一時的好奇心に基づく内発的報酬設計を用いた強化学習によるローグライクゲームの学習

加賀谷昂輝[1]　鶴岡慶雅[2]

概要：ゲーム AI を評価する環境の１つとして, ローグライクと呼ばれるダンジョン探索型環境がある. ローグライク環境は, 初期値にランダム性がある点, 報酬が疎である点, 部分観測しかできない点など, 難易度の高い環境であると同時に現実環境により近く, 興味深い環境でもあるため, これに対する学習手法を検討する. 先行研究では好奇心による内部報酬を用いた手法などで高い成果を得られることが検証されているが, 同時に探索済み状態を過剰に避けるなどの環境特有の問題点も指摘されており, 本研究ではそれを改善できることが期待できる手法を提案する. 評価実験では, ローグライク環境において３種類の報酬設計で学習を行い, 提案手法での内発的報酬設計においてより学習が促進されていることを確認した.

Reinforcement Learning in Rogue-like Games with Temporary Curiosity

Koki Kagaya[1]　Yoshimasa Tsuruoka[2]

Abstract: One of the environments in which game AI can be evaluated is a dungeon-exploration type environment called a rogue-like. The rogue-like environment is challenging since the initial states are random, the rewards are sparse, and only partial observation is possible. Previous studies have verified that methods using intrinsic rewards based on curiosity can produce high results, but at the same time, problems specific to the environment, such as excessive avoidance of already explored states, have been pointed out. We conducted evaluation experiments in a roguelike environment using three types of reward designs and confirmed that the proposed intrinsic reward design promotes learning more than the other two.

1. 導入

近年目覚ましい発展を遂げているゲーム AI の研究背景として, 成果の評価が比較的容易であり, AI の性能指標として利用しやすい点がある [1], [2]. これにより, ゲーム AI を通してより効果的なアルゴリズムの開発や, 評価の難しい現実問題への応用ができることが期待される. ゲーム AI の研究で用いられる環境としては, 将棋や囲碁といったボードゲームや, Atari 2600 と呼ばれる一連のゲームなどが多く用いられている. しかしこうした環境の多くで高い性能を発揮する手法であっても, より難易度の高い環境において学習が進まない場合がある. その要因の１つとして, そうした環境から得られる報酬が疎であるという点がある. 実際の現実的環境を仮定した場合でも, 報酬は疎である場合も多く, 従来の環境で開発されてきた手法が現実的問題において十分に成果を上げることは難しいことが指摘されている [3]. こうした背景から, より多くの環境に応用可能な手法を考える上では, 従来のよく用いられる環境だけではなく, より現実に近い構造を持つ環境での研究を行うことが望ましい. そこで, 本研究ではそうした環境としてローグライクゲームの環境に取り組む.

ローグライクゲームは 1980 年頃開発された Rogue をはじめとするダンジョン探索型ゲームであり, 様々な性質か

[1] 東京大学工学部電子情報工学科
Department of Information and Communication Engineering
The University of Tokyo, Bunkyo, Tokyo 113 – 8654, Japan
[2] 東京大学大学院情報理工学系研究科電子情報学専攻
Graduate School of Information Science and Technology
The University of Tokyo, Bunkyo, Tokyo 113 – 8654, Japan

ら強化学習の題材として難易度が高いとともに非常に興味深いものとされている [2][2][4]. ローグライクゲームを対象とした強化学習に関する研究の中には, 高い成果を示したものもあったが, 一方でローグライク環境特有の性質から既存手法では学習が進まない場合があることも指摘されている [1]. 本研究では, こうしたローグライク環境の問題を既存手法を用いた手法により検証するとともに, 環境により適応した強化学習手法の提案を目指す.

2. 背景

2.1 強化学習

強化学習は, 機械学習の手法の 1 つである. 強化学習では, 環境が与えられ, その環境で行動を行い得られたフィードバックを基に学習を進めていくという方法をとる. 強化学習で用いられる環境はは, 基本的にマルコフ決定過程 (Markov Decision Process, MDP) で定式化される. MDP では, 遷移先の状態が現在の状態及びそこでの行動によってのみ決定され, 構成要素は以下のように表される.

- 状態の集合 : S
- 行動の集合 : A
- 状態遷移関数 : 状態 s で行動 a を行った時に状態 s' に遷移する確率 $T(s'|s,a)$
- 報酬関数 : 状態 s で行動 a を行い. 状態 s' に遷移した場合の報酬 $r(s,a,s')$

エージェントは, 各時刻 t において, 環境の状態 s_t を観測し, 方策 $\pi(a|s)$ に基づいて行動 a_t を選ぶ. ここで, $\pi(a|s)$ は状態 s において行動 a を選択する確率である. この行動に対して環境は遷移関数を基に状態遷移を行い, 報酬関数に従ってエージェントに報酬 $r(s_t,a_t,s_{t+1})$ を与える. 強化学習における目標は, こうして与えられる報酬の累計を最大化することである. 時刻 t 以降に得る累積報酬 R_t は, それ以降に受け取る報酬 $r_{t+1}, r_{t+2}, \ldots$, 及び割引率 γ を用いて以下のように表される.

$$R_t = r_{t+1} + \gamma r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \tag{1}$$

ここで, 割引率は $0 < \gamma < 1$ を満たし, 通常は 1 に非常に近い値を用いる. 割引率は, 直近の報酬をどれだけ重視するかを表すパラメータと言える.

2.2 内発的報酬

強化学習においては, 累積報酬の最大化が目的であったように, 環境から得られる報酬が学習の指針となる。しかし, 一部の環境においては, 学習が進行していないエージェントでは有効な報酬を得られる確率が極めて低い場合がある. このような環境は報酬が疎な環境といわれる. このような環境は, 行動に対するフィードバックが少なく学習が難しい. 例えば, 疎な報酬を持つゲームとして多く挙げられる Montezuma’s Revenge においては, DQN [5], A3C [6] などといった既存の手法の多くで有効な方策を得ることができなかった [2].

報酬が疎な環境での学習手法として, 学習を行うエージェントに外部報酬以外の報酬を与えることで擬似的にその問題を解決する方法がある. 単純な方法として, 環境からの外部報酬自体を改変する方法が考えられる. 例えば, 環境に予め設定されている目標タスクとは別に, エージェントの特定の行動に対して報酬を与えるようにすることが考えられる. しかし, こうした方法の場合, 外部報酬の獲得は直接的に最終目標となるタスクの達成を意味しないため, 報酬の与え方によっては全く学習に寄与しない可能性も考えられる. また仮に有効な報酬の設計が出来た場合でもタスクや環境に依存する面が大きすぎ, 広く有効な手法とは言い難い. これと異なるアプローチとして, 外部報酬とは別に内発的報酬を追加で与えるものがある. 内発的報酬としては, 未知の状態を探索することに価値を見出し, 新規性のある状態に到達した場合に与えられるようにする方法がある. 近年一般的になっているこのような内発的報酬設計の方法として, 好奇心をベースにした探索がある [7][8]. これは, エージェントが探索を行いながら環境の予測を行い, その予測誤差を内発的報酬として利用するものであり, これによって予測が十分な精度で行われていない部分への探索を促進できる.

2.3 Random Network Distillation (RND)

Random Network Distillation (RND) [7] は好奇心ベースの内発的報酬設計手法の 1 つである. RND は target-network と predictor-network の 2 つのニューラルネットワークを持つ. 環境から得られる状態 s_{t+1} を入力として, 2 つのネットワークからそれぞれ得られる特徴表現 $f_{t+1}(s_{t+1})$、$f'_{t+1}(s_{t+1})$ の二乗誤差である

$$r_t^i = ||f_{t+1}(s_{t+1}) - f'_{t+1}(s_{t+1})||^2 \tag{2}$$

が内発的報酬として利用される. このとき, target-network は学習を行わず初期値のまま保たれ, predictor-network は上記の予測誤差を最小化するように勾配降下法を用いて学習される. 最終的に, エージェントが得られる報酬は環境からの外部報酬 r_t^e を用いて

$$r_t^{all} = r_t^e \times c^e + r_t^i \times c^i \tag{3}$$

となる. ここで、c^e, c^i は報酬の重みの係数である. 学習の進行に伴って内発的報酬（≒ 予測誤差）が減少していくため, チューニングが難しい問題もあるが、これは内部報酬の実行推定標準偏差で内部報酬を割り正規化を行う [7] ことで改善される. これによって, target-network からの出力を predictor-network が正しく予測できているかどうかを新規性の基準とし探索が行われる. この手法は, 従来の

手法, 特に RND 以前の好奇心ベースの手法を用いても学習が困難であった Montezuma’s Revenge などの環境に対しても学習を促進できた [7].

2.4 ローグライク環境

本研究では環境として, ローグライク環境を扱う. ローグライクゲームは, 1980 年頃の Rogue から派生した, 不思議のダンジョンシリーズなどに代表されるダンジョン探索型ゲームの総称である. ゲームによりそのルールや仕様は様々だが, 多くのローグライクゲームに共通する特徴は以下のようなものが挙げられる.

- ランダムに生成された, 複数のフロアからなるダンジョンをプレイヤーが探索し, フロアに存在するゴールを発見することで次の階層へ移動することを繰り返す.
- ダンジョンのフロアはいくつかの部屋とそれを繋ぐ通路で構成される.
- プレイヤーは, フロアの全容を最初から知ることはできず, 自分が探索した範囲のみを把握できる.
- その他, 探索を有利に進めるアイテムや, 探索を妨害する敵などが出現する.

ローグライクゲームは人間がプレイするゲームとしても難易度の高いものが多いが, 実際強化学習を行う対象としても様々な要因から難易度の高いものとされる. これは, 以下のような要因による.

- **部分観測マルコフ決定過程 (POMDP)** フロアの構造が探索した範囲しか明らかにならず, 部分観測マルコフ決定過程 (Partialy Observable Markov Decision Process, POMDP) と呼ばれるモデルを持ち, こうした環境では学習難易度が著しく高くなる [9]。
- **環境のランダム性** フロア構造はフロアに到達した時点でランダムに決定され, 決まった初期値や構造を持たない. これによって, 特定の初期値やフロア構造にのみ適応した学習では成果を得られない. よって, ゴールへの到達方法や戦略をより一般的に捉える必要があり, 学習難度を上げる要因となっている.
- **報酬が疎** ローグライク環境は報酬が疎な環境である. 具体的には, ローグライク環境における報酬は基本的に, フロアに 1 カ所存在するゴール地点への到達によって得られるものである. ランダムな行動でゴールに到達することは多くの場合難しく, 従って探索の過程ではほぼ報酬を得ることができない。

以上の特徴は主にフロアでのゴール探索というタスクに関するものだが, 敵やアイテムといった要素が加わった場合は行動空間が大きくなることや, 長期的戦略（アイテムの節約など）が求められることも加わりさらに学習難易度は高くなる. 関連する先行研究の多くにおいては, 学習難易度を過剰に高めないため, ローグライクの全ての特徴を反映した環境を扱ってはおらず, 単純化した環境を用いているものが多い. 例えば, 敵やアイテムの要素を排除し, ゴールの探索というタスクに特化したものがあり [1][10], これは上記で特に挙げた 3 つの特徴に焦点を合わせた学習を行なっているものとも言える.

3. 関連研究

3.1 ローグライク環境に対する強化学習

ローグライク環境を対象とした研究のうち, 好奇心ベースの手法として加納ら [1] の研究では学習手法として Proximal Policy Optimization (PPO) [11] を用い, 報酬設計として RND を組み合わせたものを用いて, 敵やアイテムのない純粋な探索タスクに対して学習を行なっている. その研究においては, 独自に作成した環境を用い, 2 種類の入力サイズの環境に対して RND を用いた場合と用いない場合で学習を行なっている. 特に入力サイズの大きい複雑な環境において, RND を用いた場合に用いない場合と比較して高いクリア（＝ゴールへの到達）率を達成している. 学習の高速化のためフロア形状のパターンを予め生成したものから選ぶ方式をとっており完全にランダムなフロア生成を再現しているわけではないが, この結果からローグライク環境に対しても好奇心ベースの内発的報酬が有効に働くと考えることができる.

一方, 同研究においては RND を単に適用することに関する問題点も指摘されている。問題点の 1 つとして, 未知の状態への到達のみが直接的にゴール到達に寄与しない点が挙げられている. 特に, フロア探索においては, 例えば図 1 のような状況を考えると, 初めゴールと別方向に向かった場合, ゴールに到達するには一度通った道に引き返す必要がある. しかし, 内発的報酬を基に考えた場合, 一度通った道にひき返す行為は新規性のない状態へ移動する行為であり, 内発的報酬は小さくなってしまう. 実際, 上述の研究でもクリアに失敗したエピソードにおいてこのような場合が見受けられたと述べられている.

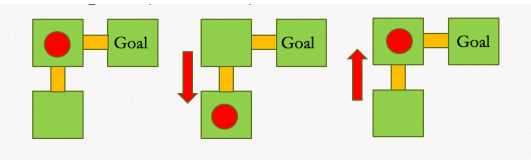

図 1 道を戻る必要があるフロア形状のケース

3.2 一時的好奇心

3.1 節で述べた RND の問題点は, Hu ら [3] によっても指摘されている. RND における好奇心を persisting curiosity (PC) とし, PC のみを利用した探索においては探索済み状態を保持し続けることにより過剰に新規性に執着するように学習が行われてしまい, 一貫性のない戦略をとるようになるといった問題が生じるとしている. 同論文では, PC に

代わる手法として, 内発的報酬の計算に使用する観測列を一定範囲に制限した temporary curiosity (TC) を導入した PCTC を提案し, Atari の環境での実験において, 好奇心の範囲が単に新規性の探索だけでなく, 目標達成に対して重要な特徴量に注目するよう拡大されるなど一定の成果を示した.

4. 提案手法

本研究では, 先行研究で述べられている問題点を基に, RND に加えて PCTC に基づく一時的好奇心 (TC) を導入した内発的報酬設計をローグライクに適用することを提案する. 一時的好奇心の導入により, ローグライク環境における RND において問題であった, 新規性に執着することにより探索に失敗するケースが軽減され, より効率的に学習が促進されることが期待されると共に、PCTC の有効性が検証できる.

4.1 内発的報酬設計

学習手法としては PPO を用い, 報酬設計に RND + 一時的好奇心 によるものを用いる. PPO, RND の学習は一時的好奇心とは独立に行われるため, 以下では一時的好奇心に基づく報酬生成について述べる。

RND の問題点は, 探索済みの状態に関する情報が全て反映されるために, 探索済みの状態を過剰に避けることであった. そこで, 一時的好奇心では, 探索済みの状態でもある程度の探索を行うため, 観測した情報の一部のみを保持し, それに基づく報酬を生成することでこの問題に対応した. 具体的には, 現在時刻 t から連続する一定の長さの観測列 $H_t = o_{t-l} : o_t$ を保持し, この観測の変化量によって報酬を生成する. すなわち, 観測 o_t に対応する状態表現 g_t を用いて,

$$i_t^T = \sum_{n=1}^{l} w_n ||g_{t-n+1} - g_{t-n}||^2 \tag{4}$$

として表される. ここで, w_n は重みを表す係数であり, 直近の状態変化を重視するため $w_n \geq w_{n+1}$ を満たす. 結果的に, 全体の報酬 R_{all} は以下のように表される.

$$r_t^{all} = c_e \times r_t^e + c_i \times E(r_t^i, i_t^T) \tag{5}$$

ここで, E は RND による報酬と一時的好奇心による報酬の比率を決定する関数である.

4.2 使用環境

ローグライクを対象とした先行研究においては, 使用されている環境は研究ごと多様であり, Atari 2600 のような統一的に利用されている実験環境が存在しない. こうした背景から実験環境を独自に作成・提案しているものも多いが, 本研究ではカスタマイズ性や, 環境の再現の容易さから MiniHack [12] と呼ばれる環境を用いる. MiniHack は多様なローグライク環境を提供するが, 今回特に用いたものは以下のようなものである.

- フロア：(21, 79) の大きさであり, 定数個（変更可能）の部屋及びそれらを結ぶ通路で構成される. ゴールは部屋のいずれかに配置される.
- 状態空間：フロアサイズと同じ (21, 79) の 2 次元配列。（各要素はフロアの各点のオブジェクトの種類を表す整数。）入力の際には、one-hot 表現に変換を行う。
- 行動空間：離散、8 方向への移動.
- 報酬：ゴールに到達したとき +1.0. 1 回の移動, 及び無意味な行動（移動できない場所へ移動しようとする）をしたときにも報酬を設定することができる.
- エピソード長：ゴール到達で終了, 上限 500 step で終了. step 数は変更可能.

フロアはエピソード毎にランダム生成され, 図 2 のようなものが生成される. 実際には, エージェントが観測できるのは自身の周囲及び探索済みの範囲のみであるため, 生成されたマップを最初から把握することは出来ず, 従って観測される状態も探索によって変化していく. また, 先行研究同様, 敵・アイテムなどのオブジェクトは配置せず, 純粋にゴールを移動により探索するタスクとした.

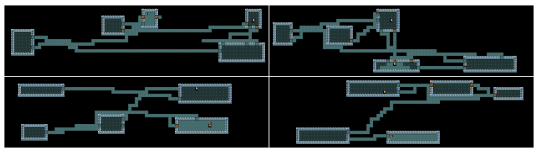

図 2 生成されるフロア形状の例

5. 実験

5.1 実験環境

提案手法の検証では, 環境を表 1 のように設定した.

表 1 環境の設定

変数	値
部屋数	2
状態空間	(4, 21, 79)
行動空間	8 （離散）
エピソード長上限	500
ゴール時報酬	1.00
移動時報酬	-0.001
無意味行動時報酬	-0.001

状態の入力は, プレイヤーの位置, 壁（通過不能場所）の位置, 床（通過可能場所）の位置、階段（ゴール）の位置の 4 つの有無を 0, 1 で表した.

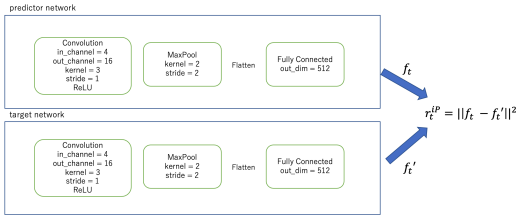

図 3 RND ネットワーク構造

表 2 環境の設定

変数	
観測列の長さ	128
係数 w_n	$w_n = (0.99)^n$

5.2 モデル

実験では, (1) PPO, (2) PPO + RND, (3) 提案手法の 3 つのモデルを用いて学習を行った. どれも学習手法としては PPO を用い, 内発的報酬設計のみ異なる。(1) では内発的報酬は与えず, (2) では RND による内発的報酬を, (3) では提案手法による内発的報酬を与える. (1) 〜 (3) それぞれに対応した内発的報酬を与える環境のラッパーを作成し, それぞれの環境で PPO を用いて学習させることで比較を行った. PPO の実装としては, stable baselines3 [13] のものを用いた. RND については, (2), (3) 共通のネットワークを用いた. ネットワーク構造を図 3 に示す. 一時的好奇心については, 観測の状態表現として観測を CNN に入力し 512 次元のベクトルに変換したものを用い, その他パラメータについては表 2 のように設定した. 最終的な内発的報酬としては, 以下を用いる.

$$r_t^{all} = c_e \times r_t^e + c_{i_P} \times r_t^{iP} + c_{i_T} \times t_t^{iT} \tag{6}$$

ここで, $r_t^e, r_t^{iP}, r_t^{iT}$ はそれぞれ外部報酬, RND による内発的報酬, TC による内発的報酬である. 今回の実験では, $c_e = 1.0, c_{iP} = 1.0, c_{iT} = 0.10$ とした.

5.3 結果

実験の評価指標として, 直近 10000 エピソードにおけるゴール到達回数, 及びゴールに到達した早さとしてエピソード長を計測した. また, 総ステップ数は 6,500,000 とした. 結果を図 4, 図 5 に示す.

エピソード長は, ゴールした時点で 1 エピソードが終了するため, 短いほど早くゴールへ到達できていることに注意する. 実験結果から, PPO のみを用いた場合と比較して, 内発的報酬を用いた手法ではどちらもゴール率が大きく向上していることが分かる. また, 内発的報酬として RND のみを用いた場合と比較して, 一時的好奇心を用いた手法ではゴール率がさらに向上していることが確認できた.

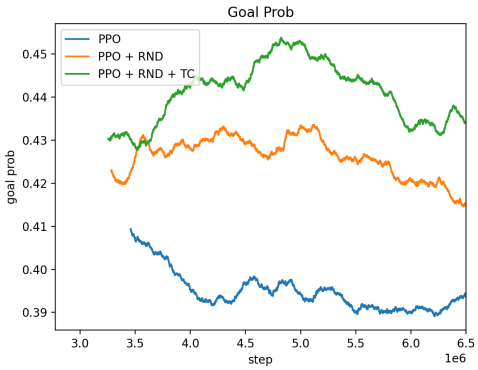

図 4 ゴール率

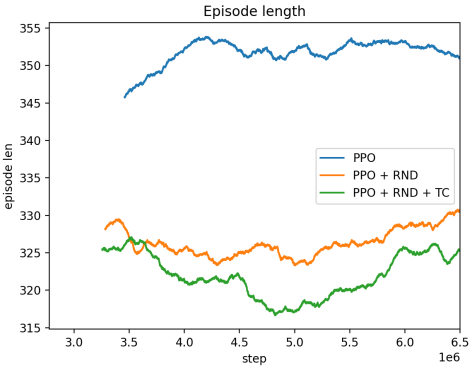

図 5 エピソード長

6. 結論

6.1 まとめ

本稿では、ローグライク環境での強化学習について, RND による内発的報酬が新規性に執着してしまう問題への対処として, 一時的好奇心に基づく内発報酬設計を導入することを提案した. 実験により, ローグライク環境への内発的報酬の導入の効果, 及び一時的好奇心の導入によって学習が促進できていることが確認できた.

6.2 課題

提案手法が学習に効果があることは検証できたが, 現時点では RND をローグライク環境に適用した場合の問題点を本質的に解消できていることを検証できた訳ではなく, 今後はこの点の検証が必要である. 今回の実験設定では学習難度・速度を考慮し部屋数を 2 として実験を行ったが, 部屋数 2 の場合の部屋構造は 2 部屋を 1 つの通路が結ぶ構造である. 3.1 節で指摘したようなゴールに必要となる往復移動は, 例えば部屋内で通路から遠ざかったために戻る必要が生じる, といったケースは考えられるものの, 部屋間の移動という単位で見れば生じない. よって, より問題の検証を行う上では, 部屋数をより増加させての検証や, 特定の地形を用いた検証も必要であると考えられる. また, 一

時的好奇心の生成の際に用いた観測の状態表現についても, 検討が必要である. 今回は単に CNN へ入力したものを用いたが, よりフロアの特徴量を適切に表現できる状態表現を考慮すべきである.

参考文献

[1] 加納由希夫, 好奇心に基づく内部報酬を用いた強化学習によるローグライクゲームの 学習, 修士論文, 東京大学大学院, 2020.

[2] Andrea Asperti and Daniele Cortesi. Reinforcement learning in rogue. 修士論文, University Of Bologna, 2018.

[3] Hangkai Hu, Shiji Song, and Gao Huang. Self-attention-based temporary curiosity in reinforcement learning exploration. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019.

[4] 高橋一幸, Sila Temsiririrkkul, 池田心. ローグライクゲームの研究用ルール提案とモンテカルロ法の適用. ゲームプログラミングワークショップ 2017 論文集, 第 2017 巻, pp. 19-25, nov 2017.

[5] 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.

[6] Volodymyr Mnih, Adri‘a Puigdom‘enech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lill- icrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. CoRR, Vol. abs/1602.01783, , 2016.

[7] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. 2018.

[8] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven explo- ration by self-supervised prediction. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, pp. 2778 – 2787, 2017.

[9] Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In Advances in Neural Infor- mation Processing Systems, pages 4697 – 4707, 2017.

[10] 金川裕司, 金子知適, ローグライクゲームによる強化学習ベンチマーク環境 RogueGym の提案, Proceedings of The 23rd Game Programming Workshop, pp. 120-127, 2018.

[11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. CoRR, Vol. abs/1707.06347, , 2017.

[12] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich K ¨ uttler, Edward Grefenstette, and Tim Rockt ¨ aschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. CoRR, abs/2109.13202, 2021.

[13] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. Journal of Machine Learning Research, 22(268), 1 – 8.

