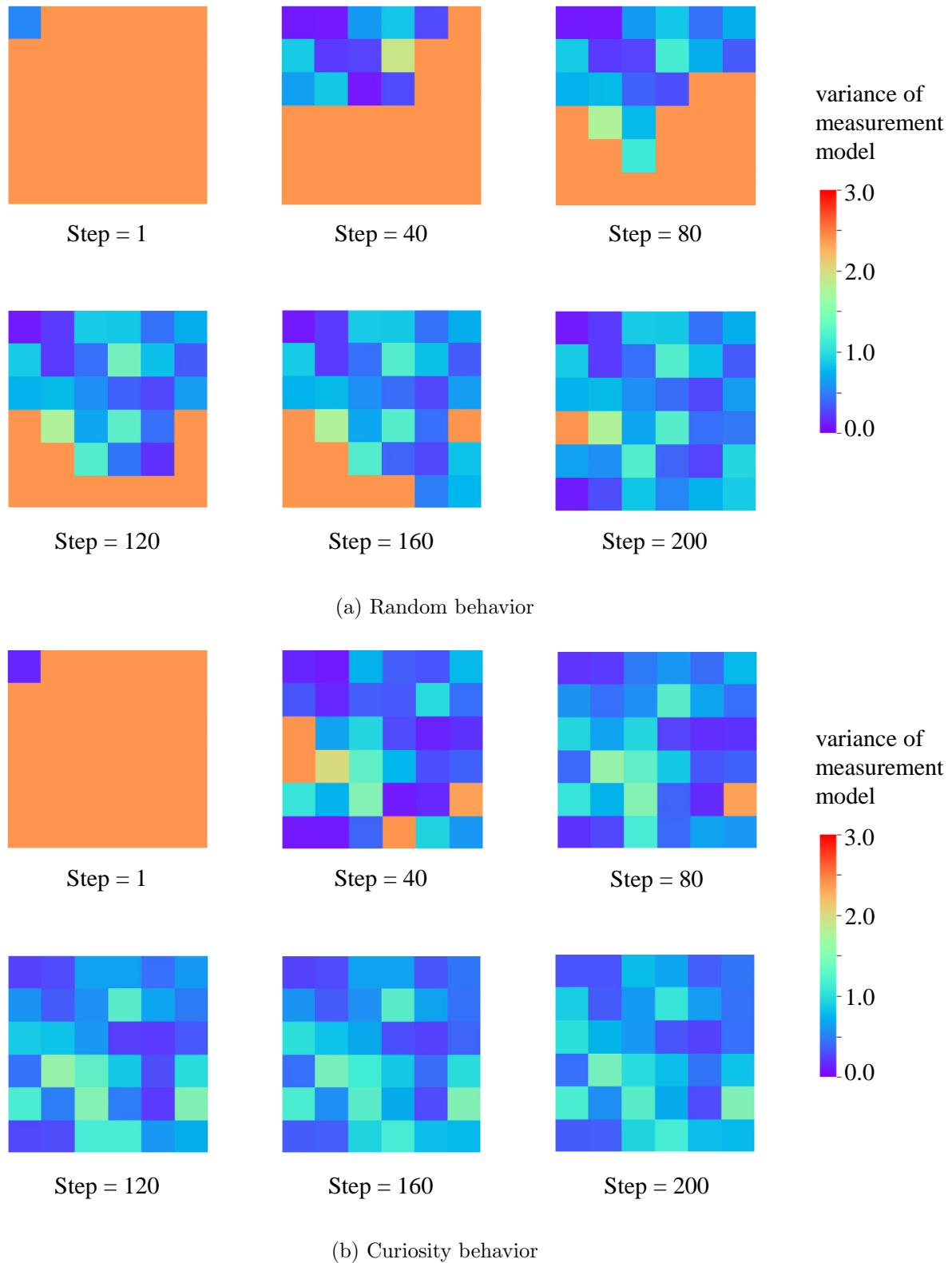


Fig. 3.8: Change of the variance of the measurement model in each cell in environment1

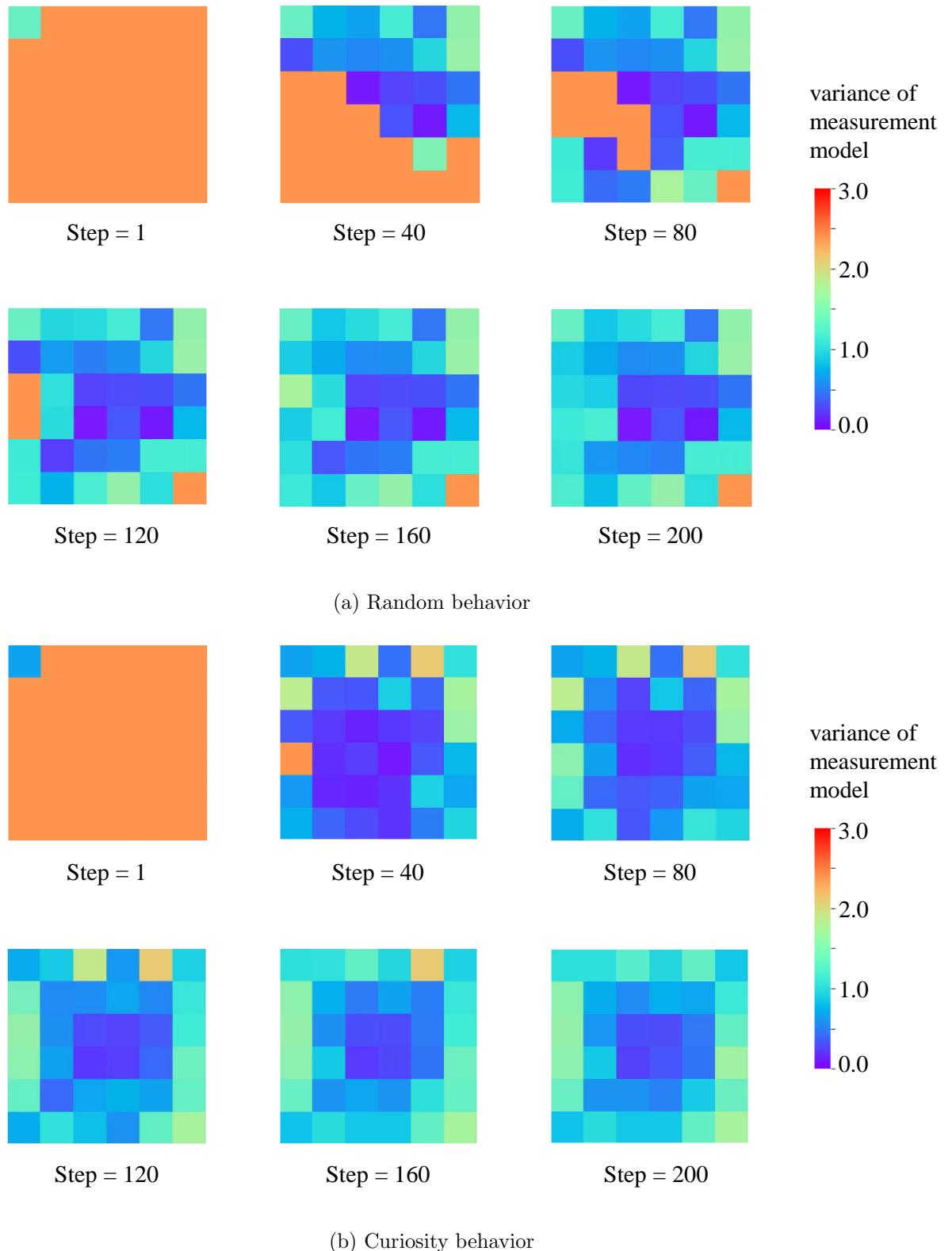


Fig. 3.9: Change of the variance of the measurement model in each cell in environment2

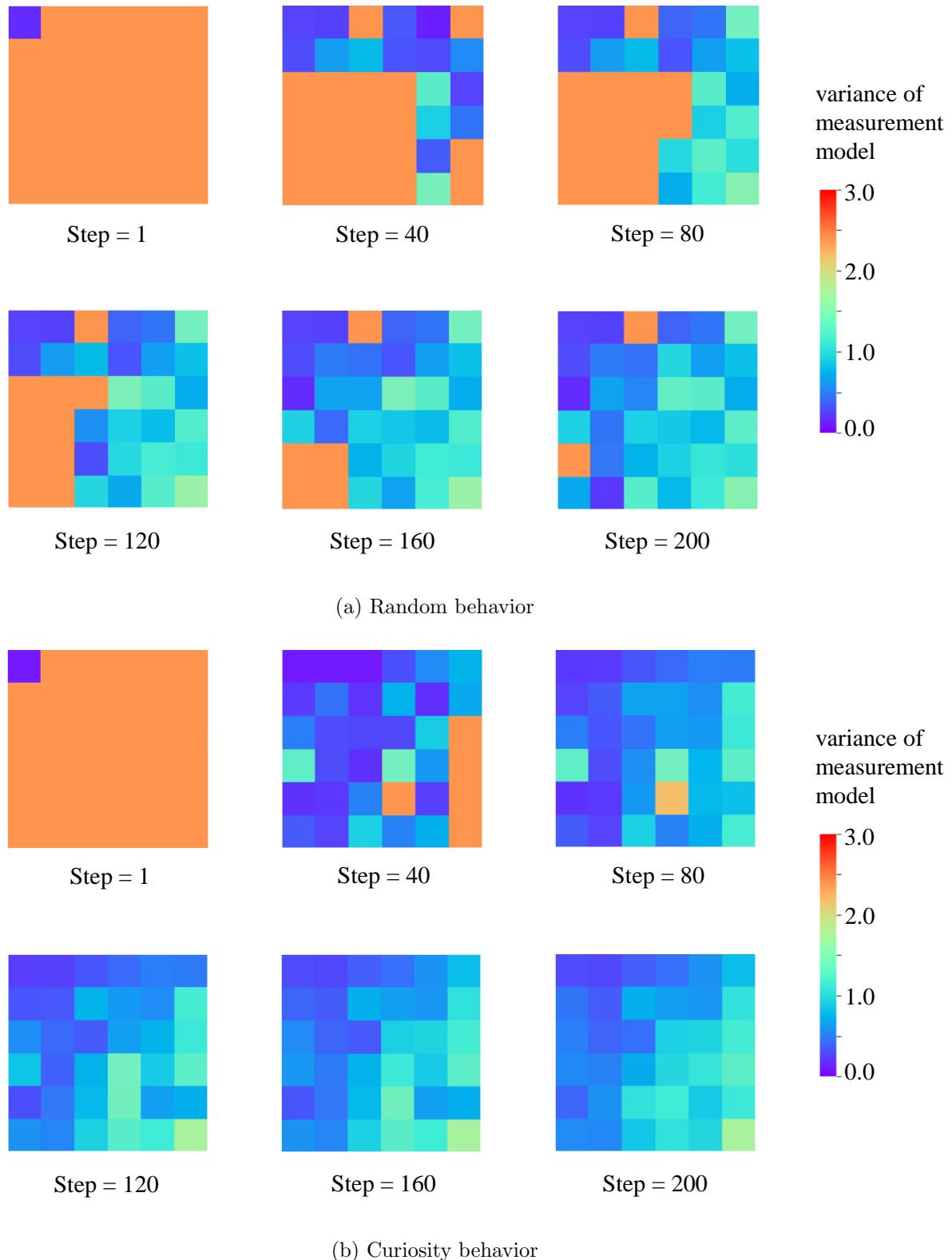


Fig. 3.10: Change of the variance of the measurement model in each cell in environment3

第4章

複雑なタスク学習におけるロボットの
好奇心アルゴリズムの提案と検証実験

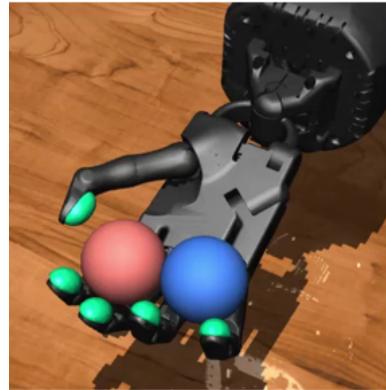


Fig. 4.1: Two balls on the robot hand

4.1 はじめに

本章では、複雑なタスク学習において、サンプル効率の良いデータ収集のための好奇心アルゴリズムを提案し、その有効性をシミュレーション実験により検証する。

本章では、ロボットが外界のモデルを学習することをタスクとする。外界のモデルとは、ロボットが行動したときに周囲の環境がどのように変化するかのモデルである。例えば、図4.1に示すように、ロボットハンドの上にボールを乗せた状態からロボットハンドの指を動かしたときに、ボールの位置や速度などの環境の状態の変化を予測するモデルである。ロボットは外界のモデルの事前知識がない状態からスタートし、試行錯誤によって収集したデータを使って外界のモデルを学習していく。

4.2 好奇心アルゴリズム

行動空間を A 、状態空間を S とすると、外界のモデルは $S \times A \rightarrow S$ で表される。本研究では、外界のモデルの学習にはニューラルネットワークを用いる。

4.2.1 Bayesian Neural Network (BNN)

現在、主流となっているディープラーニングの問題点として過学習[54]、ノイズに対する脆弱性[55]、過度な自信を持った予測[56]などが挙げられる。これらの問題を解決できる手法としてニューラルネットワークにベイズ的手法を取り入れたベイジアンニューラルネットワーク(BNN; Bayesian Neural Network)[57]が注目されている。図4.2に示すように、従

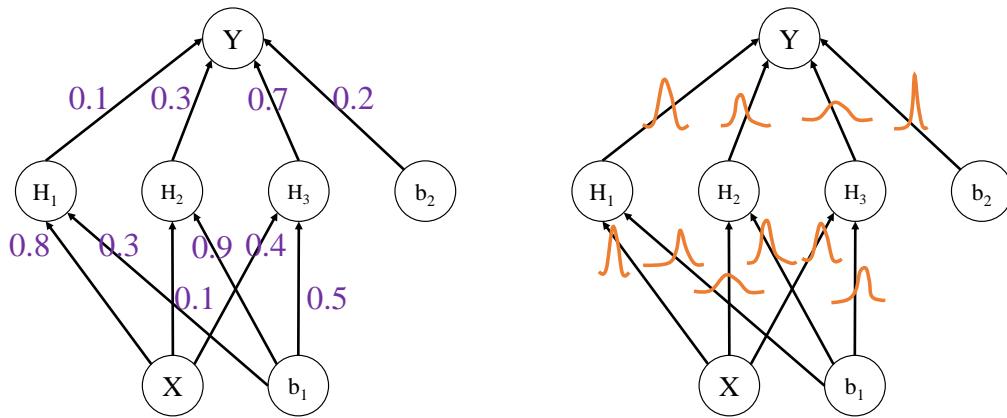


Fig. 4.2: Left(classical neural network): each weight has a fixed value. Right(bayesian neural network): each weight is assigned a distribution

来のニューラルネットワークでは実数値で表現されていた重みに対し、BNN では重みに確率分布を仮定する。これにより、予測を確率分布として得ることが可能となり、予測の不確かさの評価が可能となる。

ニューラルネットワークの予測結果を不確実なものにする要素として、以下の 2 種類が考えられる [58][59]。

- 偶然の不確実性 (Aleatory uncertainty)
 - 入力データ自体があいまいなために発生する不確実性
 - ノイズが大きすぎるなど避けられない不確実性でいくらモデルを訓練しても改善できない
- 認識の不確実性 (Epistemic uncertainty)
 - 入力データに対する予測能力が不足しているために発生する不確実性
 - 入力データに対する訓練不足が原因のため、再学習することで改善できる

最終的な予測結果の不確実性はこれらの不確実性の総和となる。BNN ではこれらの不確実性を分解して測定することができる。

BNN の重みを θ 、その事前確率分布を $P(\theta)$ とする。一般的に、事前確率分布 $P(\theta)$ にはガウス分布が用いられることが多い。入力 $x \in \mathbb{R}$ 、出力 $y \in \mathbb{R}$ であるデータ $D = \{x_i, y_i\}$

が与えられたとき、ベイズ推定の目的は、事後確率分布 $P(\theta|D)$ を求めることである。しかし、2.2.1節で述べたように、ニューラルネットワークのような複雑なモデルでは事後確率分布 $P(\theta|D)$ を直接求めることは困難である。そこでBNNでは、変分ベイズ法を用いて事後確率分布 $P(\theta|D)$ の近似解 $Q^*(\theta; \phi)$ を求める。ただし、 ϕ は重み θ の確率分布のパラメータである。重みの確率分布がガウス分布の場合、 $\phi = \{\mu, \sigma\}$ である。

近似解 $Q^*(\theta; \phi)$ は、以下に示す自由エネルギー F の最小化によって求まる。

$$F = D_{KL}[Q(\theta; \phi) || P(\theta)] - E_{Q(\theta; \phi)} \ln P(D|\theta) \quad (4.1)$$

上式はニューラルネットの損失関数に相当する。式(4.1)の第2項はデータ D に依存し、ニューラルネットの順伝播の最後に評価される。第2項は以下のように計算される[57]。

$$E_{Q(\theta; \phi)} \ln P(D|\theta) = \frac{1}{N} \sum_{i=1}^N \ln P(D|\theta_i) \quad (4.2)$$

ただし、 N は確率分布 $Q(\theta; \phi)$ からサンプリングした θ の数である。

式(4.1)の第1項はデータ D に依存せず、ニューラルネットのレイヤーごとに評価される。逆伝播の間、 $\phi = \{\mu, \sigma\}$ の勾配は、Adam[60]のようなオプティマイザによって更新されるために、誤差逆伝播法によって計算される。しかし、確率分布が間にしているため、誤差逆伝播法をそれより下に適用することができない。この問題を解決するために Reparameterization Trick[61] という手法が用いられる。この手法は、 $\epsilon \sim \mathcal{N}(0, I)$ にてノイズを発生させ、サンプリングした ϵ を使って重みを $\theta = \mu + \sigma \odot \epsilon$ と1つの値に定める。ただし、 \odot は要素毎の積を意味する。また、 σ が常に正であることを保証するために、 σ の代わりに $\rho \in \mathbb{R}$ でネットワークをパラメータ化し、ソフトプラス関数によって $\sigma = \log(1 + \exp(\rho))$ とする[61]。このように Reparameterization Trick を用いて自由エネルギー F を最適化する方法は、確率的勾配変分ベイズ (SGVB:Stochastic Gradient Variational Bayes) [61] と呼ばれる。

重みの不確かさから生じる予測の不確かさは認識の不確実性 (Epistemic uncertainty) に相当し、より多くのデータを得れば減少する。その結果、認識の不確かさは、訓練データがないまたは少ない領域では高く、訓練データが多い領域では低くなる。認識的不確かさは、式(4.1)の第1項によってカバーされる。訓練データの固有のノイズに由来する不確実性は偶然の不確実性 (Aleatory uncertainty) に相当し、データが増えても減らすことはできない。偶然の不確実性は、式(4.1)の第2項中の尤度関数によってカバーされる。

ここで、1次元の回帰問題に BNN を適用してみる。教師数は 12 個、中間層は 2 層、各層の素子数は 20 個、活性化関数は ReLU (Rectified Linear Unit) で全結合とした。ReLU 関数は一般に最も高い精度が得られ、高速な処理が実現できる [62]。

学習前後における BNN の予測結果をそれぞれ図 4.3, 4.4 に示す。学習後の BNN の予測の曖昧さは、データが多い領域よりもデータが少ない領域で大きくなっていることが分かる。

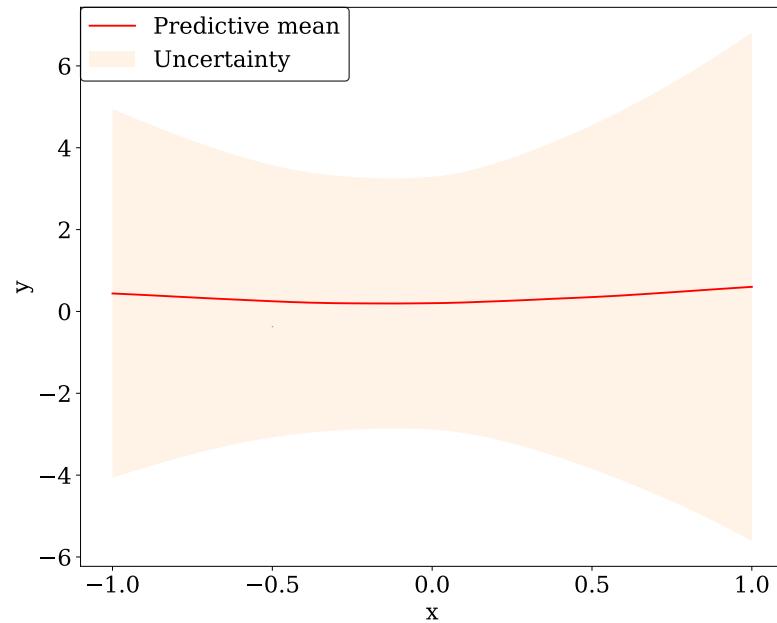


Fig. 4.3: BNN output before learning

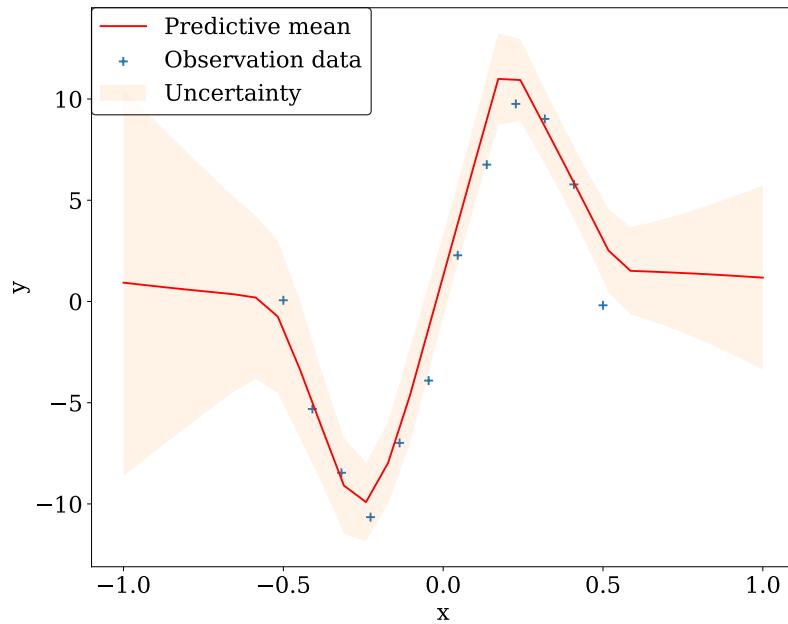


Fig. 4.4: BNN output after learning

4.2.2 提案手法：BNN + Curiosity

3.2.3 節の議論と同様、本章でも直接期待自由エネルギー G を求めて、それが最小となる行動を選択するのではなく、次の状態 s_{t+1} に対する予測の曖昧さが最も高い行動を選択するものとする。BNN の順伝播の際に、重みの確率分布から $Q(\theta; \phi)$ からランダムに N 個サンプリングした θ から N 通りの BNN の出力が得られる。この N 通りの BNN の出力から、次の状態 s_{t+1} に対する予測の平均と分散（曖昧さ）が統計的に求まる。

第3章ではロボットの行動空間が離散的であり行動ごとの予測の曖昧さを計算することができた。しかし、本章ではロボットの行動空間は連続的であり、すべての行動に対して予測の曖昧さを計算することは困難である。そこで、本章では毎ステップごとに k 個の行動をランダムに生成し、その中で次の状態 s_{t+1} の予測の曖昧さが最も大きい行動を選択する。以上の計算手順をまとめたものを Algorithm2 に示す。

Algorithm 2 BNN+Curiosity

- 1: $\theta \leftarrow$ bayesian neural network weights
 - 2: $\phi \leftarrow$ parameters of bayesian neural network weights
 - 3: $N \leftarrow$ number of θ sampled from probability distribution $Q(\theta; \phi)$
 - 4: $F \leftarrow$ free energy defined Eq.(4.1)
 - 5: $D \leftarrow$ observation data
 - 6: Initialize the prior probability distribution $P(\theta)$
 - 7: **for** $t = 1, T$ **do**
 - 8: Randomly generate k actions a
 - 9: N samples drawn according to $\theta \sim Q(\theta; \phi)$
 - 10: Compute the mean and variance of the neural network predictions using the sampled θ
 - 11: Select action a with the highest variance of the neural network prediction
 - 12: observe D
 - 13: Sample $\epsilon \sim \mathcal{N}(0, I)$
 - 14: Set network parameters to $\theta = \mu + \sigma\epsilon$
 - 15: Minimize $D_{KL}[Q(\theta; \phi_t) || P(\theta)] - E_{Q(\theta; \phi)} \ln P(D|\theta)$ following Eq.4.1, leading to updated posterior $Q(\theta; \phi_{t+1})$
 - 16: **end for**
-

4.3 実験方法

本節以降のシミュレーションは物理シミュレータである MuJoCo[63] を使用した。本節では、図 4.1 に示す環境においてロボットが外界のモデルを学習することをタスクとした。図 4.1 に示すロボットハンドは Shadow Dexterous Hand[64] である。Shadow Dexterous Hand は、5 指で 20 degrees of mobility (DOM) /24 degrees of freedom (DOF) である。人差し指、中指、薬指は 3DOM/4DOF であり、第 1, 2, 3 関節の屈伸と、指付け根 (MP 関節) の指の開きが可能である。親指と小指は 5DOM/5DOF であり、掌内部に小指を親指と対抗させるための関節がある。本実験では、各指の付け根の屈伸のみ（計 5 自由度）を可動できるものとした。外界のモデルは、ロボットが指を動かしたときの 2 つのボールのそれぞれの位置 x, y (地面と水平) を予測するモデルとした。ボールの高さ z については、今回のシミュレーションでは x, y に比べ大きく変化しないことと、手のひらのうえでボールを転がしている間はボールの位置 x, y が決まればボールの位置 z も一意に決まると考え除いた。

外界のモデルは 4.2.1 節で述べた BNN によって学習した。BNN の入力は、状態入力が現在のロボットの関節角度 (5 次元) と 2 つのボールの位置 x_t, y_t と速度 \dot{x}_t, \dot{y}_t 、行動入力が各関節角度の変化量 (5 次元) とした。BNN の出力は、次の時刻における 2 つのボールの位置 x_{t+1}, y_{t+1} とした。

1 回の実験は 2000 ステップ ($T = 2000$) とした。ただし、1 ステップは 0.1 秒であり、ロボットは毎ステップ環境の状態を観測し、行動を選択する。実験途中でボールが手から落ちた場合は図 4.1 に示す初期状態から再開した。2000 ステップ終了後、学習した BNN を使って、ランダムに 500 ステップ行動したときの 2 つのボールの位置 x, y の予測誤差を求めた。Algorithm2 に示した提案手法の有効性を示すために、2000 ステップ学習中のロボットの行動をランダムにした場合と比較した。

実験に用いたハイパーパラメータを表 4.1 に示す。活性化関数には ReLU 関数を用いた。また、ニューラルネットのオプティマイザには Adam[60] を用いた。Adam は確率的勾配下降法 (stochastic gradient descent) [65] を用いており、データをまとめて投入するバッチ学習 (batch learning) と比べて高速でメモリ効率も良い。さらに、データ点ごとの勾配を計算するために局所最適解 (local optimum) を避けやすいという利点がある。

毎ステップ、ランダムに生成する行動数は $k = 10$ とした。BNN の予測の平均、分散の計算に必要な確率分布 $Q(\theta; \phi)$ からのサンプリング数は $N = 100$ とした。

Table 4.1: Hyperparameters during experiment

Parameter	Value
number of hidden layers	2
number of hidden units for layers1	100
number of hidden units for layers2	100
optimizer	Adam
learning rate	5e-3
activation function	ReLU

4.4 実験結果

2000 ステップ終了後、学習した BNN による、ランダムに 500 ステップ行動したときの 2 つのボールの位置 x, y の予測誤差を測る実験を 20 回行ったときの結果を図 4.5 に示す。図中の random はランダム行動、1step, 2step は提案手法であり、それぞれ 1 ステップ先、2 ステップ先の BNN の予測の分散（曖昧さ）が最も高い行動を選択した場合の結果である。ただし、2 ステップ先を予測する際に必要な時刻 $t + 1$ におけるボールの位置と速度は、1 ステップ先の BNN の出力の平均値から求めた。

提案手法 (2step) によって 2000 ステップ学習した BNN によるボールの位置 x, y の予測軌跡を図 4.6 に示す。

4.5 考察

図 4.5 より、提案手法によって学習したモデルがランダム行動によって学習したモデルよりも予測誤差が小さいことが分かる。このことから、提案手法がランダム行動と比べてサンプル効率良くデータ収集を行えていることが分かる。これは、提案手法によってロボットが訓練データが少ない領域を積極的に探索し、外界のモデルの学習に必要なサンプルを効率よく集めることができたためと考えられる。前章で行った実験同様、1 ステップ先よりも 2 ステップ先の予測の曖昧さに基づいて行動した方がサンプル効率が良いことが分かる。

4.6 おわりに

本章では複雑なタスク学習におけるロボットの好奇心アルゴリズムを提案した。シミュレーション実験により、提案手法がランダム行動よりもサンプル効率良くデータ収集を行えていることを示した。

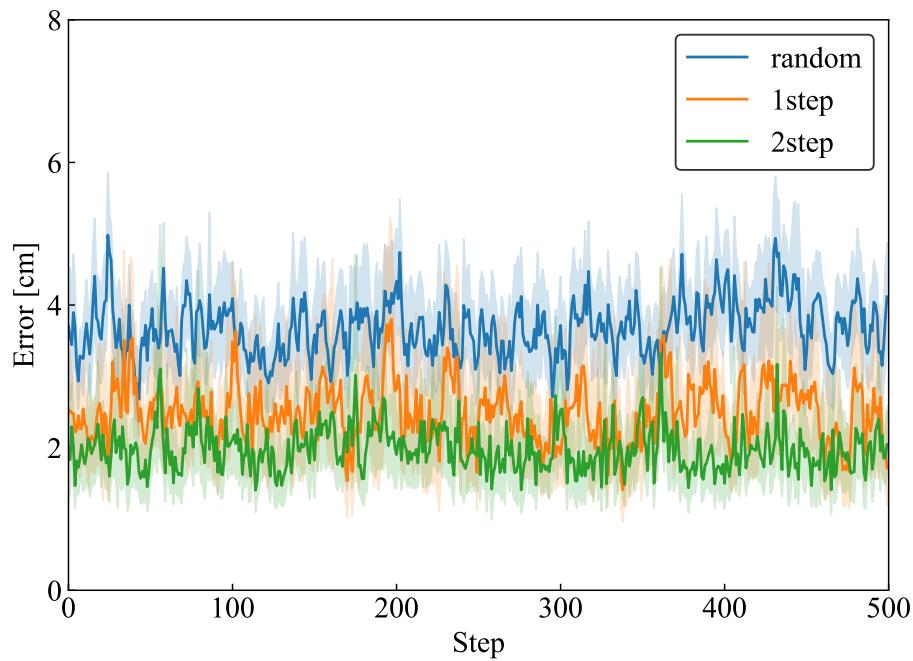


Fig. 4.5: Prediction error of the position of two balls after 2000 step learning

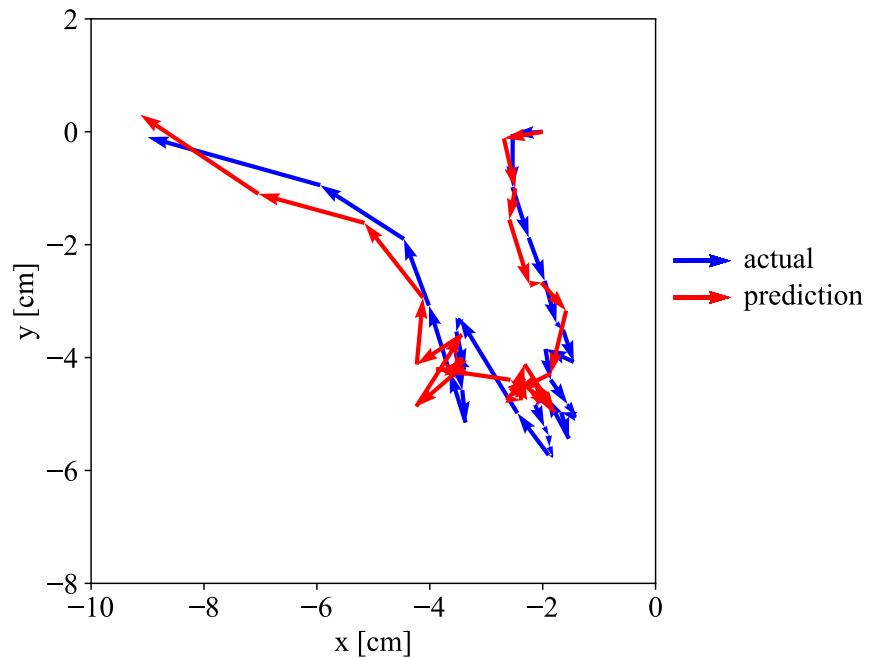


Fig. 4.6: Prediction of ball position by the proposed method after 2000-step learning

第5章

深層強化学習におけるロボットの
好奇心アルゴリズムの提案と検証実験

5.1 はじめに

本章では、深層強化学習によるロボットのタスク学習において、サンプル効率の良いデータ収集のための好奇心アルゴリズムを提案し、その有効性をシミュレーション実験により検証する。

5.2 深層強化学習

深層強化学習は強化学習 [66] と深層学習を組み合わせた機械学習の手法である。

強化学習では、学習を行なうエージェント（本研究ではロボット）が環境から状態を観測し、その観測結果を元に行動を選択し、その行動に対する報酬をもらう。エージェントは環境から得られる報酬が最も多くなるような方策（policy）を学習する。

強化学習の代表的な手法として、Q学習 [67] がある。Q学習では、各状態において可能な行動の中で最も行動価値 Q の値が高い行動をとるように学習を行う。行動価値 Q_{s_t, a_t}^π とは、方策 π のもと、状態 s_t において行動 a_t をとった場合に将来もらえると期待される割引報酬和であり、次式で表される。

$$Q^\pi(s_t, a_t) = E_\pi \left[\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i} | s_t, a_t \right] \quad (5.1)$$

ただし、 γ は割引率であり、将来もらえる報酬をどれだけ割り引いて考えるかのパラメータである。例えば、 $\gamma = 0$ なら、直後の報酬のみを考えることになり、 $\gamma = 1$ なら、将来もらえる報酬をすべてそのまま現在の価値として考慮に入れることになる。

行動価値 $Q^\pi(s_t, a_t)$ はベルマン方程式によって一つ先の時刻 $t+1$ の行動価値 $Q^\pi(s_{t+1}, a_{t+1})$ を使って次のように書くことができる。

$$Q^\pi(s_t, a_t) = E_{r_t, s_{t+1} \sim T} [r(s_t, a_t) + \gamma E_{a_{t+1} \sim \pi} [Q^\pi(s_{t+1}, a_{t+1})]] \quad (5.2)$$

ただし、上式中の T は環境の状態遷移確率モデルである。しかし、現実世界の問題において、環境の状態遷移モデル T が既知であるケース、またそれらがうまくモデル化できるケースは少ない。そのため、強化学習では一般的に環境のモデルを使わないモデルフリーの手法（Q学習もその一つ）がよく用いられる。

Q学習では、エージェントにとって未知である最適行動価値関数を学習していく。具体的な学習の流れは次のようになる。まず環境が取りうる全ての状態 s と選択できる全ての行動

a の組に対しての暫定的な行動価値 $Q(s, a)$ を保持することのできる Q-table を用意し、任意の値で初期化する。時刻 t において状態 s_t 下のエージェントが行動 a_t を選択し、報酬 r_t を受け取り、次の状態 s_{t+1} に遷移する。ここで得られた経験 (s_t, a_t, r_t, s_{t+1}) を用いて、行動価値 $Q(s, a)$ を以下のように更新する。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (5.3)$$

ただし、 α は学習率である。

Q-table では環境が取りうる全て状態と選択できる全ての行動に対して行動価値を計算する必要がある。したがって、画像のような状態が高次元である場合やロボットの関節角度のような状態が連続値の場合に Q-table が膨大になり、計算に多くの時間がかかるてしまう問題がある（次元の呪い）[68][69]。この問題を解決したのが Deep Q-Network (DQN) [70] である。DQN では Q-table の代わりに状態を入力とし、それぞれの行動に対する行動価値を出力とするネットワークを生成し関数近似を行なっている。しかし、DQN は高次元の状態空間の問題を解決するが、離散的で低次元の行動空間しか扱えない。DQN を連続的な行動空間に適応させるための明白なアプローチは、行動空間を離散化することである。しかし、行動の数は自由度の数とともに指数関数的に増加してしまい、ここでも次元の呪いが問題となる。また、行動空間の離散化によって、多くの問題を解決するために不可欠な行動の情報を不要に破棄することになる。この問題を解決したのが Deep Deterministic Policy Gradients (DDPG) [71] である。

5.2.1 Deep Deterministic Policy Gradient (DDPG)

DDPG の元となっているのは強化学習における Actor-Critic [72] という手法である。この手法では、エージェントは図 5.1 に示すように Actor と Critic から構成される。Actor は方策を学習し、Critic は価値関数を学習し、Actor の行動を評価する。DDPG では、Actor は、状態 s を入力とし、行動（連続値） a を出力するネットワークであり、Critic は、状態 s と行動 a を入力とし、行動価値 Q を出力するネットワークである。

DDPG は Q 学習と同じく off-policy の強化学習の手法であり、次式に示すように、行動価値 Q が最大になるような行動 μ を選ぶグリーディ方策である。

$$\mu = \arg \max_a Q(s, a) \quad (5.4)$$

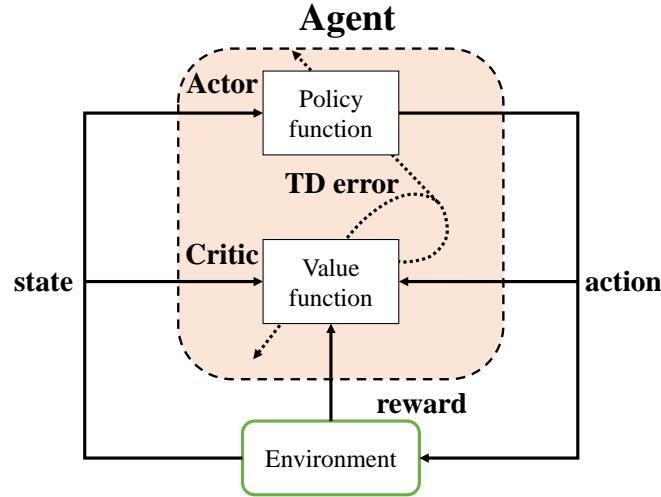


Fig. 5.1: Actor-Critic

また、DDPGは状態 s の入力に対して行動 a が一意に決まる決定論的方策である。したがって、式(5.2)は以下のようになる。

$$Q^\mu(s_t, a_t) = E_{r_t, s_{t+1} \sim T} [r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu_{t+1})] \quad (5.5)$$

Critic ネットワークのパラメータ θ^Q は以下の損失関数 L を最小化することで更新される。

$$L(\theta^Q) = \frac{1}{N} \sum_i (y_i - Q(s_t, a_t | \theta^Q))^2 \quad (5.6)$$

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'}) | \theta^{Q'}) \quad (5.7)$$

ただし、 $\theta^{Q'}$, $\theta^{\mu'}$ はそれぞれ、Critic と Actor のターゲットネットワークのパラメータ、 γ は割引率を表す。ターゲットネットワークは、DQNにおいてニューラルネットによる行動価値 Q の学習を安定化するために使われる。行動価値 Q の学習は式(5.6)の最小化によって行われる。このとき、式(5.7)は式(5.6)中の行動価値 Q に対する教師データである。したがって、式(5.7)中の行動価値 Q' は損失関数の微分対象外である。しかし、行動価値 Q は学習の度に更新されて行くので、教師データもその度変動する。この時、毎回教師データが変わってしまうと学習が不安定になる。そこで DQN では、一定期間式(5.7)の行動価値 Q' を更新せず（つまり教師データを固定する）、式(5.6)中の行動価値 Q のみ更新し、一定期間後、式(5.7)の行動価値 Q' を式(5.6)中の行動価値 Q と同期する。これは Fixed target

Q-Network [73], または Hard update と呼ばれる。DDPG では一定期間待つ代わりに、以下のように毎回少しずつターゲットネットワークをメインネットワークに近づいていく。これは Hard update に対し、Soft update と呼ばれる [71]。

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'} \quad (5.8)$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1 - \tau)\theta^{\mu'} \quad (5.9)$$

ただし、 $\tau \ll 1$ 。

Actor ネットワークのパラメータ θ^{μ} は連鎖律により以下の勾配を用いて更新される。

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_i \nabla_{\theta^{\mu}} Q(s, a | \theta^Q) |_{s=s_t, a=\mu(s_t | \theta^{\mu})} \quad (5.10)$$

$$= \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_t, a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s | \theta^{\mu}) |_{s=s_t} \quad (5.11)$$

DDPG ではエージェントの探索を促進するために、次式で示すように Actor によって決定された行動 $\mu(s_t | \theta^{\mu})$ にガウスノイズが追加される。

$$a_t = \mu(s_t | \theta^{\mu}) + \mathcal{N} \quad (5.12)$$

DDPG では DQN と同じく Experience Replay[74] と呼ばれる手法を用いる。Experience Replay とは、行動とその結果を経験として保存しておく、その過去の経験をサンプリングして学習データとする手法である。通常の逐次で学習する方法に比べ、時間的に離れたデータを使用することができ、偏りを抑えた安定した学習が可能となる。具体的には、現在の状態 s_t 、行動 a_t 、行動後の状態 s_{t+1} 、報酬 r_t のデータの組をある一定の数だけメモリに保存しておいて、その中から学習に使うデータをいくつかランダムにサンプリングし、これらを 1 つのバッチデータとしてニューラルネットワークを学習させる。アルゴリズムが安定した動作をするためには、メモリは幅広い経験値を格納するのに十分な大きさでなければならないが、すべての経験値を保持するのは学習が遅くなる可能性がある。一方、非常に最近のデータだけを使うならば、それに合わせてオーバーフィットしてしまい学習が悪化してしまう。

DDPG のアーキテクチャを図 5.2 に示す。DDPG のアルゴリズムを Algorithm3 に示す。

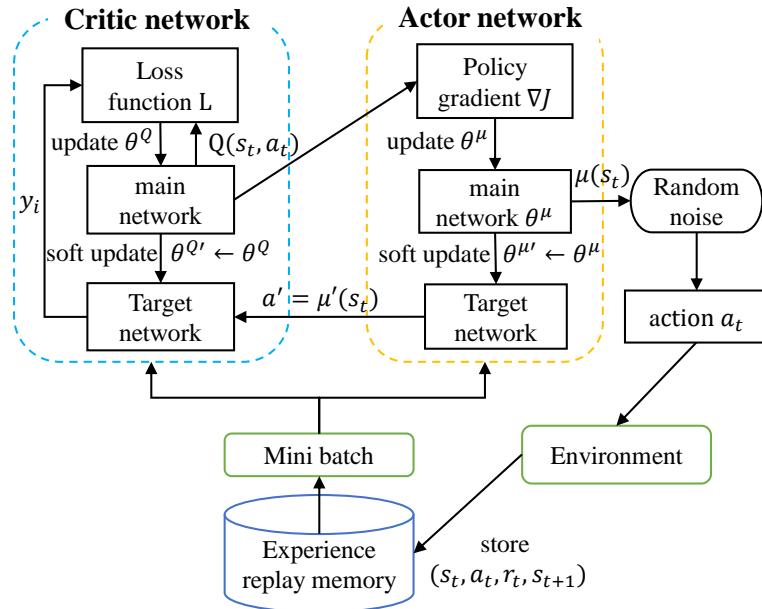


Fig. 5.2: DDPG architecture

Algorithm 3 DDPG

```

Randomly initialize critic network  $Q(s, a|\theta^Q)$  and actor  $\mu(s|\theta^\mu)$  with weights  $\theta^Q$  and  $\theta^\mu$ 
Initialize target network  $Q$  and  $\mu'$  with weights  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$ 
Initialize replay buffer  $R$ 
for episode = 1, M do
    Receive initial observation state  $s_1$ 
    for t = 1, T do
        Select action  $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}$  according to the current policy and exploration noise
        Execute action  $a_t$  and observe reward  $r_t$  and observe new state  $s_{t+1}$ 
        Store transition( $s_t, a_t, r_t, s_{t+1}$ ) in  $R$ 
        Sample a random minibatch of N transition  $(s_i, a_i, r_i, s_{i+1})$  from  $R$ 
        Set  $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$ 
        Update critic by minimizing the loss:  $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$ 
        Update the actor policy using the sampled policy gradient:
         $\nabla_{\theta^\mu} J = \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$ 
        Update the target networks:
         $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$ 
         $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$ 
    end for
end for

```

5.2.2 Soft Actor Critic (SAC)

強化学習では方策が局所最適に陥らないために探索 (Exploration) と活用 (Exploitation) のバランスが重要である。DDPG は常に行動価値 Q を最大化する行動を選択する貪欲方策 (Greedy policy) であり、探索力が弱く局所解に陥りやすい。また、ハイパーパラメータが学習結果に与える影響が大きく、学習が不安定になりやすいという問題が指摘されている [75] [76]。Soft Actor-Critic (SAC) [77] では DDPG に最大エントロピー Q 学習 (Soft-Q 学習) [78] と確率的方策を導入することで DDPG の探索の弱さを克服している。

Q 学習の探索力の弱さを補うために最もよく使われるアプローチは ϵ -Greedy 法 [70] である。これは、 ϵ の確率でランダムな行動によって探索を行い、 $1 - \epsilon$ の確率で活用目的の行動 (Greedy な行動) を行うというものである。これに対して、Soft-Q 学習では強化学習の目的関数自体に方策エントロピー項を組み込むことで探索の弱さをカバーしている。方策エンタロピー $H(\cdot|s_t)$ とは、行動 a の対数選択確率 $\ln \pi(a|s_t)$ の期待値であり、次式で表される。

$$H(\cdot|s_t) = E_{a \sim \pi}[-\ln \pi(a|s_t)] \quad (5.13)$$

強化学習では、現在の状態 s がどのくらい良いのかを計る関数として価値関数 V を考える。方策 π のもとで、状態 s の価値は以下のように定義される。

$$V^\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s] \quad (5.14)$$

Soft-Q 学習における価値関数 V は、通常の価値関数に方策のエントロピー最大化項を加えることで次のようになる。

$$V(s_t) = E_\pi[Q(s_t, a_t) - \alpha \ln \pi(a_t|s_t)] \quad (5.15)$$

ただし、 α は温度パラメータであり、行動価値に対するエントロピー項の相対的な重要度を決定する。

soft 価値関数 (soft value function) のパラメータ ψ は、以下の二乗誤差を最小化することで学習する。

$$J_V(\psi) = E_{s_t \sim D}[\frac{1}{2}(V_\psi(s_t) - E_{\pi_\phi}[Q_\theta(s_t, a_t) - \ln \pi_\phi(a_t|s_t)])^2] \quad (5.16)$$

soft Q 関数のパラメータ θ は、以下の二乗誤差を最小化することで学習する。

$$J_Q(\theta) = E_{s_t, a_t \sim D}[\frac{1}{2}(Q_\theta(s_t, a_t) - r(s_t, a_t) + \gamma E_{s_{t+1} \sim p}[V_{\bar{\psi}}(s_{t+1})])^2] \quad (5.17)$$

ただし、 $\bar{\psi}$ は soft V 関数の target network のパラメータである。target network の更新則には移動平均（moving average）を用いることで学習が安定する [73]。

勾配は以下のようになる。

$$\begin{aligned}\hat{\nabla}_{\theta} J_Q(\theta) &= \nabla_{\theta} Q_{\theta}(s_t, a_t)(Q_{\theta}(s_t, a_t) - r(s_t, a_t) \\ &\quad + \gamma(Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha \ln \pi_{\phi}(a_{t+1}|s_{t+1})))\end{aligned}\tag{5.18}$$

方策関数のパラメータ ϕ は、次式を最小化することで学習する。

$$J_{\pi}(\phi) = E_{s_t \sim D}[D_{KL}[\pi_{\phi}(\cdot|s_t) || \frac{\exp(\frac{1}{\alpha}Q_{\theta}(s_t, \cdot))}{Z_{\theta}(s_t)}]]\tag{5.19}$$

しかし、式 (5.19) には確率分布が入っているため誤差逆伝播法をそれより下に適用できない。4.2.1 節でも述べたように、reparameterization trick を用いることで、確率分布を可微分な確定的関数に置き換える。具体的には、次式のようにパラメータ ϕ のニューラルネットワークを使って置き換える。

$$a_t = f_{\phi}(\epsilon_t; s_t)\tag{5.20}$$

ただし、 ϵ は reparameterization trick で用いられるノイズを表す。

このとき、 Z は ϕ に依存しないので、式 (5.19) は以下のように書き換えられる。

$$J_{\pi}(\phi) = E_{s_t \sim D, \epsilon_t \sim \mathcal{N}}[\alpha \ln \pi_{\phi}(f_{\phi}(\epsilon_t; s_t) - Q_{\theta}(s_t, f_{\phi}(\epsilon_t; s_t)))]\tag{5.21}$$

勾配は、以下のようになる。

$$\hat{\nabla}_{\phi} J_{\pi}(\phi) = \nabla_{\phi} \alpha \ln \pi_{\phi}(a_t|s_t) + (\nabla_{a_t} \alpha \ln \pi_{\phi}(a_t|s_t) - \nabla_{a_t} Q(s_t, a_t)) \nabla_{\phi} f_{\phi}(\epsilon_t; s_t)\tag{5.22}$$

SAC では一般的に、方策のニューラルネットワークは平均 μ と分散 σ を出力する。

SAC のアルゴリズムを Algorithm4 に示す。

Algorithm 4 SAC

```

Initialize parameter vectors  $\psi, \bar{\psi}, \theta, \phi$ 
Initialize replay buffer  $R$ 
for episode = 1, M do
    Receive initial observation state  $s_1$ 
    for t = 1, T do
        Select action  $a_t = \pi_\phi(a_t|s_t)$ 
        Execute action  $a_t$  and observe reward  $r_t$  and observe new state  $s_{t+1}$ 
        Store transition( $s_t, a_t, r_t, s_{t+1}$ ) in  $R$ 
        for each gradient step do
             $\psi \leftarrow \psi - \lambda_V \nabla_V \hat{\nabla}_\psi J_V(\psi)$ 
             $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$  for  $i \in \{1, 2\}$ 
             $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ 
             $\bar{\psi} \leftarrow \tau\psi + \bar{\psi}$ 
        end for
    end for
end for

```

5.3 従来研究

深層強化学習におけるサンプル効率改善のためにエージェントに好奇心をもたせる手法として、 Intrinsic Curiosity Module (ICM)[15] と呼ばれる好奇心ベースの報酬生成手法が最もよく知られている。最近では、深層強化学習と ICM を組み合わせた手法を実環境下のロボットへ応用した研究も存在する [79][80]。ICM では、エージェントによる環境の予測と実際の環境との違い、つまり予測誤差を「好奇心」と見なし、それを外部報酬とは別に内部報酬として利用した強化学習を行っている。エージェントによる探索が十分に行われていない領域では環境の予測誤差が大きくなることを利用し、予測誤差をそのまま内部報酬（好奇心）として用いることで、探索が十分に行われていない領域を積極的に探索できるようにする狙いがある。

ICM のアーキテクチャを図 5.3 に示す。ICM は 2 つのモデルによって構成されている。一つは、逆モデル (inverse dynamics model) である。逆モデルは 2 つのモジュールから構成される。1 つ目のモジュールでは、環境から観測した状態 s_t をニューラルネットワークを利用して、特徴ベクトル $\phi(s_t)$ へ変換する。2 つ目のモジュールでは、時間的に連続する特徴ベクトル $\phi(s_t), \phi(s_{t+1})$ をニューラルネットの入力とし、エージェントが状態 s_t から s_{t+1} へ

遷移するための行動 a_t を予測する。このニューラルネットワークの学習は、次式で定義される関数 g の学習に相当する。

$$\hat{a}_t = g(s_t, s_{t+1}; \theta_I) \quad (5.23)$$

ここで、 \hat{a}_t が逆モデルが出力する行動の予測である。 θ_I はネットワークのパラメータであり、以下に示す最適化によって学習を行う。

$$\min_{\theta_I} L_I(\hat{a}_t, a_t) \quad (5.24)$$

ここで、 L_I は実際に取った行動 a_t と予想した行動 \hat{a}_t との予測誤差を表す損失関数である。

ICM を構成するもう一つのモデルが、順モデル (forward dynamics model) である。このモデルでは、エージェントが時刻 t において観測した状態の特徴ベクトル $\phi(s_t)$ と、その時点で選択した行動 a_t から、遷移した環境の次状態の特徴ベクトル $\phi(s_{t+1})$ をニューラルネットワークを使って予測する。順モデルの出力は次式で表される。

$$\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t; \theta_F) \quad (5.25)$$

ここで、 f は順モデルのニューラルネットワークで表現される関数であり、 θ_F はニューラルネットワークのパラメータである。順モデルでは、次式に示す損失関数 L_F の最適化によって学習を行う。

$$L_F(\phi(s_t), \hat{\phi}(s_{t+1})) = \frac{1}{2} \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2 \quad (5.26)$$

この時、内部報酬 r_t^i は次式で計算される。

$$r_t^i = \frac{\eta}{2} \|(\hat{\phi}(s_{t+1}) - \phi(s_{t+1}))\|_2^2 \quad (5.27)$$

ただし、 $\eta > 0$ はハイパーパラメータであり、好奇心による探索具合を調整する。

ICM では、逆モデルと順モデルを同時に学習させる。次式のように、2つのモデルの損失関数である L_I と L_F のそれぞれに重みを付けた和を最小化するように学習を行う。

$$\min_{\theta_I, \theta_F} = [(1 - \beta)L_I - \beta L_F] \quad (5.28)$$

ただし、 $0 \leq \beta \leq 1$ である。

ICM の生成する内部報酬によって VizDoom[81] や Openai gym[82] などのゲーム環境下で学習がより効率的に進んだことが報告されている [15][83]。一方で、ICM の問題点もいく

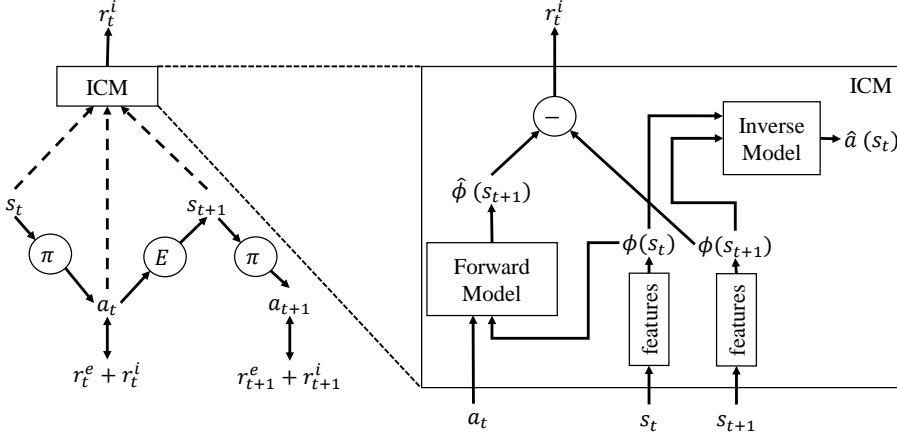


Fig. 5.3: ICM architecture [15]

つか指摘されている。その一つが、The Noisy-TV Problem[83]と呼ばれる問題である。これは、ICMの生成する好奇心の内部報酬のみを用いてゲームの学習を行う際に、ゲーム環境の状態遷移のランダム性が高いと学習が不安定になりやすいという問題である。状態遷移が決定的な場合は、その遷移を十分に経験し学習すれば、その遷移で計算される予測誤差は0に近づいていき、内部報酬も0に近づくのでエージェントは他の遷移を探索するように動機付けられる。一方、状態遷移が確率的な場合は、何度もその遷移を経験してモデルを学習しても、どうしても一定以上の予測誤差が出てしまう。これは、4.2.1節で述べた「偶然の不確実性」(Aleatory uncertainty)に相当する。ICMはその遷移に対して常に一定の内部報酬を生成してしまうため、エージェントはその遷移の探索を続けざるを得なくなってしまい、環境の探索に悪影響をもたらしてしまう。

5.4 好奇心アルゴリズム

ICMでは、「偶然の不確実性」(Aleatory uncertainty)が探索に悪影響を及ぼすことは先に述べた。ロボットをまだ探索していない未知の領域へ促すには「偶然の不確実性」ではなく、「認識の不確実性」(Epistemic uncertainty)を考えるべきである。認識の不確かさは、訓練データがないまたは少ない領域では高く、訓練データが多い領域では低くなる。4.2.1節で述べたように、ベイジアンニューラルネットでは「偶然の不確実性」と「認識の不確実性」を区別して測定できる。認識の不確かさは重みの不確かさから生じる、

本研究では、好奇心としてロボットに対し以下のような内部報酬 r_t^i を与える。

$$r_t^i = \eta \sum_{n=1}^N (H(\theta_{t,n}^F) - H(\theta_{t,n}^F | D_t, a_t)) \quad (5.29)$$

ただし、 $\eta \in \mathbb{R}$ はハイパーパラメータであり、好奇心による探索の具合を調整する。また、 θ^F はベイジアンニューラルネットの重みであり、 N はその総数である。式 (5.29) は行動 a によってデータ D を得たときの重みの不確かさの減少量を表している。したがって、ロボットは重みの不確かさ、すなわち認識の不確かさをできるだけ減らす方向に行動する。

ICMに対する本研究の提案手法の優位性は、BNNによって2種類の曖昧さを区別することにより、偶然の不確実性による動機づけを排除し、認識の不確実性に基づいた動機づけが可能な点である。

DDPG, SAC に式 (5.29) の計算を取り入れたものをそれぞれ Algorithm5, 6 に示す。

Algorithm 5 DDPG+Curiosity

$k \leftarrow$ number of BNN weights
 $D = \{s_t, a_t, s_{t+1}\} \leftarrow$ data for each time
 Initialize BNN weights prior probability distribution $P(\theta^F)$
 Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ

Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
 Initialize replay buffer R
for episode = 1, M **do**
 Receive initial observation state s_1
 for t = 1, T **do**
 Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}$ according to the current policy and exploration noise

 Execute action a_t and observe reward $r_t(s_t, a_t)$ and observe new state s_{t+1}
 Minimize $D_{KL}[Q(\theta^F; \phi_t) || P(\theta^F)] - E_{Q(\theta^F; \phi_t)} \ln P(D_t | \theta^F)$ following Eq.4.1 , leading to updated posterior $Q(\theta^F; \phi_{t+1})$
 $r'_t \leftarrow r_t(s_t, a_t) + \eta \sum_{i=1}^k (H(\theta_{t,i}^F | D_t, a_t) - H(\theta_{t,i}^F))$
 Store transition $(s_t, a_t, r'_t, s_{t+1})$ in R
 Sample a random minibatch of N transition $(s_i, a_i, r'_i, s_{i+1})$ from R
 Set $y_i = r'_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
 Update critic by minimizing the loss: $L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
 Update the actor policy using the sampled policy gradient:
 $\nabla_{\theta^\mu} J = \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$
 Update the target networks:
 $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$
 $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$
 end for
end for

Algorithm 6 SAC+Curiosity

$k \leftarrow$ number of BNN weights
 $D = \{s_t, a_t, s_{t+1}\} \leftarrow$ data for each time
Initialize BNN weights prior probability distribution $P(\theta^F)$
Initialize parameter vectors $\psi, \bar{\psi}, \theta, \phi$
Initialize replay buffer R
for episode = 1, M **do**
 Receive initial observation state s_1
 for t = 1, T **do**
 Select action $a_t = \pi_\phi(a_t|s_t)$
 Execute action a_t and observe reward $r_t(s_t, a_t)$ and observe new state s_{t+1}
 Minimize $D_{KL}[Q(\theta^F; \phi_t) || P(\theta^F)] - E_{Q(\theta^F; \phi)} \ln P(D_t | \theta^F)$ following Eq.4.1 , leading to updated posterior $Q(\theta^F; \phi_{t+1})$
 $r'_t \leftarrow r_t(s_t, a_t) + \eta \sum_{i=1}^k (H(\theta_{t,i}^F | D_t, a_t) - H(\theta_{t,i}^F))$
 Store transition(s_t, a_t, r'_t, s_{t+1}) in R
 for each gradient step **do**
 $\psi \leftarrow \psi - \lambda_V \nabla_V \hat{\nabla}_\psi J_V(\psi)$
 $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
 $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
 $\bar{\psi} \leftarrow \tau\psi + \bar{\psi}$
 end for
 end for
end for

5.5 カメラを用いた物体操作の学習

5.5.1 実験方法

本節では Algorithm5 に示した提案手法の有効性を示すために以下の実験を行った。図 5.4 に示すように、ロボットハンドの上にボールを乗せた状態から、床に定めた目標の位置にボールを落とすことをタスクとした。図 5.4 に示すロボットは 4.3 節と同じく Shadow Dexterous Hand である。ロボットの行動空間は 5 次元であり、5 本指の付け根の屈伸方向の関節角度を動かせるものとした。行動は現在の関節角度からの角度変化量とした。状態空間は 9 次元であり、現在のロボットハンドの関節角度（行動空間と同じ 5 次元）とボールの位置 x_t, y_t (2 次元) および速度 \dot{x}_t, \dot{y}_t (2 次元) とした。ボールの位置はカメラで計測されるものとした。

強化学習に用いる（外部）報酬は、ボールを落とす目標位置 $target_{xy}$ (2 次元) と実際にボールを落とした位置 $object_{xy}$ (2 次元) が近いほど高い報酬になるように、以下のようにした。

$$r = 100 - \|target_{xy} - object_{xy}\| \quad (5.30)$$

1 エピソードは開始から 30 ステップ (3 秒) 経過、あるいはボールが床に落ちた時点で終了とした。初期状態は図 5.4 に示す通りである。1 回の実験は 300 エピソードとし、20 回実験した。Algorithm5 に示した提案手法の有効性を確かめるために、式 (5.29) の好奇心による報酬を入れない場合 ($r_t^i = 0$) と比較した。

DDPG の探索用ノイズは、OrnsteinUhlenbeck 過程 [84] によって生成した。また、DDPG のニューラルネットの学習にはバッチ正則化 (Batch Normalization) [85] を行った。これは、ミニバッチデータに対して、データの平均と分散を計算して入力データを正規化する手法であり、ニューラルネットの勾配収束・爆発を防ぐことで学習の安定化につながる。

実験に用いたハイパーパラメータを表 5.3 に示す。本章の実験では、BNN、アクターネットワーク、クリティックネットワークの中間層、層ごとの素子数、活性化関数、最適化手法 (Optimizer) は同じにした。

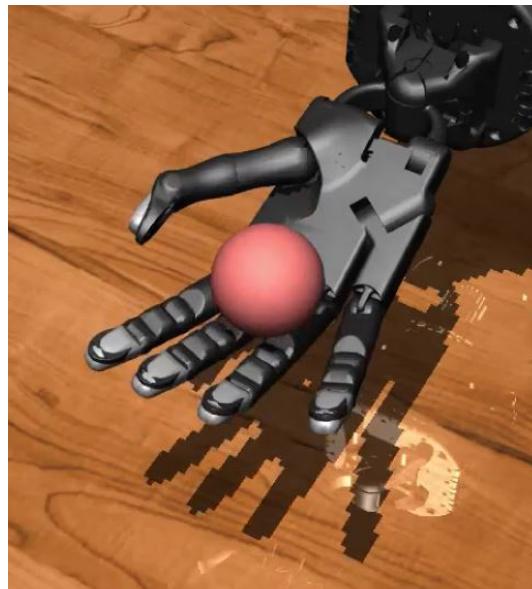


Fig. 5.4: Initial state in object manipulation experiment using camera

Table 5.1: Hyperparameters during experiment

Parameter	Value
number of hidden layers	2
number of hidden units for layers1	100
number of hidden units for layer2	100
minibatch size	64
buffer size	10000
optimizer	Adam[60]
learning rate l_c	5e-4
learning rate l_a	1e-4
learning rate l_f	1e-4
activation function	ReLU

5.5.2 実験結果

実験結果を図 5.5 に示す。横軸はエピソード数、縦軸は報酬である。

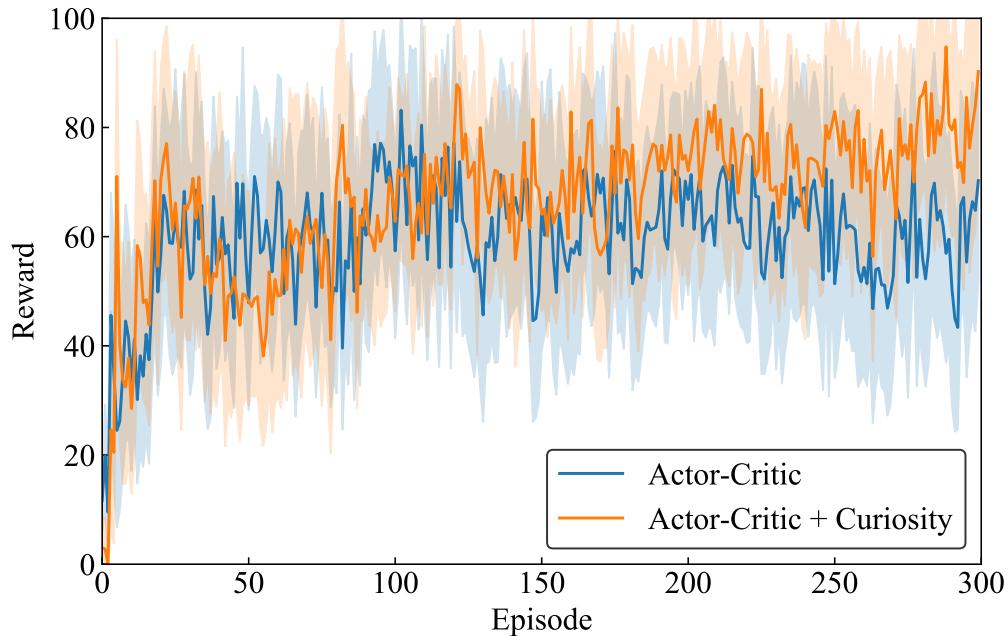


Fig. 5.5: Comparison of the reward by the proposed method compared to no curiosity actor-critic

5.5.3 考察

図 5.5 より、好奇心を入れない場合では、報酬の平均値はエピソード数が 100 あたりで頭打ちになり、最終エピソードまでは横ばいに近い状態が続いている。一方、好奇心を入れた場合では、最終エピソードまで報酬が右肩上がりになっており、好奇心を入れない場合よりも高い報酬が得られていることが分かる。この理由として、式 (5.29) による好奇心をロボットに与えることで、ロボットは訓練データが少なく、まだ探索が不足している領域を積極的に探索することで、ランダムな探索では見つけきれなかったより高い報酬が得られる状態へ遷移できたことが考えられる。

5.6 触覚センサを用いた5本指ロボットハンドによる 物体操作タスクの学習

5.5 節では、ボールの位置はカメラで計測されるものとした。しかし、カメラのような視覚センサは手前にある物体が背後にいる物体を隠すオクルージョンの問題があり、ボールの

位置を特定できない場合がある。一方、触覚センサでは、接触している状況からボールの位置を特定できるため視覚センサ特有の問題を回避することができる。

5.6.1 実験方法

本節では、図 5.6、5.7 に示すように、複数のボールをロボットハンドの上に乗せた状態から、1 個ずつ時間間隔をあけてボールを落とすことをタスクとした。ロボットハンドは 5.5.1 節と同じく Shadow Dexterous Hand であり、行動空間も同様である。報酬については次のように与えた。まず、ボールを落とした時点で 2 秒間ロボットの行動を停止させた。2 秒停止の間に次のボールが落ちなければ、ボールを落とした直前の状態 s と行動 a に対して、 $r(s, a) = 100$ の報酬を与えた。2 秒停止の間に次のボールが落ちたら、ボールを落とした直前の状態 s と行動 a に対して、 $r(s, a) = -100$ の報酬を与えた。ボールが落ちたかどうかは床からの高さで判定した。

ボールの位置は触覚センサで計測した。触覚センサは図 5.6、5.7 に示すロボットハンドの白い正方形 ($1\text{cm} \times 1\text{cm}$) であり、手のひらに合計 31 個付けた。触覚センサはアナログ値である。

1 エピソードは図 5.6 に示すボール 2 個の場合は、開始から 60 ステップ (6 秒) 経過、あるいはボールが全て床に落ちた時点で終了とした。図 5.7 に示すボール 3 個の場合は、開始から 100 ステップ (10 秒) 経過、あるいはボールが全て床に落ちた時点で終了とした。

1 回の実験に含まれるエピソード数はボール 2 個が 300、ボール 3 個が 700 である。

実験に用いたハイパープラメータを表 5.3 に示す。

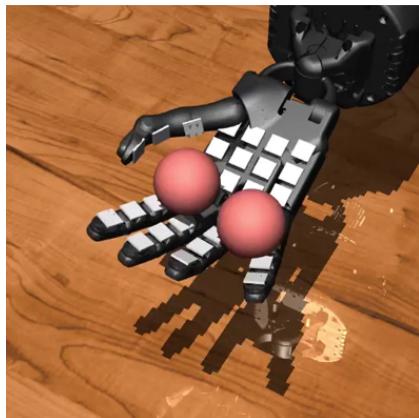


Fig. 5.6: Two balls on the robot hand with tactile sensors

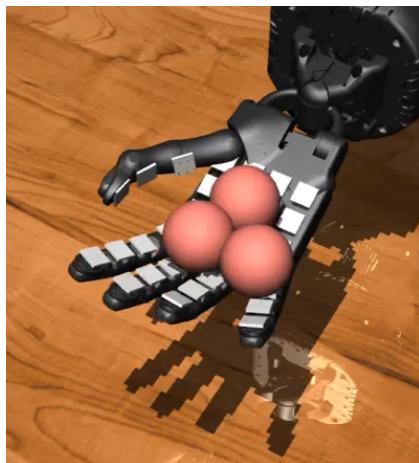


Fig. 5.7: Three balls on the robot hand with tactile sensors

Table 5.2: Hyperparameters during experiment

Parameter	Value
number of hidden layers	2
number of hidden units for layers1	300
number of hidden units for layer2	300
minibatch size	64
buffer size	10000
optimizer	Adam[60]
learning rate l_c	5e-4
learning rate l_a	1e-4
learning rate l_f	1e-4
activation function	ReLU

5.6.2 実験結果

ボール2個の場合の実験を20回行った時のエピソード数と報酬の関係を図5.8に示す。また、Rewardが200まで到達するのに必要なエピソード数を図5.10に示す。

ボール3個の場合の実験を20回行った時のエピソード数と報酬の関係を図5.9に示す。また、Rewardが300まで到達するのに必要なエピソード数を図5.11に示す。

5.6.3 考察

図5.10、5.11より、ボール2個と3個のどちらの場合においても、好奇心を入れて学習した方が少ないサンプルでタスクの学習ができていることがわかる。

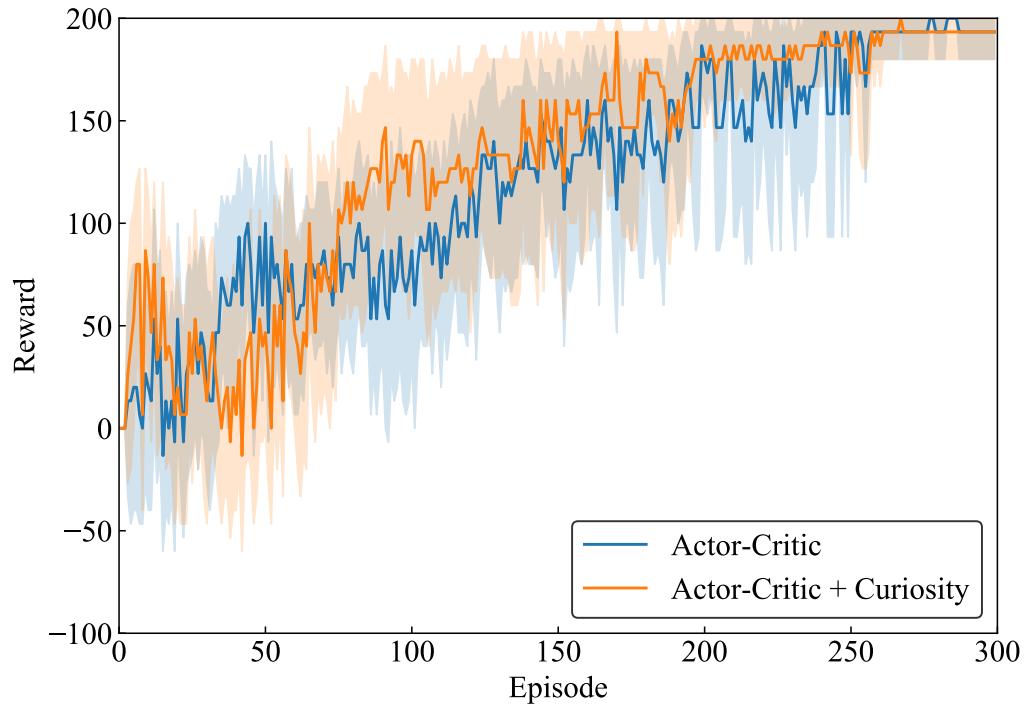


Fig. 5.8: Differences in reward between the proposed method and Actor-Critic in the case of two balls with a five-fingered robot hand

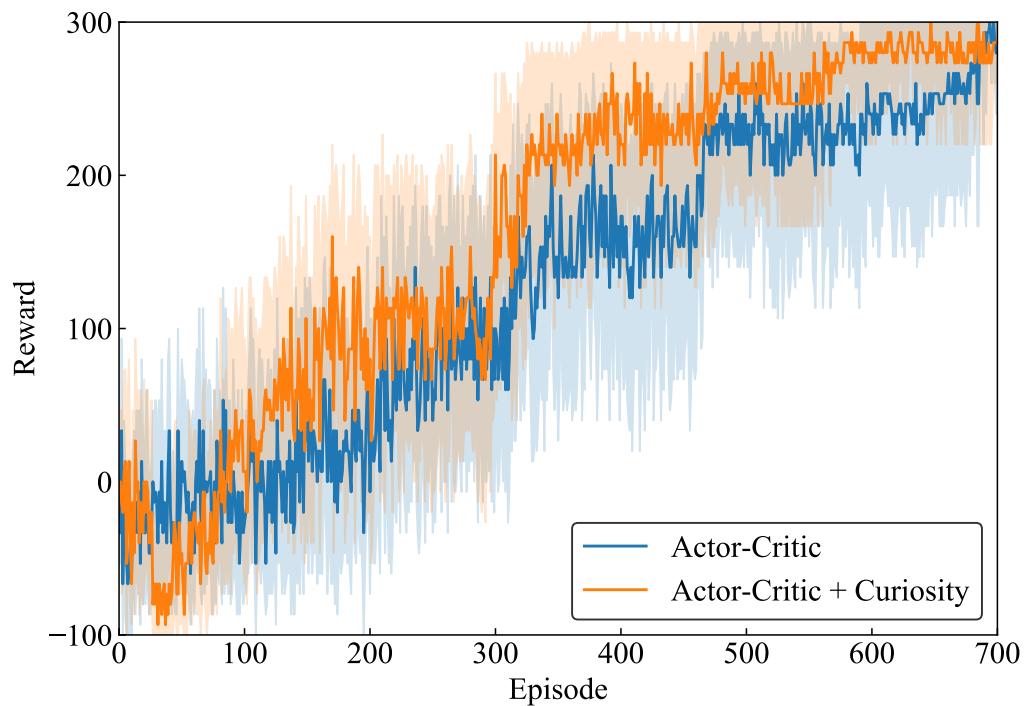


Fig. 5.9: Differences in reward between the proposed method and Actor-Critic in the case of three balls with a five-fingered robot hand

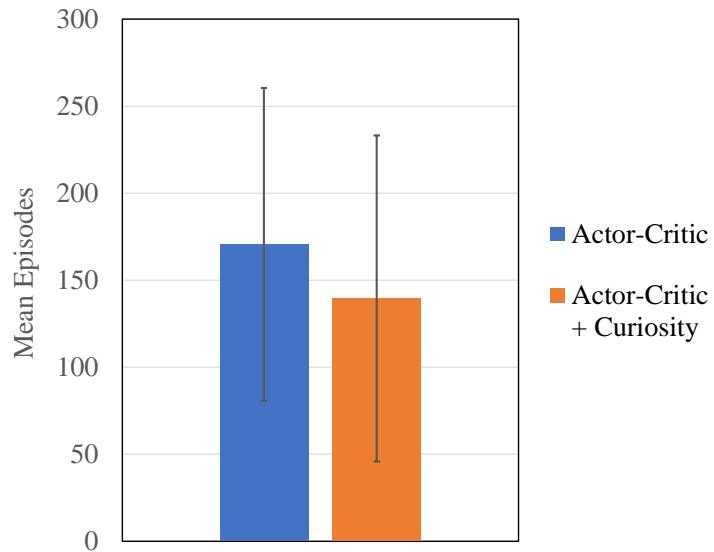


Fig. 5.10: Number of samples needed for reward to reach 200 in the case of two balls with a five-fingered robot hand

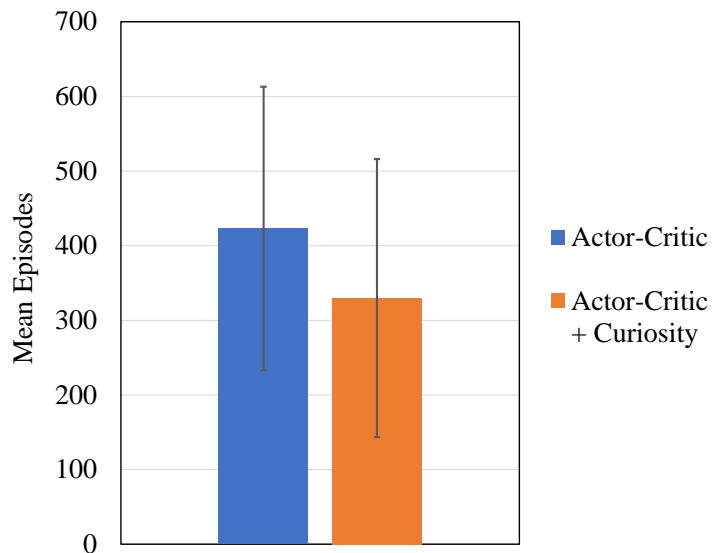


Fig. 5.11: Number of samples needed for reward to reach 300 in the case of three balls with a five-fingered robot hand



Fig. 5.12: Three large balls and robot hand

5.7 触覚センサを用いた3本指ロボットハンドによる 物体操作タスクの学習

本節では、次章で実機実験を行うため、図 5.12 に示す 3 本指のロボットハンドを用いて物体操作タスク学習のシミュレーション実験を行った。

5.7.1 実験方法

図 5.12 に示すロボットには触覚センサを合計 72 個付けた。ロボットの各指は、第 1, 2 関節の屈伸と、指の付け根の開きが可能である。ただし、本節の実験では、3 本指のうちボールを支えていない上の指に関しては、付け根の開きの自由度はないものとした。したがって、ロボットの行動空間は 8 次元になる。

本節では、図 5.13～5.15 に示すように、複数のボールをロボットハンドの上に乗せた状態から、1 個ずつ時間間隔をあけてボールを落とすことをタスクとした。

図 5.13 に示すボール 3 個の場合は、5.6.1 節と同じように報酬を与えた。1 エピソードは開始から 80 ステップ、あるいはボールが全て床に落ちた時点で終了とした。

図 5.14, 5.15 に示すボール 4 個、5 個についてはボールを落とした時点で以下のような報酬を与えた。

$$r = 100 \times (10 - n) \quad (5.31)$$

ただし、 n はボールを落とすのにかかったステップである。1ステップが0.1秒であるので、ボールを落とすのに10ステップ以内、つまり1秒以内の場合に正の報酬が入るようにした。ボール3個の場合と報酬を変えた理由としては、一定間隔でボールを落とす行動を学習されるためである。ボールを落として2秒停止後、次のボールを落とすまでのステップ数を少なくすることで一定(2秒)間隔でボールを落とすことができる。

1エピソードはボール4個、5個でそれぞれ開始から90ステップ、130ステップ、あるいはボールが全て床に落ちた時点で終了とした。

図5.13、5.14のボールの直径は60mm、図5.15のボールの直径は25mmとした。

実験に用いたハイパーパラメータを表5.3に示す。

Table 5.3: Hyperparameters during experiment

Parameter	Value
number of hidden layers	2
number of hidden units for layers1	500
number of hidden units for layer2	500
minibatch size	64
buffer size	10000
optimizer	Adam[60]
learning rate l_c	5e-4
learning rate l_a	1e-4
learning rate l_f	1e-4
activation function	ReLU



Fig. 5.13: Three large balls and robot hand



Fig. 5.14: Four large balls and robot hand

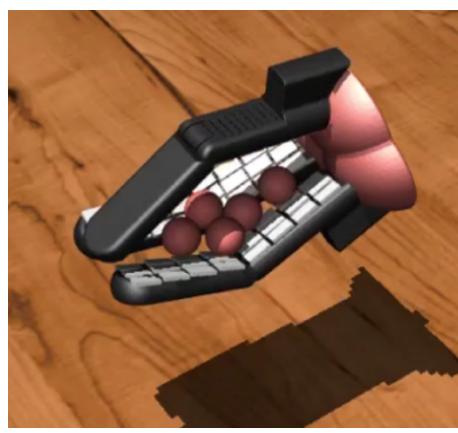


Fig. 5.15: Five large balls and robot hand

5.7.2 実験結果

ボール3個、4個、5個の実験結果をそれぞれ図5.16、5.17、5.18に示す。各ボールの実験は20回行った。

5.7.3 考察

図5.16、5.17、5.18から分かるように、提案手法によりロボットに好奇心をもたせた場合とそうでない場合のロボットが環境から受け取る報酬に大きな違いは見られなかった。ボール3個とボール4個を1個ずつ落とすタスクにおいては、図5.16、5.17より、エピソードの後半になるまで好奇心を入れた場合よりも好奇心を入れない場合の方が得られる報酬が高くなっている。これは、ロボットが環境から報酬を受け取った際、好奇心を入れない場合には、その報酬が得られる行動が取られやすくなるのに対し、好奇心を入れた場合には、まだ探索が十分に行えていない状態があればそちらへの行動を優先することがあるためだと考えられる。環境から受け取る報酬が疎な場合はこれまで訪れた状態を繰り返すよりもまだ探索していない領域へ行動をしたほうが報酬を見つける可能性が高くなる。今回の実験ではエピソード前半においてすでにロボットは環境からの報酬を獲得しており、報酬が疎な環境であるとは言い難い。しかし、エピソードの後半では好奇心を入れた場合の方が好奇心を入れなかった場合よりも少しだけ高い報酬を獲得していることが分かる。これは、今回の実験で高い報酬を得るためににはボール1個目だけでなく、2個目、3個目とより多くのボールを1個ずつ落とす必要があり、そのためにはある程度の探索が必要となる。すでに報酬が得られた状態まわりでの探索では不十分な恐れがある。好奇心を入れることでより広範囲の探索が可能となり、結果として最終エピソードにおける報酬に違いが生まれたものと考えられる。

5.8 おわりに

本章では深層強化学習によるロボットのタスク学習において、サンプル効率の良いデータ収集のための好奇心アルゴリズムを提案した。また、シミュレーション実験により一部のタスク学習においてサンプル効率が改善することを示した。

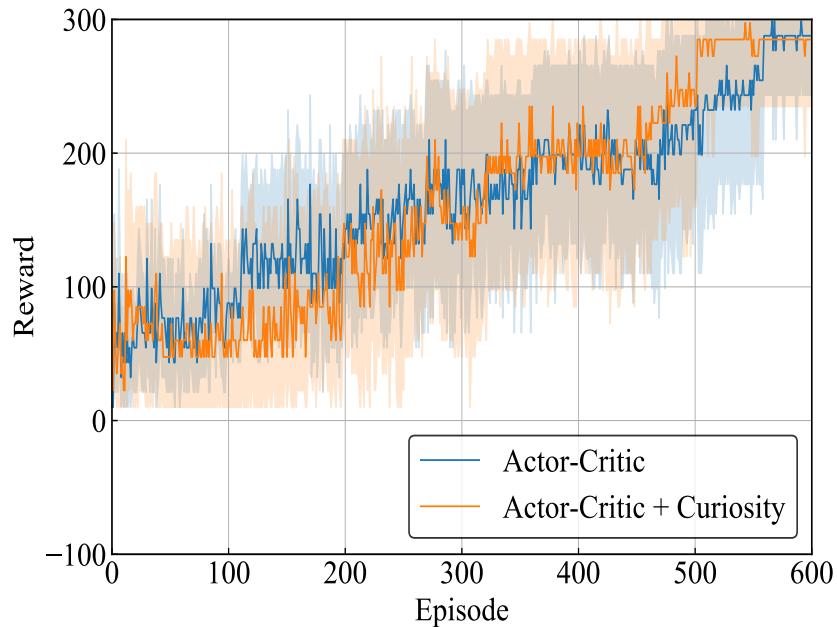


Fig. 5.16: Differences in reward between the proposed method and Actor-Critic in the case of three balls with a three-fingered robot hand

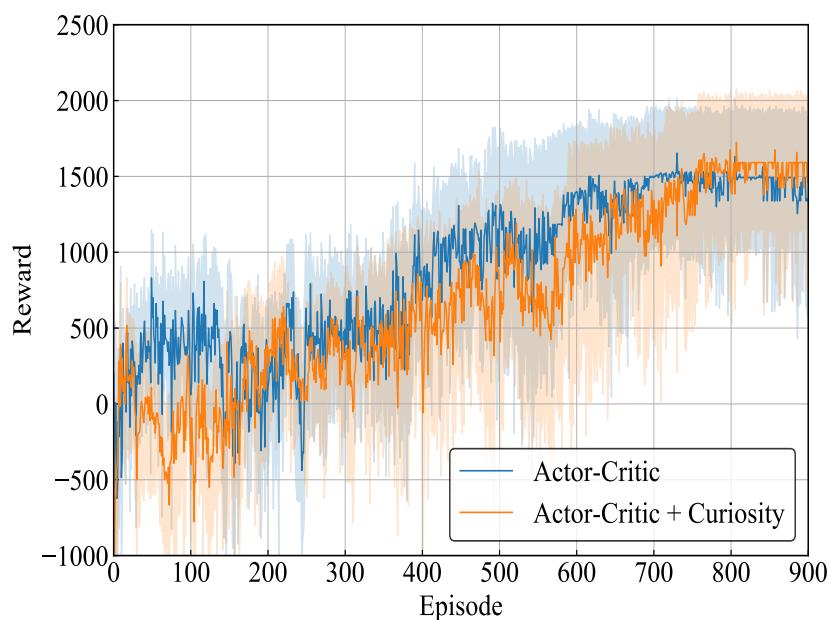


Fig. 5.17: Differences in reward between the proposed method and Actor-Critic in the case of four balls with a three-fingered robot hand

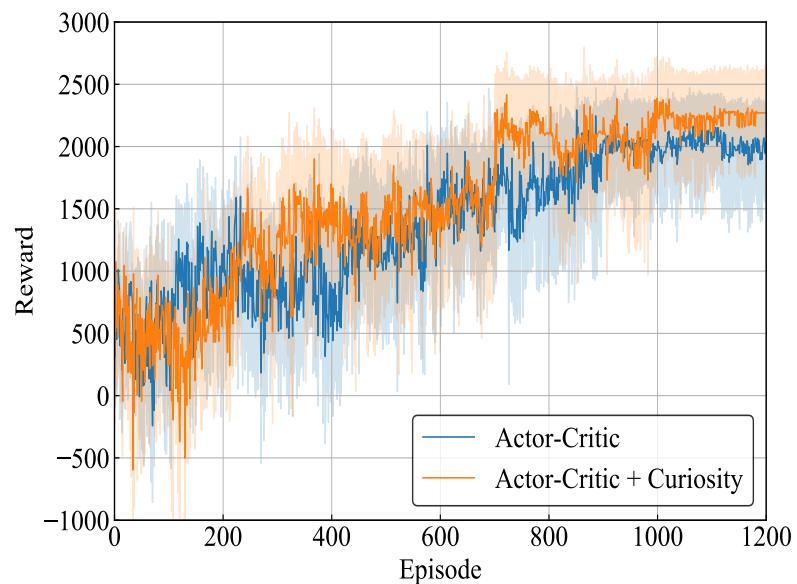


Fig. 5.18: Differences in reward between the proposed method and Actor-Critic in the case of five balls with a three-fingered robot hand

第6章

好奇心行動によるロボットハンドの 物体操作実験



Fig. 6.1: Robot hand

6.1 はじめに

本章では前章で提案した手法の有効性を実機を使った実験により検証する。

6.2 実験方法

本節では図 6.1 に示すロボットハンドを用いて実験を行った。5.7 節と同様、ロボットハンドの上に乗せているボールを一定間隔で落とすというタスクを行った。タスクの学習は Algorithm5 に示す提案手法によってシミュレーション (MuJoCo) 上で学習した。学習した動作を実機に関節角度の時系列データとして送り、実機を動かした。

ボールの大きさは 55 mm で 3 個とした。また、5.7 節でボールを落とした後、行動を 2 秒停止させていたが、停止中に手から落ちるボールに対して落ちないようにすることができない。そこで本節では、行動を 2 秒停止させるのをやめ、代わりにボールを落としてから次のボールを落とすまでの相対時間（ステップ数）を目標に学習を行った。次のボールを落とすのにかかったステップ数を t とおく。また、目標の相対ステップ数を T としたとき、ボールを落とした直前の状態と行動に対して以下の報酬 r を与えた。

$$r = 100 - 10 \times |T - t| \quad (6.1)$$

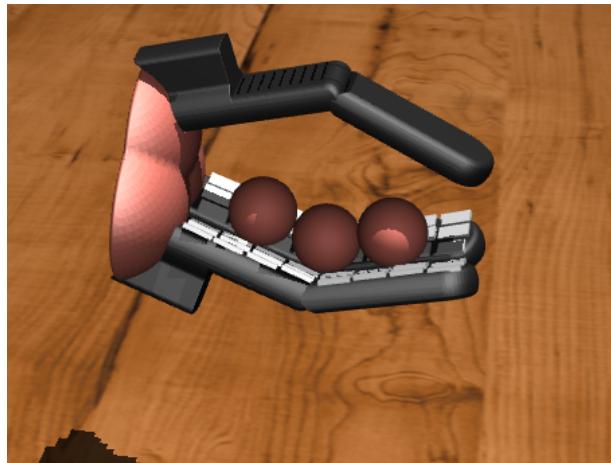


Fig. 6.2: Robot hand in simulation

それ以外の報酬は0とした。実験では $T = 20$ とした。1ステップは0.1秒である。

6.3 実験結果

シミュレーションの実験結果を図6.3に示す。シミュレーションと実機で、3つのボールがそれぞれ落ちたときのステップ数を表6.1に示す。また、ステップ0～ステップ57までの実機の動作結果を図6.4に示す。

Table 6.1: Number of steps the ball fell

	simulation	real environment
Ball1	13	15
Ball2	31	30
Ball3	53	54

6.4 考察

表6.1より、3つのボールを落としたときのステップ数がシミュレーションと実機で近いことが分かる。図6.4より、シミュレーションで学習した動作により実機でもボールを1個ずつ時間間隔をあけて落とせていることが分かる。しかし、表6.1を見るとシミュレーション

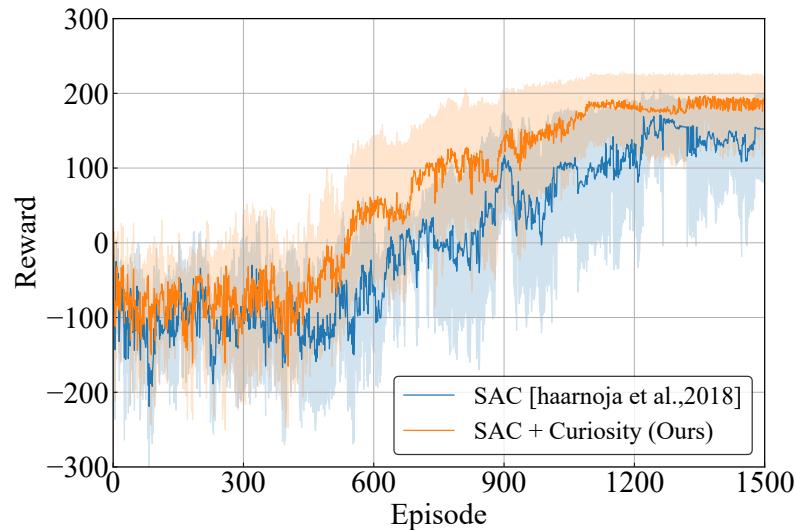


Fig. 6.3: Comparison between Soft Actor-Critic+Curiosity based on the Free energy principle and the baseline Soft Actor-Critic. The lines represent the medians, and the shaded areas depict the percentiles 5 to 95 over 10 experiments.

ンと実機で数ステップ数の違いがみられる。原因としては、シミュレーションと実環境の差によるものと考えられる。とくに、ボールの接触モデルやロボットの初期姿勢のずれ、ロボットを動かしている間の振動などが差を生む主な要因と考えられる。対策としては、ロボット性を高めるためにシミュレーション時にロボットの行動や観測値にノイズを入れて学習されるなどの工夫が必要であると考えられる。

6.5 おわりに

本章では前章で提案した手法の有効性を実機を使った実験により検証した。



Fig. 6.4: Motion of the robot hand using time series data of the motion learned by the proposed method on the simulation

第7章

結論と今後の展望

7.1 結論

本研究では、ロボットのタスク学習におけるサンプル効率を改善するため、自由エネルギー原理に基づく好奇心アルゴリズムを提案した。

ロボットハンドによる物体操作学習のシミュレーション実験を行い、提案手法を用いることで一部のタスク学習においてサンプル効率が改善することを示した。

提案手法によってシミュレーション上で学習した動作を用いて、実環境でロボットハンドの物体操作タスクを成功させた。

7.2 今後の展望

今後の展望として、提案手法の有効性を物体操作以外のロボットのタスク学習において検証することや、ICMなどの従来研究との比較が考えられる。

また、ロボットのセンサ値に基づいた実環境でのタスク学習も考えられる。

謝辞

本論文を執筆するにあたって、多大なるご助言やご指導をいただきました東京農工大学機械システム工学科 水内郁夫 教授に深く感謝すると同時に厚く御礼申し上げます。また、研究に関する相談に乗っていただきたり、原稿や発表資料にコメントをしてくださったりした先輩方、同輩にも深く感謝いたします。

参考文献

- [1] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.
- [2] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pp. 651–673, 2018.
- [3] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, Vol. 37, No. 4-5, pp. 421–436, 2018.
- [4] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, Vol. 4, No. 26, 2019.
- [5] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018.
- [6] Yoshihisa Tsurumine, Yunduan Cui, Eiji Uchibe, and Takamitsu Matsubara. Deep dynamic policy programming for robot control with raw images. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1545–1550. IEEE, 2017.
- [7] Yoshihisa Tsurumine, Yunduan Cui, Eiji Uchibe, and Takamitsu Matsubara. Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation. *Robotics and Autonomous Systems*, Vol. 112, pp. 72–83, 2019.
- [8] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, Vol. 17, No. 1, pp. 1334–1373, 2016.

- [9] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, Vol. 135. MIT press Cambridge, 1998.
- [10] Yuxi Li. Deep reinforcement learning. *arXiv preprint arXiv:1810.06339*, 2018.
- [11] Iyaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*, 2017.
- [12] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.
- [13] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*, 2018.
- [14] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pp. 1471–1479, 2016.
- [15] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.
- [16] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *arXiv preprint arXiv:1605.09674*, 2016.
- [17] Karl Friston. The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, Vol. 13, No. 7, pp. 293–301, 2009.

- [18] Karl Friston, James Kilner, and Lee Harrison. A free energy principle for the brain. *Journal of Physiology-Paris*, Vol. 100, No. 1-3, pp. 70–87, 2006.
- [19] Harriet Brown and Karl J Friston. Free-energy and illusions: the cornsweet effect. *Frontiers in psychology*, Vol. 3, p. 43, 2012.
- [20] Karl Friston, Rick Adams, Laurent Perrinet, and Michael Breakspear. Perceptions as hypotheses: saccades as experiments. *Frontiers in psychology*, Vol. 3, p. 151, 2012.
- [21] Thomas Parr and Karl J Friston. Active inference and the anatomy of oculomotion. *Neuropsychologia*, Vol. 111, pp. 334–343, 2018.
- [22] Karl J Friston, Tamara Shiner, Thomas FitzGerald, Joseph M Galea, Rick Adams, Harriet Brown, Raymond J Dolan, Rosalyn Moran, Klaas Enno Stephan, and Sven Bestmann. Dopamine, affordance and active inference. *PLoS Comput Biol*, Vol. 8, No. 1, p. e1002327, 2012.
- [23] Harriet Feldman and Karl Friston. Attention, uncertainty, and free-energy. *Frontiers in human neuroscience*, Vol. 4, p. 215, 2010.
- [24] Karl Friston, Jérémie Mattout, and James Kilner. Action understanding and active inference. *Biological cybernetics*, Vol. 104, No. 1-2, pp. 137–160, 2011.
- [25] Karl J Friston, Rebecca Lawson, and Chris D Frith. On hyperpriors and hypopriors: comment on pellicano and burr. *Trends in cognitive sciences*, Vol. 17, No. 1, p. 1, 2013.
- [26] Karl J Friston, Vladimir Litvak, Ashwini Oswal, Adeel Razi, Klaas E Stephan, Bernadette CM Van Wijk, Gabriel Ziegler, and Peter Zeidman. Bayesian model reduction and empirical bayes for group (dcm) studies. *Neuroimage*, Vol. 128, pp. 413–431, 2016.
- [27] Karl Friston. Embodied inference: or “ i think therefore i am, if i am what i think”. 2011.

- [28] Karl J Friston, Jean Daunizeau, James Kilner, and Stefan J Kiebel. Action and behavior: a free-energy formulation. *Biological cybernetics*, Vol. 102, No. 3, pp. 227–260, 2010.
- [29] Karl Friston and Stefan Kiebel. Cortical circuits for perceptual inference. *Neural Networks*, Vol. 22, No. 8, pp. 1093–1104, 2009.
- [30] Karl Friston, Philipp Schwartenbeck, Thomas FitzGerald, Michael Moutoussis, Tim Behrens, and Raymond J Dolan. The anatomy of choice: active inference and agency. *Frontiers in human neuroscience*, Vol. 7, p. 598, 2013.
- [31] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
- [32] Karl Friston. The history of the future of the bayesian brain. *NeuroImage*, Vol. 62, No. 2, pp. 1230–1233, 2012.
- [33] Kenji Doya, Shin Ishii, Alexandre Pouget, and Rajesh PN Rao. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.
- [34] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, Vol. 7, No. 6, pp. 1129–1159, 1995.
- [35] Ralph Linsker. Self-organization in a perceptual network. *Computer*, Vol. 21, No. 3, pp. 105–117, 1988.
- [36] Takuma Tanaka, Takeshi Kaneko, and Toshio Aoyagi. Recurrent infomax generates cell assemblies, neuronal avalanches, and simple cell-like selectivity. *Neural computation*, Vol. 21, No. 4, pp. 1038–1067, 2009.
- [37] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, Vol. 2, No. 1, pp. 79–87, 1999.

- [38] Yanping Huang and Rajesh PN Rao. Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, Vol. 2, No. 5, pp. 580–593, 2011.
- [39] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, Vol. 5, No. 7, pp. 1688–1703, 1985.
- [40] Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, Vol. 5, No. 11, pp. 1226–1235, 2002.
- [41] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, Vol. 11, No. 2, pp. 127–138, 2010.
- [42] Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, Vol. 360, No. 1456, pp. 815–836, 2005.
- [43] Charles W Fox and Stephen J Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, Vol. 38, No. 2, pp. 85–95, 2012.
- [44] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, Vol. 22, No. 1, pp. 79–86, 1951.
- [45] Karl Friston, Francesco Rigoli, Dimitri Ognibene, Christoph Mathys, Thomas Fitzgerald, and Giovanni Pezzulo. Active inference and epistemic value. *Cognitive neuroscience*, Vol. 6, No. 4, pp. 187–214, 2015.
- [46] Karl Friston, Thomas Fitzgerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural computation*, Vol. 29, No. 1, pp. 1–49, 2017.
- [47] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision research*, Vol. 49, No. 10, pp. 1295–1306, 2009.
- [48] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, Vol. 69, No. 6, p. 066138, 2004.

- [49] Roland Benabou and Jean Tirole. Intrinsic and extrinsic motivation. *The review of economic studies*, Vol. 70, No. 3, pp. 489–520, 2003.
- [50] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, Vol. 47, No. 1, pp. 90–100, 2003.
- [51] Mark J van der Laan and Susan Gruber. Collaborative double robust targeted maximum likelihood estimation. *The international journal of biostatistics*, Vol. 6, No. 1, 2010.
- [52] Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 73, No. 1, pp. 3–36, 2011.
- [53] Giorgio Parisi. *Statistical field theory*. Addison-Wesley, 1988.
- [54] Igor V Tetko, David J Livingstone, and Alexander I Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of chemical information and computer sciences*, Vol. 35, No. 5, pp. 826–833, 1995.
- [55] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.
- [56] Laurent Valentin Jospin, Wray Buntine, Farid Boussaid, Hamid Laga, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *arXiv preprint arXiv:2007.06823*, 2020.
- [57] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.
- [58] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, Vol. 31, No. 2, pp. 105–112, 2009.

- [59] Hermann G Matthies. Quantifying uncertainty: modern computational representation of probability and applications. In *Extreme man-made and natural hazards in dynamics of structures*, pp. 105–135. Springer, 2007.
- [60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [61] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, Vol. 28, pp. 2575–2583, 2015.
- [62] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, Vol. 521, No. 7553, pp. 436–444, 2015.
- [63] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.
- [64] ShadowRobot. Shadowrobot dexterous hand, 2005. <https://www.shadowrobot.com/dexterous-hand-series/>.
- [65] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pp. 421–436. Springer, 2012.
- [66] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [67] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, Vol. 8, No. 3-4, pp. 279–292, 1992.
- [68] Bellman Richard Ernest. Dynamic programming, 2003.
- [69] Bellman Richard Ernest and E Bellman. Adaptive control processes: a guided tour, 1961.

- [70] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [71] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [72] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, No. 5, pp. 834–846, 1983.
- [73] 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.
- [74] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, Vol. 8, No. 3-4, pp. 293–321, 1992.
- [75] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.
- [76] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, 2018.
- [77] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.
- [78] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

- [79] Boyao Li, Tao Lu, Jiayi Li, Ning Lu, Yinghao Cai, and Shuo Wang. Acder: Augmented curiosity-driven experience replay. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4218–4224. IEEE, 2020.
- [80] Boyao Li, Tao Lu, Jiayi Li, Ning Lu, Yinghao Cai, and Shuo Wang. Curiosity-driven exploration for off-policy reinforcement learning methods. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1109–1114. IEEE, 2019.
- [81] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8. IEEE, 2016.
- [82] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [83] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.
- [84] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, Vol. 36, No. 5, p. 823, 1930.
- [85] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.