UNREALにおける補助タスクの適応的選択

Adaptive selection of auxiliary tasks in UNREAL

板谷 英典 *1　平川 翼 *1　山下 隆義 *1　藤吉 弘亘 *1
Hidenori Itaya　Tsubasa Hirakawa　Takayoshi Yamashita　Hironobu Fujiyoshi

*1中部大学
Chubu University

Deep reinforcement learning has a difficulty to solve a complex problem because such problem consists of a larger state space. To solve this problem, Unsupervised Reinforcement learning and Auxiliary Learning (UNREAL) has been proposed, which uses several auxiliary tasks during training. However, all auxiliary tasks might not perform well on each problem. Although we need to carefully design these tasks for solving this problem, it requires significant cost. In this paper, we propose an additional auxiliary task, called auxiliary selection. The proposed method can adaptively select auxiliary tasks that contributes the performance improvement. Experimental results with DeepMind Lab demonstrate that the proposed method can select appropriate auxiliary tasks with respect to each game tasks and efficiently train a network.

1. はじめに

強化学習とは，数値化された報酬を最大とするために，何をすべきかを学習する問題である．また，教師あり学習のように，どのような行動を選択すれば良いかは教えられず，どの行動を選択すればより良い結果に結び付くかを見つけ出す問題となっている．

強化学習は，教師あり学習のように教師信号を用意する必要がない点などから，ロボット制御 [Gu 17] [Rajeswaran 17] やゲーム攻略 [Justesen 17] [Firoiu 17] などの様々なタスクに応用されている．ゲーム攻略については，Silver ら [Silver 16] のコンピュータ囲碁プログラム AlphaGo がプロ囲碁棋士に勝利し，非常に注目された．また，Atari2600 のゲーム攻略において，深層強化学習手法の一つである Deep Q-Network (DQN) [Mnih 15] と呼ばれる手法が提案され，人間を凌駕するスコアを達成した．DQN は Q 学習 [Watkins 92] と Deep Convolutional Neural Network (DCNN) を組み合わせた手法であり，画像を入力とする Atari2600 のゲームのように状態数の多い問題を扱うことを可能にしている．この DQN 以降，強化学習手法は深層学習を組み合わせた深層強化学習が主流となった．

強化学習における学習データは，エージェントが環境を探索し収集する．そのため，学習に寄与するデータを獲得するために時間を要するという問題がある．そこで，この問題を解決するために Asynchronous Advantage Actor-Critic (A3C) [Mnih 16] が提案されている．A3C は，学習で用いる経験の生成を並列に実行することで高速化し，パラメータの更新を非同期的に行う手法である．また，A3C をベースとし，教師なし学習の補助タスクをメインタスクと並列に実行する UNspervised REinfrocsment learning and Auxiliary Learning (UNREAL) [Jaderberg 16] が提案されている．UNREAL は，複数の補助タスクを導入することによって，ゲームタスクにおいて A3C より高いスコアを達成している．しかし，UNERAL で用いられる全ての補助タスクは，あらゆる環境において必ずしも有効であるとは限らない．また，補助タスクを用いることでメインタスクの学習を妨げるという問題が存在する．そのため，補助タスクは環境に応じて適切に設計する必要があるが，適切な補助タスクの設計は多大な手間と時間を要する．

そこで本研究では，UNREAL の補助タスクに着目し，補助タスクを環境に合わせ適応的に選択するタスク Auxiliaty Selection を導入することで，上記の問題を解決する．最適な補助タスクの選択には，Auxiliaty Selection により出力された各補助タスクの重みと，各補助タスクとの損失関数の積を取ることで実現する．DeepMind Lab [Beattie 16] の 3 つのゲームを用いて，UNREAL および各補助タスクのみの場合とスコアを比較することで，本手法の有効性を示す．また，各補助タスクの選択回数を調査することで，最適な補助タスクを選択できているか確認する．

2. 関連研究

メインタスクと補助タスクを並列に学習させることでメインタスクの高精度化を図る手法は様々提案されている．Liebel ら [Liebel 18] は，自動車の運転シーンにおいて，セマンティックセグメンテーションと深度推定をメインタスクとし，時刻と天候推定の補助タスクを並列に実行することでメインタスクの精度向上を実現している．Jaderberg ら [Jaderberg 16] は，深層強化学習において，ベースである A3C [Mnih 16] に加えて，教師なし学習の補助タスクをメインタスクと並列に実行する手法を提案している．この手法は，3 つの異なる補助タスクを用いることで，DeepMind Lab の迷路攻略タスクにおいて，高いスコアを獲得している．1 つ目の補助タスク Pixel Control は画像の画素が大きく変化する行動を学習するタスクである．2 つ目の補助タスク Value Function Replay は過去の経験をシャッフルし，状態価値関数 $V(s)$ を学習するタスクである．3 つ目の補助タスク Reward Prediction は報酬を獲得した経験を優先して学習し，未来の報酬を予測するタスクである．

しかし，上記の手法では，メインタスクに適していない補助タスクを用いた場合，メインタスクの学習を妨げるという問題が存在する．そのため，補助タスクの導入にはメインタスクに

連絡先:
板谷 英典：itaya@mprg.cs.chubu.ac.jp
平川 翼：hirakawa@mprg.cs.chubu.ac.jp
山下 隆義：yamashita@cs.chubu.ac.jp
藤吉 弘亘：hf@cs.chubu.ac.jp

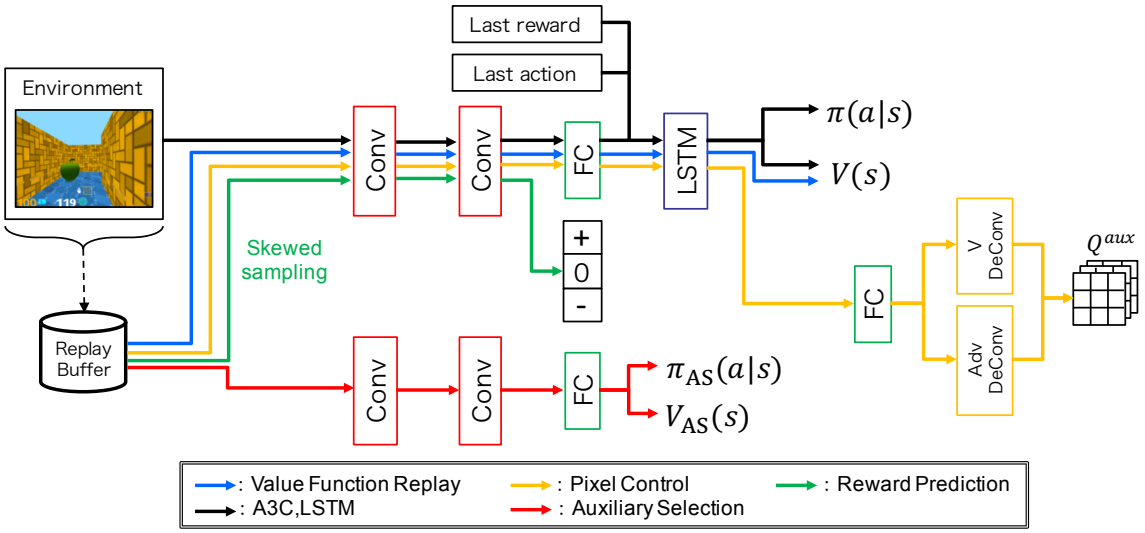


図 1: 提案手法のネットワーク構成

適したタスクを導入する必要がある．そこで，深層強化学習において，上記の問題を解決する手法がいくつか提案されている．Teh ら [Teh 17] は，蒸留によりタスク間に共通する行動を捉えた方策を獲得し，学習の妨げを回避することで，学習を安定させる手法を提案した．この手法は，共有する方策が全タスクにおいて有効な方策から離れないような制約を導入することで，異なるタスク間での頑健性と安定性を獲得している．一方，提案手法では，メインタスクの学習に有効な補助タスクを適応的に選択することで，メインタスクの学習を妨げることなく，学習の効率化を行う．

Riedmiller ら [Riedmiller 18] は，どの補助タスクの方策を使用すれば，メインタスクを解決できるか学習する Scheduled Auxiliary Control (SAC-X) を提案している．この手法は，各補助タスクを低レベルのタスクとして設計し，それぞれの目的に沿った方策を学習する．そして，SAC により用いる方策を選択することで，ロボットアームのような報酬が疎である問題を解決している．一方，提案手法における補助タスクは方策を必要としないため，UNREAL における Value Function Replay などが使用可能である．また，メインタスクである A3C の学習に対して，適応的に補助タスクを選択するため，補助タスクを環境に左右されず設計が可能である．

3. 提案手法

UNREAL の補助タスクは，環境によって有効性が異なるため，メインタスクの学習を妨げるという問題がある．そのため，環境に合わせた補助タスクの選択が求められる．本研究では，環境に合わせて用いる補助タスクを適応的に選択するタスク Auxiliary Selection を提案する．

3.1 Auxiliary Selection

図 1 に Auxiliary Selection を導入した UNREAL のネットワーク構成を示す．UNREAL の 3 つの補助タスクは，Pixel Control (PC)，Value Function Replay (VR)，Reward Prediction (RP) である．Auxiliary Selection には，Replay Buffer 内に格納された画像を入力し，状態価値関数 $V_{\mathrm{AS}}(s)$ と方策 π_{AS} を出力する．方策 π_{AS} は各補助タスクを用いるかどうかを表す値である．各補助タスクに対する重みを $C_{\mathrm{PC}} = \{0, 1\}$，$C_{\mathrm{VR}} = \{0, 1\}$，$C_{\mathrm{RP}} = \{0, 1\}$ とするとき，$\pi_{\mathrm{AS}} = (C_{\mathrm{PC}}, C_{\mathrm{VR}}, C_{\mathrm{RP}})$ となる．Auxiliary Selection のネットワークは，畳み込み層 2 層と全結合層 1 層から構成される．また，他の補助タスクとは異なり，A3C のネットワークとは共有せず，独立したネットワークとして学習を行う．このように，環境に合わせて補助タスクを適応的に選択することで，補助タスクを設計する際の効率化を図る．

3.2 損失関数

提案手法の損失関数 L_{proposed} は，従来の UNREAL の損失関数をもとに設計し，式 (1) のように定義する．

$$L_{\mathrm{proposed}} = L_{\mathrm{A3C}} + C_{\mathrm{VR}} L_{\mathrm{VR}} + C_{\mathrm{PC}} \sum_{c} L_{Q}^{(c)} + C_{\mathrm{RP}} L_{\mathrm{RP}} \quad (1)$$

提案手法では，Auxiliary Selection から獲得する C_{VR}，C_{PC}，C_{RP} と各補助タスクの損失関数の積を取ることで，最適な補助タスクのみを用いた学習を実現する．

また，提案手法の損失関数 L_{propoped} には，Auxiliary Selection から獲得する $C_{\mathrm{VR}}, C_{\mathrm{PC}}, C_{\mathrm{RP}}$ を用いている．そのため，Auxiliary Selection の学習を他の補助タスクと同様に，L_{proposed} に基づいて行うと，各補助タスクの重み $C_{\mathrm{VR}}, C_{\mathrm{PC}}, C_{\mathrm{RP}}$ が 0 になるように学習されるという問題がある．したがって，Auxiliary Selection の学習では，L_{proposed} とは異なる Auxiliary Selection の損失関数を定義し，A3C と各補助タスクのネットワークとは独立して学習を行う．

Auxiliary Selection の損失関数は状態価値関数と方策の損失関数で表すことができる．状態価値関数の損失関数 $L_{\mathrm{AS}v}$ を式 (2)，方策の損失関数 $L_{\mathrm{AS}p}$ を式 (3) に示す．ここで，θ^{-} は更新前のネットワークのパラメータである．また，$H(\pi_{\mathrm{AS}})$ は局所的な最適解に収束しないように，探索を促進するためのエントロピーであり，β はエントロピーの正則化項の強さを制御するパラメータである．

$$L_{\mathrm{AS}v} = (r + \gamma V_{\mathrm{AS}}(s_{t+1}, \theta^{-}) - V_{\mathrm{AS}}(s_t, \theta))^2 \quad (2)$$

$$L_{\mathrm{AS}p} = -\log(\pi_{\mathrm{AS}}(a|s)) A(s, a) - \beta H(\pi_{\mathrm{AS}}) \quad (3)$$

Auxiliary Selection の損失関数は，式 (2) の状態価値関数の損失関数と式 (3) の方策の損失関数の和によって表される．式 (4) に Auxiliary Selection の損失関数 L_{AS} を示す．

$$L_{\mathrm{AS}} = L_{\mathrm{AS}v} + L_{\mathrm{AS}p} \quad (4)$$

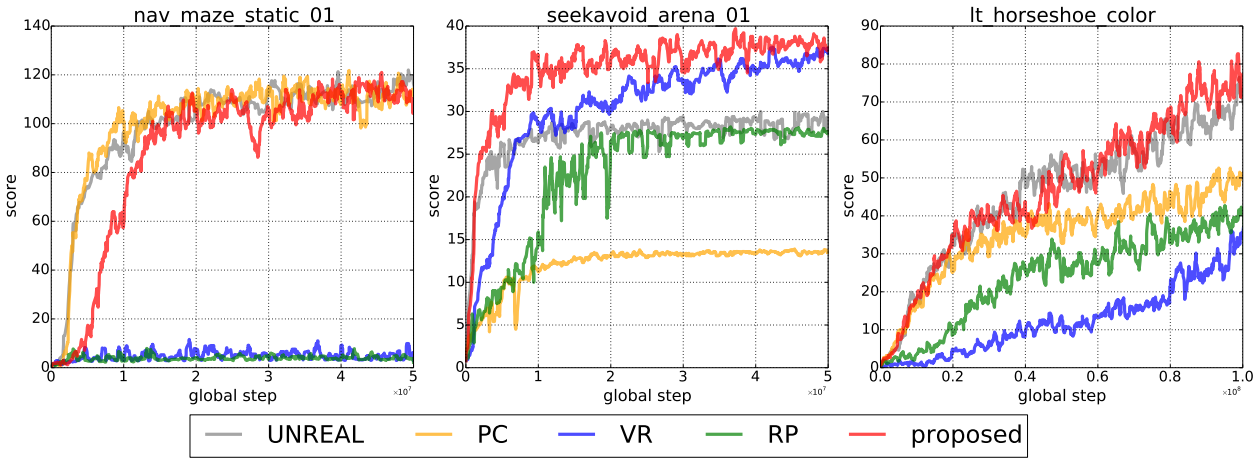
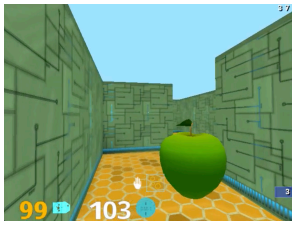
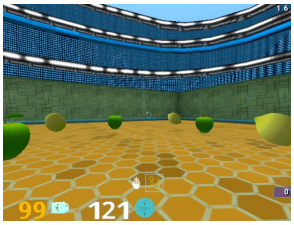


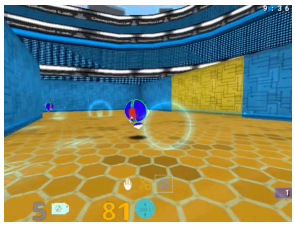
図 2: DeepMind Lab におけるステップ毎のスコア



(a) nav_mazze_static_01



(b) seekavoid_arena_01



(c) lt_horseshoe_color

図 3: DeepMind Lab の各ゲーム画面

4. 評価実験

4.1 実験環境

本論文では，実験環境として，DeepMind Lab [Beattie 16] を用いる．DeepMind Lab は，一人称視点映像の 3D ゲーム環境であり，大きく分けて nav_maze_static_01 (maze)，seekavoid_arena_01 (seekavoid)，lt_horseshoe_color (horseshoe) の 3 つのゲームが存在する．

maze は一人称視点の迷路探索ゲームである．道中にあるリンゴを獲得すると+1，ゴールに到達すると+10 のスコアを獲得することが可能であり，時間内に獲得できたスコアを競うゲームである．エージェントが取りうる行動は，左視点移動，右視点移動，前へ進む，後ろへ進む，左に平行移動，右に平行移動の計 6 つである．

seekavoid はステージ内の特定の物体を集めるゲームである．リンゴを獲得すると+1，レモンを獲得すると-1 のスコアを獲得し，時間内に獲得できたスコアを競う．行動については，左視点移動，右視点移動，前へ進む，後ろへ進む，左に平行移動，右に平行移動の計 6 つである．

horseshoe は一人称視点シューティングゲームである．ステージ内にスポーンする敵をレーザーで攻撃し倒すと+1 のスコアを獲得することが可能であり，時間内で獲得したスコアを競う．行動については，左視点移動，右視点移動，前へ進む，後ろへ進む，左に平行移動，右に平行移動，攻撃の計 7 つである．

4.2 実験概要

前述の DeepMind Lab の 3 つのゲームにおいて，ステップ毎のスコアを比較することで提案手法の有効性を確認する．比較手法として，全補助タスクを用いた場合 (UNREAL)，Pixel Control のみの場合 (PC)，Value Function Replay のみの場合 (VR)，Reward Prediction のみの場合 (RP)，提案手法 (proposed) の 5 通りで学習を行う．学習時の各ハイパーパラメータは変更せず，maze および seekavoid では 50,000,000 ステップ，horseshoe では 100,000,000 ステップまで学習を行う．また，1 エピソード内での各補助タスクの選択回数を調査することで，最適な補助タスクの選択が実現できているか確認する．worker 数は 8 で行う．

4.3 実験結果

4.3.1 nav_maze_static_01

nav_maze_static_01 におけるステップ数毎のスコアを図 2 左に示す．maze においては，UNREAL と Pixel Control が約 110 の高いスコアを獲得している．また，Value Function Replay と Reward Prediction では 0 に近く，全くスコアを獲得できていないことが分かる．これは，壁の端の画素が大きく変化する行動を Pixel Control によって獲得することが可能であり，この行動が maze の迷路攻略に有効である為だと考えられる．したがって，maze では UNREAL か Pixel Control が最適な補助タスクの選択であると言える．

提案手法は，UNREAL と Pixel Control と同様のスコアを獲得できることが分かる．これら 2 つの結果から，maze において，提案手法は最適な補助タスクと同様に高いスコアを獲得していることが確認できる．

4.3.2 seekavoid_arena_01

seekavoid_arena_01 におけるステップ数毎のスコアを図 2 中央に示す．seekavoid においては，UNREAL と比較し，Value Function Replay が UNREAL を約 10 スコア上回っていることが分かる．また，maze で有効な Pixel Control は約 14 であり，高いスコアを獲得できていない．これは，ゲーム画面の画素が大きく変化する行動がゲーム攻略とは適しておら

表 1: 1 エピソードにおける補助タスクの選択回数

環境 \ 補助タスク	PC	VR	RP
maze	435.4 (48.3%)	487.8 (54.1%)	369.0 (41.0%)
seekavoid	0.3 (0.1%)	300.0 (100.0%)	0.0 (0.0%)
horseshoe	8545.1 (94.9%)	14.1 (0.1%)	8998.2 (99.9%)

ず，報酬が密に獲得できるゲームである為，Pixel Control と Reward Prediction が有効ではないと考えられる．したがって，seekavoid では Value Function Replay が最適な補助タスクの選択であると言える．

提案手法は，Value Function Replay と同様のスコアを獲得できることが分かる．これら 2 つの結果から，seekavoid において，提案手法は最適な補助タスクと同様に高いスコアを獲得していることが確認できる．

4.3.3 lt_horseshoe_color

lt_horseshoe_color におけるステップ数毎のスコアを図 2 右に示す．horseshoe において，UNREAL が約 75 で最もスコアが高く，各補助タスクのみでは Pixel Control が約 50 で高いスコアを獲得している．これは，敵を倒す行動が画素を大きく変化させる行動にあたる為，各補助タスクのみでは Pixel Control が最も有効であったと考えられる．したがって，horseshoe では UNREAL が最適な補助タスクの選択であると言える．

提案手法は，UNREAL と同様のスコアを獲得できることが分かる．これら 2 つの結果から，horseshoe において，提案手法は最適な補助タスクと同様に高いスコアを獲得していることが確認できる．

4.3.4 選択された補助タスクの解析

各ゲームの 1 エピソードにおける補助タスクの選択回数を表 1 に示す．ここで，選択回数とは 50 エピソード間の平均の選択回数であり，括弧内は 1 エピソード内で選択する割合を表す．1 エピソードの総ステップ数は，maze では 900，seekavoid では 300，horseshoe では 9,000 である．seekavoid では最適な補助タスクである Value Function Replay，horseshoe では Pixel Control と Reward Prediction を安定して選択し，maze では全ての補助タスクが同等に選択されている．maze において，UNREAL と同様に全ての補助タスクを選択するため，最適な補助タスクである UNREAL と同等のスコアを獲得したと考えられる．したがって，UNREAL に Auxiliary Selection を導入することで，環境に合わせた補助タスクを選択でき，効率的な学習を実現していると言える．

5. おわりに

本研究では，学習に用いる補助タスクを適応的に選択するタスク Auxiliary Selection を提案した．提案手法では，各補助タスクの損失関数と Auxiliary Selection により出力する重みの積を取ることで，学習時における最適な補助タスクの選択を実現した．これにより，環境に合わせた補助タスクを設計する必要がなく，補助タスクを用いた学習において，効率化することが可能である．DeepMind Lab を用いた実験により，効率的に学習できることを示した．今後の予定としては，異なる環境や多様な補助タスクを導入した場合における提案手法の有効性の調査などが挙げられる．

参考文献

[Beattie 16] Beattie, C., Leibo, J. Z., *et al.*: DeepMind Lab, *arXiv preprint, arXiv:1612.03801* (2016)

[Firoiu 17] Firoiu, V., Whitney, W. F., *et al.*: Beating the World's Best at Super Smash Bros. Melee with Deep Reinforcement Learning, *arXiv preprint, arXiv:1702.06230* (2017)

[Gu 17] Gu, S., Holly, E., *et al.*: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in *ICRA*, pp. 3389–339 (2017)

[Jaderberg 16] Jaderberg, M., Mnih, V., *et al.*: Reinforcement Learning with Unsupervised Auxiliary Tasks, *arXiv preprint, arXiv:1611.05397* (2016)

[Justesen 17] Justesen, N., Bontrager, P., *et al.*: Deep Learning for Video Game Playing, *arXiv preprint, arXiv:1708.07902* (2017)

[Liebel 18] Liebel, L. and Körner, M.: Auxiliary Tasks in Multi-task Learning, *arXiv preprint, arXiv:1805.06334* (2018)

[Mnih 15] Mnih, V., Kavukcuoglu, K., *et al.*: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015)

[Mnih 16] Mnih, V., Badia, A. P., *et al.*: Asynchronous Methods for Deep Reinforcement Learning, in *ICML*, pp. 1928–1937 (2016)

[Rajeswaran 17] Rajeswaran, A., Kumar, V., *et al.*: Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations, *arXiv preprint, arXiv:1709.10087* (2017)

[Riedmiller 18] Riedmiller, M., Hafner, R., *et al.*: Learning by Playing – Solving Sparse Reward Tasks from Scratch, in *ICML* (2018)

[Silver 16] Silver, D., Huang, A., *et al.*: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, No. 7587, p. 484 (2016)

[Teh 17] Teh, Y., Bapst, V., *et al.*: Distral: Robust multitask reinforcement learning, in *NIPS*, pp. 4496–4506 (2017)

[Watkins 92] Watkins, C. J. and Dayan, P.: Q-Learning, *Machine learning*, Vol. 8, No. 3-4, pp. 279–292 (1992)