スタークラフトIIのミニゲームにおけるマルチタスク強化学習

徐 凡超[1,a)]　金子 知適[1,,b)]

概要：リアルタイムストラテジー・コンピュータゲームのスタークラフト II は人工知能の新しい実験対象として，注目されている．特に，人間プレーヤーと同じ様なインターフェースと制限で束縛されると，人工知能エージェントがフルゲームを人間レベルに学習させるのは困難である．現時点の最新手法で学習したエージェントは，一般的な人間プレーヤと互角に対戦する性能があるが，トッププレーヤには及ばない．このゲームそのもの (フルゲーム) でトッププレーヤに勝利することが，スタークラフト II 研究における大きな目的である．しかし，膨大な計算資源を消費するなどの原因で，一般研究者が研究に用いるのは簡単ではない．DeepMind 社が提出した，フルゲームのサブタスクに基づいて作られた 7 つのミニゲームのお掛けで，スタークラフト II の研究をよりやりやすくなった．本研究はこれらのミニゲームを対象にして，複数のミニゲームをマスターできる汎化型エージェント開発するために，現時点で最上位のパフォーマンスをもつエージェントを実装し，シングルタスクの再現実験とマルチタスクの性能調査実験を行った．実験ではタスク間の異なる部分がマルチタスクのパフォーマンスに対する影響を示した．

キーワード：リアルタイムストラテジーゲーム，スタークラフト II，強化学習，マルチタスク

Multi-task Reinforcement Learning in StarCraftII Mini-games

Fanchao Xu[1,a)]　Tomoyuki Kaneko[1,,b)]

Abstract: As a new domain for AI researches, StarCraftII becomes more and more popular. However, training AI agent to play StarCraftII in human level with human-like interface and limitations is a challenge. The state-of-the-art agent from DeepMind can beat some professional players, but not the best one. We believe that AI research will achieve a new level if AI agents can beat top-level human professional players. Meanwhile, the research on full game requires the huge amount of computing resources which are unavailable for most of researchers. Thanks to mini-games proposed by DeepMind which are based on subtasks from full game, starting from mini-games is an easier way than full-game. We implement the state-of-the-art agent architecture and algorithm, then test the performance in single-task training and multi-task training. Experimental results showed that different tasks affect the performance of multi-task learning.

Keywords: Real-time strategy game, StarCraftII, Reinforcement learning, Multi-task

1. はじめに

人工知能エージェントが現実に近い複雑な問題を解決できるようにするために，囲碁，チェスなどの伝統ボードゲームだけでなくコンピュータゲームも学習環境として使われている．本研究の対象としてのスタークラフト II は Blizzard Entertainment が開発したリアルタイムストラテジー・コンピュータゲームである．2017 年，DeepMind 社が Open Source の学習環境 PYSC2 [1] を発表して以来，より多くの研究者がこのゲームに興味を示してきた．フルゲームの学習はまだ難しいため，その代わりに DeepMind 社が提案した 7 つのミニゲームを対象にして研究を行った．

スタークラフト II ミニゲームに関する既存研究は主にシングルタスクで訓練し，高いパフォーマンスを求める研究である．本研究では人間がフルゲームをプレイするために，様々なミニゲームの戦略を身につけるように，複数の

[1] 東京大学大学院情報学環
Interfaculty Initiative in Information Studies, the University of Tokyo
[a)] xu-fanchao@g.ecc.u-tokyo.ac.jp
[b)] kaneko@acm.org

ミニゲームをマスターできる汎化型エージェントを目指して，将来的にフルゲームに適用することを目指すことである．そのために，スタークラフト II ミニゲームにシングルタスク実験と，Atari game などの環境でマルチタスク実験を成功した文献 [2] に基づいて，7 つのミニゲームを 4 種類に分類し，マルチタスク設定で，エージェントを訓練して，パフォーマンスと改善点を議論する．

2. 既存研究

2.1 Single-Task

2.1.1 PYSC2 and A3C

スタークラフト II の 3D 画像からゲームの状態を直接に観測するのは難しい．故に Blizzard Entertainment と DeepMind が開発した PYSC2-StarCraft 2 Learning Environment [1] で feature layers という 3D 画像を抽象化する処理層が提供された．大きく分類すると spatial features(Minimap: 7 feature layers, Screen: 17 feature layers) と non-spatial features(General player information, Control groups など全部で 10 種類) が存在する．人間プレーヤーとの対戦を公平にする必要があるので action space が 1 つの action function と複数の action arguments の形で定義されている．Action function は全部で 549 種類があり，action arguments は 13 種類がある．

Vinyals O，et al [1] が提案した手法は既存研究の Asynchronous Advantage Actor-Critic [3](A3C) を利用し，3 つのネットワーク構造で single-task のスタークラフト II ミニゲームを学習する手法である．

A3C は Actor-Critic の構造で複数の環境を用意し，始めに，Actors が Critic からの最新のネットワークパラメータを使って更新する．次は，それぞれの環境で経験を積み重ねて，n-steps および 1 episode が終了するたびに勾配を計算し，Critic に渡す．Critic がそれらの勾配を用いてパラメータを更新し，新しいパラメータを Actor に渡すという仕組みである．

Atari-net，FullyConv と FullyConv LSTM の 3 つのネットワーク構造が提案された．

2.1.2 IMPALA

Importance Weighted Actor-Learner Architecture [4](IMPALA) は A3C の上で発展した学習効率と安定性が優れる手法である．

A3C に似ている構造を持っているが．Actors が経験を集めるだけで勾配計算を行わない．学習に参加する learner が全 actors からの trajectories(n-steps の states，actions，rewards)，batch size 分を集めると学習する．Actors は次の n-steps が始まる前に learner の最新のパラメータを使って更新する．

マルチタスク学習の一種はニューラルネットワーク部分層を共有することで複数のタスクが学習できる仕組みである [5]．IMPALA では学習を行う learner のニューラルネットワークを共有することでマルチタスク設定で学習するのは可能である．文献 [4] は DMLab-30 と Atari-57(57 Atari 2600 games) でマルチタスク実験を行った．マルチタスク学習の性能は A3C 以上であることが分かった，しかし，Atari game のシングルタスク訓練結果はマルチタスク訓練結果をかなり上回る．

2.1.3 RDRL

最初に IMPALA をスタークラフト II ミニゲームで訓練する研究は本研究ではなく，Relational Deep Reinforcement Learning [2] の論文である．これは入力情報の更なるハイレベルの内部関連性を抽出することで，高いパフォーマンス，学習効率，汎用性を期待している手法である．文献 [6] が提案した multi-head dot-product attention(MHDPA) に基づいて，relational block という構造が提案し，複数の relational blocks が構成された relational module を用いて，新しいニューラルネットワーク構造を提案した．要するに，新しく提案されたネットワーク構造と IMPALA の組み合わせである．

文献 [2] は Box-World とスタークラフト II ミニゲーム二つの環境でシングルタスク実験をした．Box-World での実験は，relational module を使うことにより，期待通りの高いパフォーマンスが観察された．しかしながらスタークラフト II ミニゲームでの実験は relational block を使うエージェント (relational agent) は使わないエージェント (control agent) と近いパフォーマンスが得られた，要するに，relational module を使うことで，ミニゲームの single-task パフォーマンスを上がる証拠が不十分である．他の手法と比べると，2019 年 9 月 26 日現在，single-task 設定で，ミニゲームの CollectMineralShards，DefeatRoaches，DefeatZerglingsAndBanelings と FindAndDefeatZerglings の記録ランキング [7] の一位を維持している．故に，本研究は，relational agent と control agent はスタークラフト II ミニゲーム single-task の最適ネットワーク構造と認識し，control agent を baseline として，研究を行った．

2.2 Multi-Task

単一エージェントが複数の異なるタスクを学習するのは，また，人工知能領域の難題である．問題点の一つは違うタスクの reward のスケールと分布が大きく変わると，学習とパフォーマンスに顕著な影響を与えることである．ただ一様に clip すると，最適 policy と全く違う方向に収束する可能性がある．故に，Preserving Outputs Precisely while Adaptively Rescaling Targets(PopArt) [8] が提出され，value-based 手法の Q-learning with a deep neural network(DQN) [9] 及び Double DQN [10] と組み合わせて，Atari-57 で実験を行った (multi-task ではなく，シングルセットのパラメータで複数のタスクを学習することであ

表 1　スタークラフト II ミニゲームのテスト平均点数
Table 1　Mean scores for StarCraftII mini-games.

		MoveToBeacon	CollectMineralShards	FindAndDefeatZerglings	DefeatRoaches
single-task	Relational agent [2]	27	196	62	303
	Control agent [2]	27	187	61	295
	FullyConv LSTM [1]	26	104	44	98
	Human Expert [1]	28	177	61	215
	Control agent 実装	21.64(1.62) [18, 25]	117.47(8.13) [97, 134]	51.71(4.94) [27, 58]	147.03(83.20) [37, 333]
multi-task(本研究)	2 tasks	23.49(1.62) [20, 28]	111.97(6.98) [91, 129]		
	3 tasks	23.67(1.96) [20, 29]	103.68(6.61) [88, 117]	45.66(5.85) [17, 52]	
	4 tasks	1.47(1.24) [0, 7]	25.94(8.40) [8, 43]	16.94(5.04) [2, 24]	148.97(77.94) [37, 314]

る)，一様に clip するやり方より良い手法であることを証明した．

文献 [11] が，IMPALA と PopArt の手法を提出した．Atari-57 と DmLab-30 二つの環境でマルチタスク学習を行って，オリジナルの IMPALA よりマルチタスク性能が高いであることが分かった．

3. 実験

DeepMind 社が提出したミニゲームは特定のユニットの構築，リソースの収集，地図を用いる移動など全部で 7 つがある [1]．タスク目標で 4 種類に分類した．それぞれは

1. **Building task** ：BuildMarines
2. **Moving task** ：MoveToBeacon
3. **Collecting tasks** ： CollectMineralShards CollectMineralsAndGas
4. **Fighting tasks** ： FindAndDefeatZerglings DefeatRoaches DefeatZerglingsAndBanelings

である．

既存手法の control agent [2]，baseline として，4 つのスタークラフト II ミニゲーム (MoveToBeacon，CollectMineralShards，FindAndDefeatZerglings，DefeatRoaches) におけるシングルタスクと，2 タスク，3 タスク，4 タスクのマルチタスク実験を行った．計算資源制限のため，本実験での observation は spatial features(Minimap: 5 feature layers, Screen: 7 feature alyers) と non-spatial features(General player information) である．同じ原因で，control agent [2] の 12 層 4blocks の deep residual model の代わりに 6 層 2blocks の residual model が使われている．**表 1** は既存研究のミニゲームにおけるそれぞれの学習済みベストエージェントの 30 episode の $Mean\ score$ と，本研究の学習済みエージェントの 100 episode の $Mean\ score(standrad\ deviation)[Min, Max]$ になる．

実験で使用した library は pysc2 2.0.2，dm-sonnet 1.27，numpy 1.14.5，tensorflow-gpu 1.13.1，tensorflow-probability 0.5.0 である．

3.1 Single-task

Atari game に適用する IMAPLA はすでに open-source[*1]になったため，そのコードに基づいて，control agent [2] を実装し，スタークラフト II に適用できるように書き換えた．

Single-task 実験対象にしたミニゲームは MoveToBeacon，CollectMineralShards，FindAndDefeatZerglings，DefeatRoaches である．すべての Single-task 実験は 1 learner，32 actors で設定して，CPU: AMD Ryzen Threadripper 2990WX と GPU: GeForce GTX2080Ti 2 枚を持っている計算機サーバー 2 台を用いて，1 台に 1 learner と 16 actors を配置し，も 1 台に 16 actors を配置することで動かした．

ミニゲームを研究対象として選ぶ原則は 4 種類のタスクから少なくとも 1 種類ずつ選ぶことである．CollectMineralsAndGas にも実験したが，回復できない擬似勾配消失現象を観測した．その現象のため，CollectMineralsAndGas と他のミニゲームの multi-task 実験にも顕著なパフォーマンス落ちを観察したので，CollectMineralsAndGas を今回の実験対象から除外した．ゲームの流れから見ると，BuildMarines は CollectMineralsAndGas の上で作られたものなので，擬似勾配消失現象を発生する可能性が高い，故に，BuildMarines も今回の実験対象から除外した．Fighting task の DefeatZerglingsAndBanelings は他の Fighting tasks と比べると，特に特徴がないので，実験対象から除外した．

学習時の episode return は**図 1** になる．パラメータ，部分 observation などの要素を考えると，文献 [2] のパフォーマンスを完全に再現するのは難しいことである．但し，この実験によって，ミニゲーム MoveToBeacon 以外，他の

[*1] www.github.com/deepmind/scalable_agent

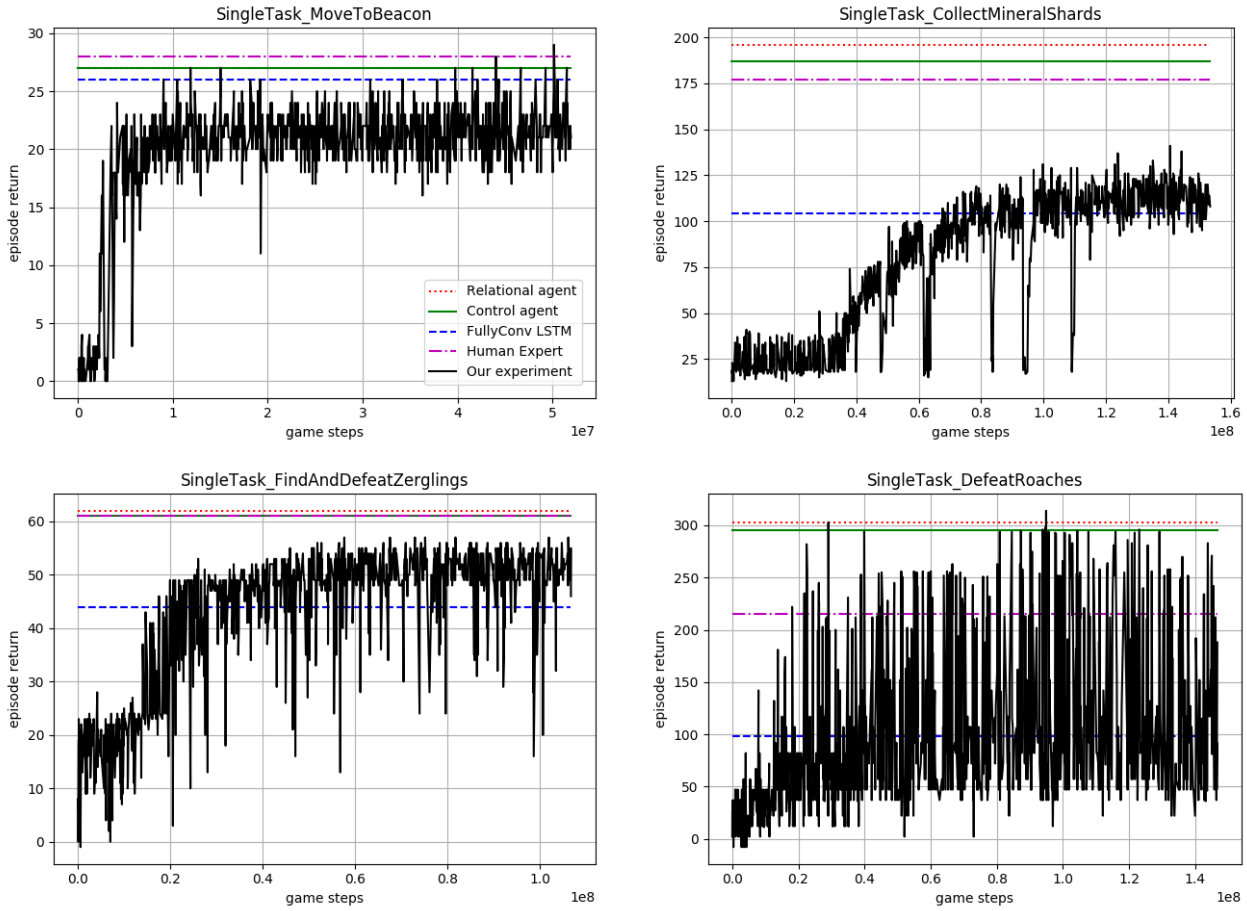


図 1　シングルタスクの学習曲線

Fig. 1　Learning curve of single-task trainning.

ミニゲームでのパフォーマンスは文献 [1] を超えたので，control agent の有用性を証明した．これらのシングルタスク実験結果を baseline にして，マルチタスク実験の対照対象になる．

3.2　Multi-task

タスク間の関連性は科学的に定義するのは難しいことである [5]．ミニゲームを分析するために，本稿では observation space，reward，action space から議論する．

始めは，observation space に関して，すべてのミニゲームは同じ spatial features(Minimap: 5 feature layers, Screen: 7 feature alyers) と non-spatial features(General player information) を共有している．

次は，reward に関して，

(1) MoveToBeacon:ユニットを操作して，指定座標に移動させると+1.

(2) CollectMineralShards:1 単位の Mineral Shard を回収できると+1.

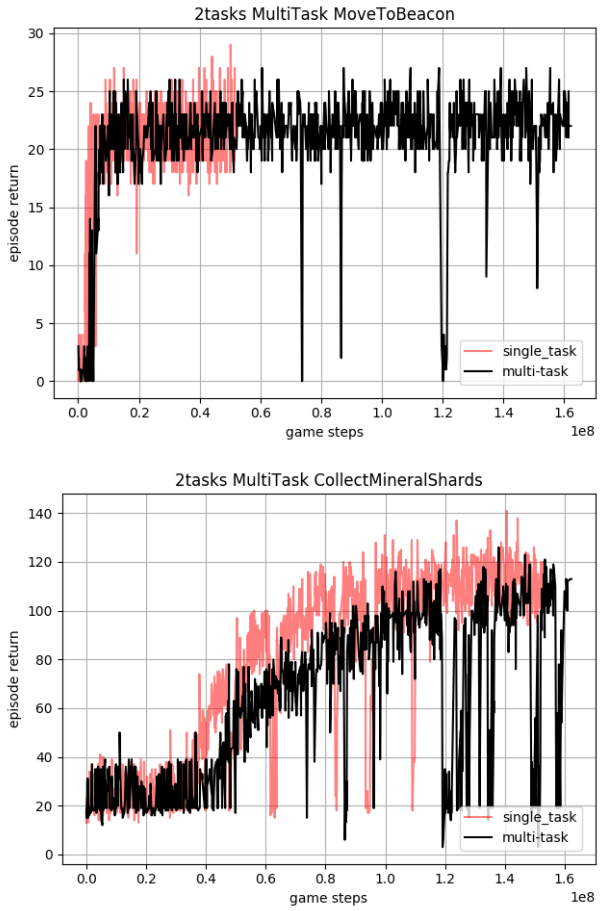
(3) FindAndDefeatZerglings:敵ユニット 1 体を倒したら+1，見方ユニット 1 体が倒されたら-1.

(4) DefeatRoaches:敵ユニット 1 体を倒す時+10，見方ユニット 1 体が倒されると-1 である．

FindAndDefeatZerglings と DefeatRoaches の reward はマイナスになる可能性がある．DefeatRoaches では敵 1 体を倒す時+10 なので，他三種のミニゲームの何倍以上の reward が出現する可能性がある．故に，FindAndDefeatZerglings および DefeatRoaches はマルチタスク訓練に加えると，全体的のパフォーマンスに影響を与えることを予想する．

最後は，actions space に関して，設定的には full action space(全 549 種 action) を使うが，ゲームのルールではユニット毎の available action list(unavailable と定義された action に対して，その action に対応するニューラルネットワーク出力が 0 にマスクされる) が違う．分析の結果はこの 4 種のミニゲームの中に，MoveToBeacon だけ，ID 番号 5 の action が使えない以外，全部一致する．実験中，ID 番号 5 の action が選択された数を追跡した．

全部のマルチタスク実験は 1 learner で， 1task 毎に 16 actors で設定した．CPU: AMD Ryzen Threadripper 2990WX と GPU: GeForce GTX2080Ti 2 枚を持っている計算機サーバー複数台を用いて，1 台に 1 learner と 16 actors を配置し，残りは 16 actors を配置することで動かした．全てのシングルタスク実験は同じハイパーパラメー



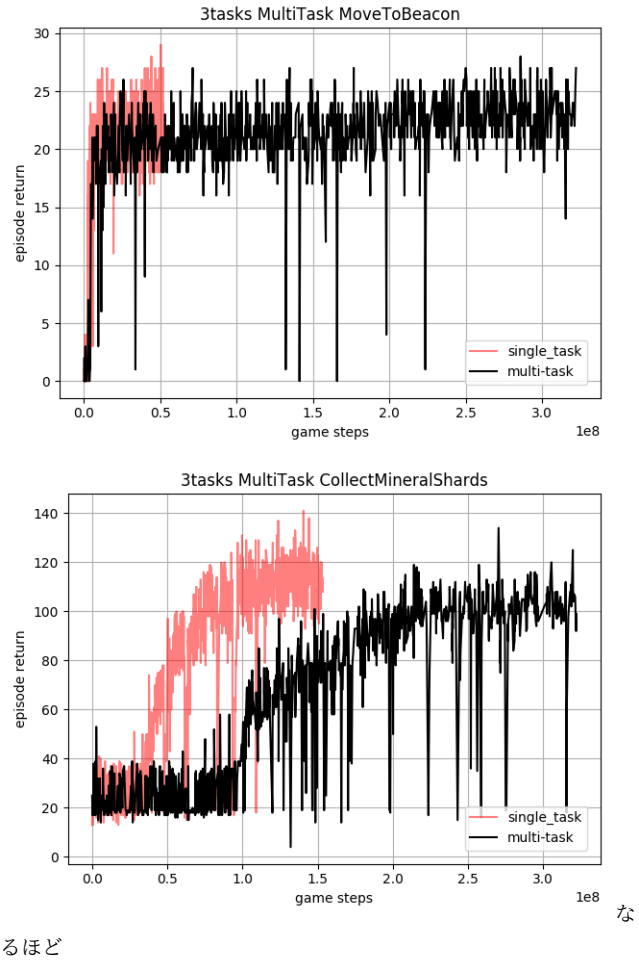


図 2　2 タスクのマルチタスクの学習曲線

Fig. 2　Learning curve of 2 tasks multi-task trainning.

タを使われているので，population based training(PBT) を使わず，マルチタスク実験を行った．

3.2.1　2 タスクのマルチタスク学習

始めに，reward の相似性が高い CollectMineralShards と MoveToBeacon で，マルチタスク実験を行った．学習曲線は図 **2** になる．パフォーマンスから見ると，二つのミニゲームでシングルタスクと相当する性能が得られた．さらに，パフォーマンスが上昇始める step 数もほぼ同じである．CollectMineralShards では，40 million game steps ぐらいから上り始めて，120 million game steps ぐらい限界にたどり着いた．MoveToBeacon では，2 million game steps ぐらいから上り始めて，3 million game steps ぐらい限界にたどり着いた．つまり，2 タスクのマルチタスク学習はシングルタスクと比べて，学習が遅くなるわけではない．しかし，CollectMineralShards の結果で，110 million game steps から 130 million game steps までのパフォーマンスが多少不安定と観測した．

3.2.2　3 タスクのマルチタスク学習

次に，3 タスクのマルチタスク実験のミニゲームは CollectMineralShards，MoveToBeacon，FindAndDefeatZerglings なるほど

図 3　3 タスクのマルチタスクの学習曲線

Fig. 3　Learning curve of 3 tasks multi-task trainning.

である．図 **3** に，3 タスクのマルチタスク学習曲線とシングルタスク学習曲線を示す．MoveToBeacon 以外のミニゲームの学習はシングルタスクと比べると，かなり遅くなった．実装した control agent のシングルタスクと比べると，マルチタスクのパフォーマンスが劣化したが，MoveToBeacon 以外，A3C [3] のシングルタスク実験結果より高いである．

3.2.3　4 タスクのマルチタスク学習

最後に，CollectMineralShards，MoveToBeacon，Find-

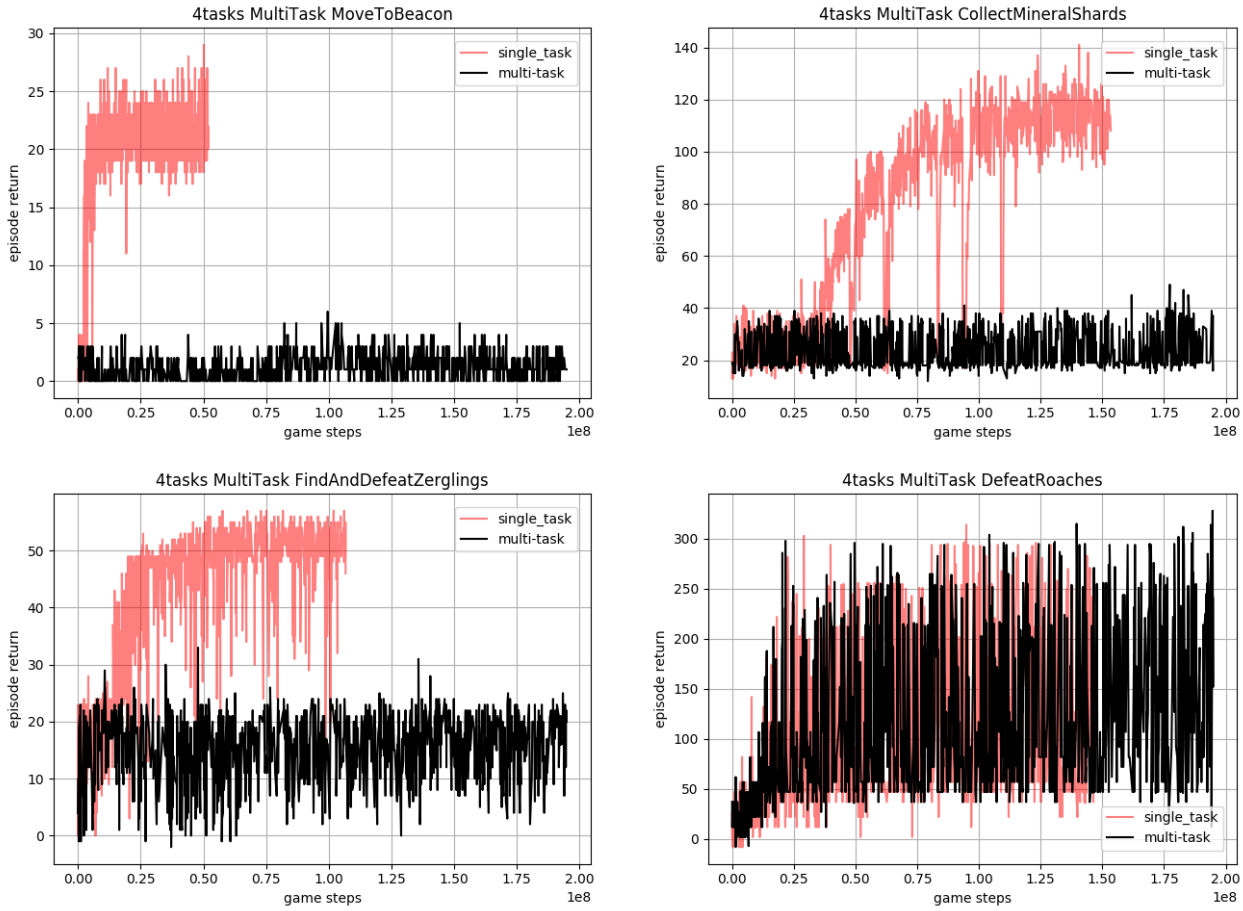


図 4　4 タスクのマルチタスクの学習曲線

Fig. 4　Learning curve of 4 tasks multi-task trainning.

図 5　3 タスクと 4 タスクのマルチタスク学習段階の平均 reward

Fig. 5　Mean reward in learning step from 3 tasks and 4 task multi-task.

AndDefeatZerglings，DefeatRoaches で 4 タスクのマルチタスク実験を行った．図 **4** に学習曲線を示す．4 つのタスクの中に，DefeatRoaches しか学習できなかった．その原因は前文で分析したように，他のミニゲームより何倍の reward を用いる DefeatRoaches は学習に対する影響が大きい．推測を証明するために，図 **5** に 3task と 4task のマルチタスクで，learner が学習を行う時集めった reward の平均値を示す．DefeatRoaches を加えると，reward の平均値を 4 倍ぐらい増えた．更に，学習の時の reward の最大値を記録した，その結果は 3 タスクが $[0, 5]$，4 タスクが $[2, 30]$ である．

3.2.4　ID 番号 5 の action の比率の分析

MoveToBeacon で ID 番号 5 の action が使えないことがマルチタスク学習に対する影響を分析するために，図 **6** にシングルタスク，2 タスク のマルチタスクの学習ステップで集めた ID 番号 5 の action の比率 (R) を示す．CollectMineralShards のシングルタスクとマルチタスクの ID 番号 5 の action の比率比較である．

$$R = \frac{the\ number\ of\ ID\ 5\ action}{unroll\ length * batch\ size * \alpha} \qquad (1)$$

α について，理論的にはすべての actors は同じ確率で learner にデータを送る．例えば，2 タスクのマルチタスク設定だと，CollectMineralShards，MoveToBeacon それぞれ同じ actor 数 (16) を持っているので，ID 番号 5 の action が使えるミニゲームは CollectMineralShards であり，ID 番号 5 の action が使えないミニゲームは MoveToBeacon であ

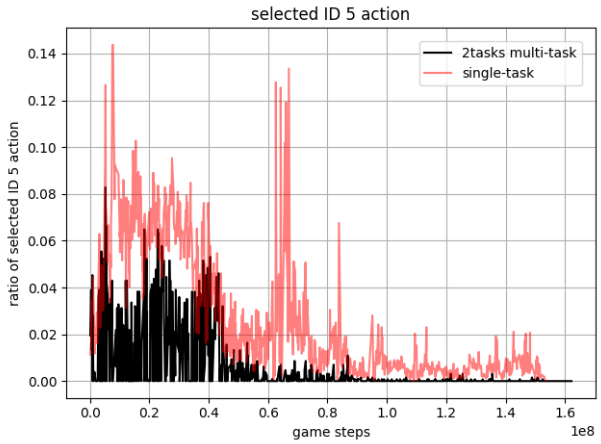


図 6 シングルタスク，2 タスクの学習段階の ID 番号 5 の action の比率比較

Fig. 6 Ratio of selected ID 5 action in learning step from single-task and 2tasks multi-task.

る．その時 $\alpha = \frac{1}{2}$．α の計算式は以下になる．

$$\alpha = \frac{the\ number\ of\ actors\ which\ can\ take\ ID\ 5\ action}{the\ number\ of\ actors} \tag{2}$$

もし MoveToBeacon の影響がなかったら，図 6 の赤線と黒線は近い数値になるはずだが，実際にマルチタスクの方は半分ぐらい減った．パフォーマンスに対する影響について，ID 番号 5 の action を使うと episode return が必ず上昇する証拠がないので，何ともいえない．

3.3 まとめ

本稿では, スタークラフト II ミニゲームでのマルチタスク性能を検証した．シングルタスクとマルチタスクの学習の対照実験により, マルチタスク学習に含まれるミニゲームの種類を変わるとパフォーマンスに対する影響を示した．シングルタスクでの実験結果は文献 [1] を超えたので，control agent はシングルタスクで有用であることを示した．但し，既存研究のパフォーマンスを完全再現ことは困難である．マルチタスクについて，2 タスクと 3 タスクのマルチタスクで上手く学習できたので，control agent はミニゲームにおけるマルチタスク学習の有効性を示した．改善点について，タスク種類を増えると学習速度を遅くなり，パフォーマンスはシングルタスクと比べてある程度劣化した点と，reward のスケールと分布が大きく変わるタスクが存在すると，他のタスクは完全に学習できなくなる点である．

今後の課題として，まずは PopArt を応用することで，4 タスクのマルチタスクでの学習が上手くいかない状態を改善できると期待している．次は，ルールから見ると CollectMineralsAndGas は BuildMarines の subtask になるので，CollectMineralsAndGas を含まれたマルチタスク学習は BuildMarines の学習にポジティブな影響があるかどうか検証すると，将来のフルゲーム実験に有意義である．今回の実験結果から，異なるタスクを同時に学習することはパフォーマンスに対してほとんどマイナスの影響であることは今後の研究方向として考慮している．

参考文献

[1] Oriol, V., Timo, E. et al.: Starcraft ii: A new challenge for reinforcement learning, *arXiv preprint arXiv:1708.04782* (2017).
[2] Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E. et al.: Deep reinforcement learning with relational inductive biases (2018).
[3] Mnih, V. et al.: Asynchronous methods for deep reinforcement learning, *International conference on machine learning*, pp. 1928–1937 (2016).
[4] Espeholt, L. et al.: Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures, *arXiv preprint arXiv:1802.01561* (2018).
[5] Ruder, S.: An overview of multi-task learning in deep neural networks, *arXiv preprint arXiv:1706.05098* (2017).
[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, pp. 5998–6008 (2017).
[7] : starcraftgym, http://starcraftgym.com/.
[8] van Hasselt, H. P., Guez, A., Hessel, M., Mnih, V. and Silver, D.: Learning values across many orders of magnitude, *Advances in Neural Information Processing Systems*, pp. 4287–4295 (2016).
[9] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, p. 529 (2015).
[10] Van Hasselt, H., Guez, A. and Silver, D.: Deep reinforcement learning with double q-learning, *Thirtieth AAAI conference on artificial intelligence* (2016).
[11] Hessel, M., Soyer, H., Espeholt, L., Czarnecki, W., Schmitt, S. and van Hasselt, H.: Multi-task deep reinforcement learning with popart, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 3796–3803 (2019).

付　　録

表 **A·1**　ハイパーパラメータ
Table A·1　Hyperparameter.

Hyperparameter	Value
Conv2DLSTM	
Output channels	96
Kernel shape	(3, 3)
Stride	(1, 1)
Conv2DTranspose	
Output channels	16
Kernel shape	(4, 4)
Stride	(2, 2)
Discount (γ)	0.99
Batch size	32
Unroll Length	80
Baseline loss scaling	0.1
Entropy loss scaling	1e-2
Clip global gradient norm	100.0
Adamβ_1	0.9
Adamβ_2	0.999
Adamϵ	1e-8
Adam learning rate	1e-4

表 **A·2**　PYSC2 の環境設定
Table A·2　Parameters for PYSC2.

Parameter	Value
Feature screen size	32
Feature minimap szie	32