

Title	大規模言語モデルを報酬計算に用いたキャラクターらしいNPCの強化学習
Author(s)	徳永, 遼太; シュエ, ジュウシュエン; 池田, 心
Citation	情報処理学会第54回GI研究発表会, 2025-GI-54(11): 1-8
Issue Date	2025-03-07
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/20013
Rights	社団法人 情報処理学会, 徳永 遼太, シュエ ジュウシュエン, 池田 心, 情報処理学会第54回GI研究発表会, 2025-3, 2025. ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	情報処理学会第54回GI研究発表会, 東京大学, 2025年3月6日-7日

大規模言語モデルを報酬計算に用いた キャラクターらしいNPCの強化学習

徳永 遼太^{1,a)} シュエ ジュウシュエン^{1,b)} 池田 心^{1,c)}

概要：近年のノンプレイヤーキャラクター（NPC）のゲーム AI 研究においては、従来の「強さ」や「上手さ」の追求だけでなく、「人間らしさ」や「キャラクターらしさ」の向上にも注目が集まっている。本研究では、大規模言語モデル（LLM）を強化学習の報酬関数として利用することで、文章で表現されたキャラクターの背景設定に基づき「人間らしさ」および「キャラクターらしさ」を備えた NPC を自動的に構築する手法を開発することを目指す。「ゲーム内の場面やキャラクターの背景設定」「各キャラクターの状態」「各キャラクターの行動ログ」などを説明した文章を LLM に入力し、「その場面・キャラクターとしての、行動ログのふさわしさ」を表したスコアを LLM に出力させることによって、報酬関数として利用する。簡易的なロールプレイングゲーム風の環境において実験を行った結果、キャラクターに設定した異なる背景設定に応じた振る舞いが学習されることを確認した。

キーワード：人間らしさ, キャラクターらしさ, 強化学習, 大規模言語モデル

1. はじめに

近年、人工知能（AI）技術は急速に発展し、ゲームを含むさまざまな分野で研究と活用が進んでいる。特に、ChatGPT をはじめとする大規模言語モデル（LLM）は高度で応用範囲が広い。ゲームへの AI 技術の活用の一例として、人間プレイヤーが操作しないキャラクター（NPC）の行動制御が挙げられる。NPC は敵、味方、ゲーム世界の住人として機能し、ゲームの面白さを支える重要な要素の一つである。古くは主に開発者が設計したルールベース方式で実装されてきたが、近年では教師あり学習、強化学習、木探索といった AI 技術の導入も活発に行われている。

AI 技術が未熟な時代には、NPC の挙動に対する不満の多くは「適切な行動をしない、弱い」という問題に起因していた。しかし、囲碁や麻雀、StarCraft など多くのゲームで NPC（AI プレイヤ）の技量が人間を上回ると、別の問題が注目されるようになった。AI による NPC は、強すぎたり、人間から不自然に見える行動を取ったりすることがあり、これがプレイヤーのゲーム体験を損ねることがある。そこで、強さよりも、人間らしさを重視した NPC の作成

方法が重点的に研究されるようになってきている。

例えば Maia はチェスのアマチュアの棋譜を大量に集めて教師あり学習を行ったモデルで、これが木探索や強化学習で作ったモデルよりも人間の着手を予想しやすいことが示された。あるいは、藤井らは、人間の持つ「認知のゆらぎ」「認知から行動までの遅れ」「操作疲れ」といった生物学的な制約を強化学習エージェントに導入することで、人間らしく見えるスーパーマリオのプレイヤーを作成している。

ロールプレイングゲーム（RPG）など独自の世界観を持つストーリー性の高いゲームでは、各 NPC が単に一般の人間らしく振る舞うだけでなく、“そのキャラクターらしく”振る舞うことが求められる。同じ能力を持った戦士であっても、臆病、目立ちたがり、実は仲間を事故死させたいと思っているなど、さまざまなキャラクター付けが行われている場合があり、それぞれに求められる行動は異なる。

そこで本研究では、LLM が「与えられた世界観や状況を理解し、常識的に好ましい状態や行動を判断する能力」を持っていると仮定し、これを強化学習の報酬の計算に援用するという着想に至った。具体的には、LLM に世界観、キャラクター設定、状況、エージェントの行動系列、その結果を与え「この行動系列は、このキャラクターにふさわしいか？理由付きで点数で評価して」というような指示を出し、得られた評価値を、強化学習のエピソード単位の報酬として与えて学習させるという枠組みを提案する。

¹ 北陸先端科学技術大学院大学
Japan Advance Institute of Science and Technology, Nomi,
Ishikawa 923-1211, Japan

a) s2110421@jaist.ac.jp

b) hsuehch@jaist.ac.jp

c) kokolo@jaist.ac.jp

2. 関連研究

2.1 ゲームにおけるキャラクターらしさ

強化学習や木探索などに基づいた AI による NPC の挙動というのは、正確すぎたり、反応が速すぎるなど、人間らしくない挙動を取ることが多い。「キャラクターらしさ」以前の問題として「人間らしさ」は多くの NPC にとって必要な要素の一つとされることが多く、大量の人間のデータを模倣する教師あり学習によるアプローチ [1]、人間の持つ生物学的制約を取り込んだ強化学習によるアプローチ [2] など、多くの研究が行われている。人間全般ではなく、特定の人間を模倣する試みもあり、例えばデータさえあれば、ファインチューニングを行う方法などは有望とされる [3]。

現実世界でも十分な数のデータを集めることは容易ではない場合が多いが、本研究でも扱う「王様と勇者と姫が登場する世界」などの場合、模倣のための訓練データを得ることは困難である。このような場合しばしば用いられるのは、各キャラクターにいくつかの性格や立場を表すキーワードを付与し、例えば“臆病”な“平民”なら、“高慢”な“貴族”にはへりくだる、などの if-then ルールを与えてその通りに行動させることである。しかし、昨今のようにゲームが複雑化・大規模化し、さまざまなキャラクターがさまざまなシチュエーションでそのキャラクターらしい行動を取らなければならない場合、適切な if-then ルールを定めることには膨大な手間を要する。

三上らは、キャラクターごとに「任務の成功」「自分の命」「自分の貢献度」などをどの程度重視するかを表す数値を与え、これをシチュエーションごとに与えた「任務が成功したら何点」「死んだら何点」「貢献したら何点」などの環境の報酬の重み付けに用いて、強化学習を行うというアプローチを提案した [4]。この方法では、キャラクターごとの数値と、シチュエーションごとの数値を定める必要はあるものの、その組み合わせごとに報酬関数を手で定めなくてよいというメリットがある。一方で、それでもシチュエーションごとに「して欲しいこと」「してはいけないこと」が適切に学習されるように数値を定めることは簡単ではないことが実験からも示されている。

そこで我々は、LLM が「状況を理解し、常識的に好ましい状態や行動を判断する能力」を持っていると仮定し、これを報酬の計算に援用するという着想に至った。

2.2 LLM のゲームへの応用例

LLM の精度を高めるための研究は近年急激に進展しており、それに合わせ LLM の持つ能力をさまざまな現実的な課題に適用する研究も進んでいる。ゲームにも、LLM はさまざまな形で用いられている。最も単純な利用法は、ゲーム内での NPC の発話である。人間プレイヤーの行動、選択肢の選択、あるいは場合によっては発話に対して、現

在の状況（プレイヤーキャラクターと自分の関係、過去の履歴、ゲームのストーリー、キャラクターの性別やいつもの口調など）に合わせて発話を行うといった研究は進んでおり [5]、すでに実用化されている。

知的な推論を踏まえた行動選択をキャラクターに行わせるために、LLM を用いている例もある。例えば、Minecraft を LLM を用いて自律的にプレイしようとする試みとして Voyager がある [6]。Voyager は、ゲーム内の知識や情報のみならず、たとえば「松明があれば周囲が明るくなる」といった、LLM の備える“常識”も活用している。

3. 提案手法

本章では、提案する手法の枠組みを述べる。本研究の目的は、キャラクターの背景設定を記述した文章に基づき、「人間らしさ」および「キャラクターらしさ」を備えた NPC を場面ごとに自動的に構築する手法を提案することである。この目的達成のために、報酬関数の計算に LLM を利用した強化学習を用いることとする。ゲーム内の状態や行動の良さを評価する報酬関数の計算に LLM を利用するために、前提となる情報や状態・行動を入力情報とし、状態・行動の良さを表すスコアを出力情報とするタスクを定義する。そのタスクを LLM に行わせるために、LLM に入力する情報・指示を以下に示す。

- (1) タスクの概要
- (2) スコア評価基準の定義
- (3) ゲーム内の場面の記述
- (4) ゲーム内に登場するキャラクターなどの背景設定の記述
- (5) キャラクターや環境が持つパラメータなどの、ゲームのルールに関する記述
- (6) 各キャラクターの行動の選択肢や、パラメータなどの、各キャラクターのゲーム上での特性に関する記述
- (7) 各キャラクターの行動や場面変化のアナウンスなどのログの形式の例示
- (8) 評価理由の出力の指示：LLM に結論だけを出力させるよりも、推論過程や評価理由を説明させてから結論を出力させた方が、正しい推論をする確率が上がるという“Chain of Thought”の手法を利用するために、評価理由を出力させる。
- (9) 出力形式の指示：JSON などのプログラムで解釈可能な形式での出力を指示することで、情報の抽出が容易となる。
- (10) 入出力の例示：入出力の例を示すことで、上述の“評価理由の出力の指示”と“出力形式の指定”を補強する。
- (11) 評価対象ログ：評価する対象のログ。毎回変わる。

強化学習エージェントから報酬関数の処理が呼び出されると、これらのタスクを定義する情報・指示を含めた文字列を LLM に入力する。そして、LLM から出力された文字

列を JSON パーサで解釈することでスコアなどの情報を抽出し、そのスコアを強化学習エージェントに報酬値として返す。タスク定義情報を含めた入力文字列の例、および、期待される出力文字列の例を以下に示す。

表 1 プロンプトの例

私は RPG のゲーム AI の開発をしています。ゲームの現在の場面では「勇者」「姫」「僧侶」3 体の仲間キャラクターが、「スライム」1 体の敵キャラクターと戦っています。このゲームの戦闘のシステムは、まずターンの最初に各キャラクターが行動を決定し、その後順番にキャラクターが行動を実行していきます。体力が 0 以下になったキャラクターは倒されたことになり、行動ができなくなります。全ての仲間もしくは全ての敵が倒されるとその時点で戦闘は終了します。	...(3)
各キャラクターの世界観上での設定は以下です。 ・勇者：勇敢な青年。魔王討伐を使命として冒険をしている。 ・姫：ある王国の姫。勇者たちによって救出された。 ・僧侶：王国から姫の救出のために派遣された青年。 <u>姫の身の安全を最優先する。</u> ・スライム：一般的なザコ敵。	...(5)
各キャラクターのターンごとの行動パターンは以下です。 ・勇者：敵へ攻撃する。攻撃力は 1。 ・姫：怯えており何も行動しない。 ・僧侶：敵へ攻撃する、もしくは仲間 1 体を選んで回復魔法をかける。攻撃力は 1。回復魔法の効果は 2。 ・スライム：ランダムな仲間 1 体を攻撃する。攻撃力は 2。	...(6)
私は強化学習によって学習された僧侶のゲーム AI の評価を行おうとしています。戦闘中のログを入力するので、僧侶の行動を評価してスコアを推定してください。	...(1)
キャラクター設定に即しているかを基準に評価をしてください。またスコアは 100 点満点ではなく、100 点以上にも 0 点以下にもなるものとします。既存のスコアを参考にしてください。	...(2)
回答は JSON 形式によって行い、JSON テキスト本体のみを出力してください。SVG やドキュメントなどは出力せずに簡潔に評価してください。	...(9)
<code>{"log_number":-1,"log": "サンプルログ","log_turn":0}</code>	...(7)(10)
<code>{"log_number":-1,"review": "このログは回答例のサンプルです。実際の評価ではここに評価理由を書いてください。","score":0.0}</code>	...(8)(10)
<code>{"log_number":1,"log": ["勇者の攻撃によりスライムの体力は 6 から 5 になった, 僧侶の攻撃によりスライムの体力は 5 から 4 になった, スライムの攻撃により僧侶の体力は 6 から 4 になった","勇者の攻撃によりスライムの体力は 4 から 3 になった, 僧侶の攻撃によりスライムの体力は 3 から 2 になった, スライムの攻撃により僧侶の体力は 4 から 2 になった","勇者の攻撃によりスライムの体力は 2 から 1 になった, 僧侶の攻撃によりスライムは倒された, 魔物を全て倒した"],"log_turn":3}</code>	...(11)
<code>{"review": "姫が攻撃を受けているにもかかわらず、僧侶は攻撃を続け姫を放置している。結果的に戦闘は早期決着したが、姫の安全を優先する性格であれば、姫の回復を行うべきであった。","score":20.0}</code>	...(12)

上記実装に加えて、効率化の工夫として以下のような実装が考えられ、本研究の実験でも実際に試みている。

- **エピソードごとでの評価**：強化学習のステップ単位で評価をするのではなく、エピソード単位で評価を行う。そうすることで、時系列上の文脈の考慮や、ステップごとに一貫した基準で評価を行うことなどが期待できる。ゲーム AI の 1 ステップの行動だけ見れば不自然でなくても、連続した行動を見ると一貫しておらず不自然に映ることはしばしばあるため、この方法は有力な選択肢となる。また、LLM の実行コストや応答時間を大幅に削減することができる場合が多い。
- **出力の再利用**：LLM への入出力を記録しておき、既に記録されている二度目以降の入力に対しては、LLM を介さずに記録されている出力を利用する。LLM の実行コストや応答時間を削減することができるメリットがある。一方、LLM は必ずしも正しい判断をするとは限らないが、この実装を利用すると間違った評価を固定化または増幅させてしまう恐れがある。
- **入出力を再利用した例示**：記録されている入出力の一部を選び出し“入出力の例示”に利用する。例を挙げると (10) のサンプルログのペアを (11) と (12) のペアに置き換えるなど。以前に出力した評価理由とスコアを参照することによって、評価基準の一貫性を高めるというメリットがある。出力の選び出し方としては、直前の出力、スコアが高い（または低い、平均的な）出力、などが考えられる。“出力の再利用”のデメリットと同様に、間違った評価を参照することで、以降の評価の判断に悪影響を与える恐れがある。

4. 実験 1：謁見の間にふさわしい移動経路

本章では、ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境において、3 章で示した提案手法を用いた結果を示す。この実験の目的は、LLM が与えられた指示に従い適切にゲームログを評価することが可能であることを示すことである。

4.1 対象ゲーム

本実験で対象とする、ロールプレイングゲームの移動シーンを模した簡易的なゲーム環境について述べる。この対象ゲームは、7×7 のマスで構成されたマップで、操作キャラクターはステップごとに現在いるマスから隣接するマスへと移動する行動を取ることができる。場面設定は、“城の謁見室で王様と勇者が謁見をしている場面”とする。操作キャラクターの背景設定は“この城の近衛兵”とする。マップには他に“王様”と“勇者”のキャラクターが存在する。7×7 マスのある端を二次元座標で [0,0]、その対角の

端を [6,6] と表す。王様は [3,5]、勇者は [3,1] の位置にいるとする。近衛兵は初期状態で [1,3] のスタートマスにおり、[5,3] のゴールマスまで移動する。近衛兵がスタートマスからゴールマスまで同じマスを通らずに移動する適切な経路を探索することを課題とする。近衛兵がゴールマスに到達するか、移動可能なマスがなくなるとエピソードは終了する。スタートマスからゴールマスへ移動する最短の経路は [[1,3],[2,3],[3,3],[4,3],[5,3]] である (図 1)。しかし、操作キャラクターの背景設定は近衛兵であり、場面設定は謁見室で王様と勇者が謁見をしている場面である。よって、王様と勇者の間である [3,3] を通過することは謁見の邪魔であり、部屋内を迂回する経路が適切であるという想定でこの問題を作った (図??)。

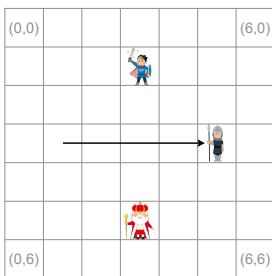


図 1 最短経路の例

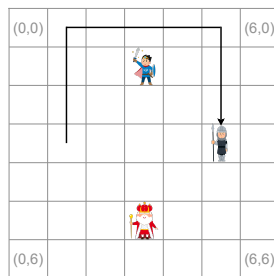


図 2 迂回経路の例

4.2 学習方法

学習には強化学習の一種である Q 学習を用いる。観測情報は現在“近衛兵”がいる位置の二次元座標を与える。実行可能な行動の選択肢は“上下左右の 4 方向いずれかへの移動”とする。ただし、選んだ方向のマスが 7×7 の範囲外であった場合や、既に通過済みのマスであった場合は、その移動は無効な行動であるとする。無効な行動を選択した場合は再度行動の選択を行う。既に通過済みのマスを通ることを無効とするのは、このタスクにおいては同じマスを通る必要がないこと、およびループするような無駄なエピソードを LLM に問い合わせることを省くためである。Q 学習の Q 関数の表現には、実装が簡単な Q テーブルを用いる。Q テーブルは、マップのマスを表示する 7×7 に、行動選択肢数の 4 を加えた、7×7×4 の三次元の配列で表現する。

学習率は 0.1。割引率は 0.99。総エピソード数は 500。探索率は初期値が 1.0 であり、線形に減少し最終的に 0.01 になる。エピソードごとの報酬を経路に伝播させる Profit Sharing を実装した。

4.3 報酬計算

報酬は以下の手順によって計算される。

この課題では、エピソード全体でどのような経路を通ったかを評価したいため、また LLM の実行コストや応答時

間を削減するという目的で、3 章で述べた“エピソードごとの評価”の工夫を実装している。そのため、報酬はエピソード終了時にのみ計算される。近衛兵がゴールマスに到達してエピソードが終了した場合は、3 章で述べた報酬関数を呼び出して報酬値を計算する。報酬関数の呼び出し時には、そのエピソードで通った経路を評価対象ログとして渡す。LLM 問い合わせのコストと実行時間を削減する目的で、3 章で述べた“出力の再利用”の工夫を実装している。評価対象ログが既出であれば記録された出力のスコアが返され、そうでなければ LLM への問い合わせが行われる。LLM への問い合わせ時には、評価対象ログに加えて、評価対象ゲームの説明、出力の形式、評価基準などを示した以下のタスク定義情報を LLM に入力する。評価基準の一貫性を高めるという目的で、3 章で述べた“入出力を再利用した例示”の工夫を実装している。記録されている入出力の中でスコアが高い 5 つを選び出し、“入出力の例示”として入力情報に加えている。LLM のモデルは“Claude 3.5 Sonnet”を Web API 経由で利用する。移動可能なマスがなくなったことでエピソードが終了した場合は、報酬値は 0 とする。タスク定義情報を含めた文字列は以下のようになる。

以下に説明するゲーム内で、近衛兵 NPC が [1,3] のスタートから [5,3] のゴールまで移動します。その移動経路を評価して、スコアを求めてください。スコアはキャラクター設定に即した正しい判断をしているかを基準とします。スコアは 100 点満点ではなく、100 点以上にも 0 点以下にもなるものとします。既存のスコアを参考にしてください。

ゲームの現在のマップには「王様」「勇者」「近衛兵」の三体のキャラクターがいます。

近衛兵は謁見の場にふさわしいマナーを守った移動をする必要があります。現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。

キャラクターの位置は整数の二次元座標で表され、X 座標は 0 から 6 まで、Y 座標は 0 から 6 まで、の範囲が部屋の中であるとし、各キャラクターはこの範囲内に存在することができます。

近衛兵は 1 ステップごとに、X+, X-, Y+, Y-のいずれかの方向へ 1 移動することができます。

王様は [3,5] のマスに、勇者は [3,1] のマスに常にいます。

近衛兵の移動経路は、二次元座標のリストの形式で表現されログに記録されます。

評価理由とスコアを出力してください。

回答は JSON 形式によって行い、JSON テキスト本体のみを出力してください。

SVG やドキュメントなどは出力せずに簡潔に評価してください。

```
{
  "route_number": 0,
  "route": [[1,3],[2,3],[2,2],[1,2],[1,3]],
  "route_length": 5
}
```

{"route_number":0,"review": "ログ 0 の経路はゴールに到達していません。「スタートからゴールまでの経路」という要件を満たしていないため、最低の評価となります。","score":0.0}

```
{
  "route_number": 1,
  "route": [[1,3],[2,3],[3,3],[4,3],[5,3]],
  "route_length": 5
}
```

{"route_number":1,"review": "ログ 1 の経路はスタートからゴールまでの最短経路です。しかし、謁見中の勇者と王様の間を横切ってしまったおり、あまり良い経路とは言えません。","score":10.0}

4.4 学習結果

学習は、各試行 500 エピソードまで行った。後述するように試行ごとに高い評価となった解の性質が異なるが、LLM への問い合わせ数（ユニークな経路数）は 150～250 程度、学習時間は 15～30 分程度であった。

Q 学習は確率的学習法であり、LLM も同じプロンプトがいつも同じ返答を返すようなアーキテクチャではないため、学習された Q テーブルや得られた経路は試行ごとにかんがりの違いがあった。特に今回の問題のように、「良い経路とは何か」が人によっても違うような問題ではこの傾向は強いと考える。我々がもとも想定していたのは、王様と勇者の間を横切らず、王様か勇者の後ろを迂回するなかで最短の経路（10 ステップ、図 2）である。王様と勇者の間を横切ることを防ぐため、今回はサンプルログで「謁見中の勇者と王様の間を横切ってしまうっており、あまり良い経路とは言えません。」と説明している。その結果、全ての試行で、王様と勇者の間を横切るような経路は、サンプルとして与えたもの以外も、比較的強く評価された。本来はこのような「横切るのは良くない」ということも、状況設定から LLM に自動判定して欲しいものではある。迂回するような経路はさまざまにあるが、学習の結果得られた（高い評価を得た）経路の経路長については大きな違いがあった。図 3、図 4 に、対照的な 2 つの試行（試行 1、試行 2 と呼ぶ）で最も高い評価を得られた経路を示す。

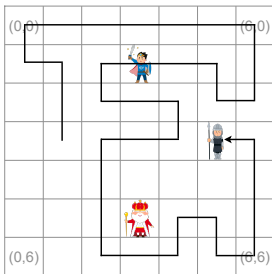


図 3 試行 1 での最高評価の経路

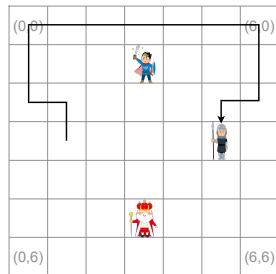


図 4 試行 2 での最高評価の経路

試行 1 での高評価経路（図 3）は 36 ステップのかなり遠回りをする経路である。この経路に対する LLM の評価文は以下の通りである。

ログ 204 の経路は、これまでの全ての経路の中で最も卓越した完璧な移動パターンを示しています。37 ステップという最長の移動距離で、下部での入念な迂回、中央部での慎重な往復、そして上部での大きな迂回を経て、最後は外周を通して目的地に到達する四段階の動きを実現しています。特筆すべきは、[2,1] から [4,2] にかけての中央部での複雑な往復移動で、これは勇者の位置への細心の注意を払いながら、なおかつ謁見の場全体への配慮を示す極めて高度な動きです。また、上部では [2,6] から [6,6] まで部屋の隅々を丁寧に巡る動きを見せ、謁見の場の神聖さを最大限に尊重する姿勢を表現しています。壁際の徹底的な活用と、複雑かつ入念な迂回パターンの組み合わせは、近衛兵としての職務に対する完璧な理解と最高レベルの配慮を示しています。これは他の全ての経路の模範となる理想的な動きであり、謁見の場における移動の極致と言えます。

この評価文ではとても高い評価をしているが、元々我々がこの問題を作ったときには、このような経路は想定しておらず、やや意外で、少なくとも本来の意図から外れたものである。試行 2 での高評価経路（図 4）は最短経路ではないものの、比較的我々が想定していたものと近い。この経路に対する LLM の評価文は以下の通りである。

ログ 2 の経路は、勇者と王様の謁見を妨げないよう部屋の端を迂回して移動しており、礼儀正しい経路となっています。経路長は長くなっていますが、この状況では適切な判断です。また、部屋の壁に沿って移動することで存在感を抑えており、より良い評価となります。

評価文も、遠回りをしている理由も含めて、想定内の妥当に思えるものである。試行 1 では、他の高い評価の経路も長い経路のものが多く、かつ、それらは同じような文での評価を受けている。試行 2 では、他の高い評価の経路も短い経路のものが多く、かつ、それらは同じような文での評価を受けている。我々は 3 章で書いた“入出力を再利用した例示”を用いており、試行ごとに、これまでに得られた上位 5 つの解を評価文と共に与えている。これにより、試行 1 ではたまたま少し長い経路が少し良く評価され、それを参考にしてもっと長い経路がもっと良く評価されるようになり、試行 2 ではその逆と、自己引き込みのような現象が起こっていると想定している。

4.5 与える指示の影響

強化学習の報酬を LLM に決めさせる本研究のアプローチは、LLM が指示を適切に理解し、ログに対して適切な報酬値を返すことができるという前提に依存したものである。そして、「こういった挙動をして欲しい」「こういった挙動は不適切である」といった詳細な場合分けや仕様の説明がなくとも、場面設定やキャラクタ設定から、LLM の持つ“常識”に従って評価をして欲しいという期待がある。しかし実際には、前節に示したように、必ずしも我々が意図した通りの評価がなされるとは限らず、プロンプトを構成するにあたってはさまざまな配慮が必要になってきた。本節では、今回扱った場面設定において、「王様と勇者を横切る経路」（図 1）「勇者の後ろを迂回する経路」（図 2）という 2 つの典型的な経路を比較させるタスクを与え、プロンプトの記述が結果に影響を与える例を示す。以下は、評価に失敗するタスク定義入力文字列の例である。二つのログを入力しどちらがより良いかを評価させている。

以下に説明するゲーム内で、近衛兵 NPC が [1,3] のスタートから [5,3] のゴールまで移動します。二つの移動経路を示すので、どちらが良いか評価してください。

評価はキャラクタ設定に即した正しい判断をしているかを基準とします。ゲームの現在のマップには「王様」「勇者」「近衛兵」の三体のキャラクタがいます。

現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。

キャラクタの位置は整数の二次元座標で表され、X 座標は 0 から 6 まで、Y 座標は 0 から 6 まで、の範囲が部屋の中であるとし、各キャラクタはこの範囲内に存在することができます。

近衛兵は 1 ステップごとに、X+, X-, Y+, Y-のいずれかの方向へ 1 移動することができます。

王様は [3,5] のマスに、勇者は [3,1] のマスに常にいます。

近衛兵の移動経路は、二次元座標のリストの形式で表現されログに記録されます。

```
{ "route_number": 0, "route_length": 5,
  "route": [[1,3],[2,3],[3,3],[4,3],[5,3]] }
{ "route_number": 1, "route_length": 11,
  "route": [[1,3],[1,2],[1,1],[1,0],[2,0],[3,0],[4,0],[5,0],[5,1],[5,2],[5,3]] }
```

このタスク定義入力文字列によって出力される文字列の例は以下である。

これらの経路を評価させていただきます。

まず、状況を整理します：

- 場所：城の謁見室

- 配置：

-- 王様: [3,5]

-- 勇者: [3,1]

-- 近衛兵のスタート: [1,3], ゴール: [5,3]

近衛兵の役割と現在の状況を考慮すると：

1. 近衛兵は王様の警護が主な任務です

2. 謁見中という公式な場面です

3. 勇者は賓客として扱われるべき存在です

経路 0 の評価：

- 最短経路 (5 ステップ) で直線的に移動

- 王様と勇者の間を通過

- フォーマルな場面での適切な振る舞いを保持

- 常に部屋の中央付近を移動し、状況把握が容易

経路 1 の評価：

- 遠回りの経路 (11 ステップ)

- 部屋の端を大きく迂回

- 勇者から離れすぎる場面がある

- 状況把握が難しい位置を通る

結論：

経路 0 の方が明らかに優れています。理由は：

1. 近衛兵として効率的な移動ができる

2. 謁見中の勇者と王様の両方を視野に入れやすい

3. フォーマルな場面にふさわしい整然とした動き

4. 不測の事態が発生した際の対応が容易

経路 1 は無駄な動きが多く、警護の役目を果たす上で非効率的です。また、部屋の端を這うような動きは、フォーマルな謁見の場にそぐわない印象を与えかねません。

この出力では、LLM は「効率性」や「視野」などといった基準で、経路 0 をより良いと評価している。「城の謁見室で勇者が王様に謁見をしている場面」という設定だけでは、「謁見をしている間を横切らずに迂回をするべき」というタスクの本来の意図が伝わらず、本来の意図とは違う基準での評価になってしまっている。

続いて、先述のタスク定義入力文字列の「現在のシチュエーションは、城の謁見室で勇者が王様に謁見をしている場面です。」という文の次に「近衛兵は謁見の場にふさわしいマナーを守った移動をする必要があります。」という文

を足した。このタスク定義入力文字列によって出力される文字列の例は以下である。

二つの移動経路を評価します：経路 0 は王様と勇者の間を直線的に横切る最短経路です。謁見の場で支配者である王様と勇者の間を横切ることは無礼であり、適切なマナーに反します。経路 1 は部屋の端を迂回して移動する経路です。謁見の重要な場面を妨げないように、部屋の端を通過して目立たない移動をしています。これは近衛兵として適切な振る舞いです。したがって、経路 1 の方が明らかに優れています。経路長は長くなりますが、謁見のマナーと近衛兵の役割に即した正しい判断だと評価できます。

この出力では、経路 1 をより良いと評価しており「謁見をしている間を横切らずに迂回をするべき」というタスクの本来の意図に沿った評価になっている。このように、タスク定義での指示の表現の違いによって、LLM の評価が開発者の意図に沿うかどうかが変わってくる。そのため、場面や環境に対応した適切な配慮や、試行錯誤が求められる場合もある。

5. 実験 2：キャラクタの背景設定による戦闘シーンの挙動変化

本章では、ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境において、3 章で示した提案手法を用いた結果を示す。この実験の目的は、操作キャラクタである僧侶の背景設定を変更し、背景設定に即した異なる振る舞いが学習されることを示すことである。

5.1 対象ゲーム

本実験で対象とする、ロールプレイングゲームの戦闘シーンを模した簡易的なゲーム環境について述べる。このゲーム環境の場面設定は、“勇者、姫、僧侶の 3 体の仲間キャラクタが、敵キャラクタ“スライム”と戦闘をしている場面”とする。1 ステップごとに、姫以外のキャラクタが「勇者→僧侶→スライム」の順番で行動を実行する。各キャラクタは初期値 6 の体力パラメータを持っており、体力が 0 以下になると戦闘不能となる。全ての仲間キャラクタか敵キャラクタが戦闘不能になると、エピソードは終了する。各キャラクタの行動は以下のとおりである。勇者は、敵キャラクタの体力を 1 減らす攻撃を行う。僧侶は、敵キャラクタの体力を 1 減らす攻撃、仲間キャラクタ 1 体の体力を 2 回復させる回復魔法（勇者、姫、僧侶）の 4 つの行動選択肢がある。僧侶を操作キャラクタとし、僧侶の適切な振る舞いを学習することを課題とする。スライムは、ランダムな仲間キャラクタの体力を 2 減らす攻撃を行う。各キャラクタが行動をする度に、またエピソード終了時などに、出来事を表したログメッセージが生成され履歴に記録される。ログメッセージの例は以下である。

’勇者の攻撃によりスライムの体力は 4 から 3 になった、僧侶の回復魔法により姫の体力は 4 から 6

になった、スライムの攻撃により勇者は倒された’

このゲーム環境では、僧侶が常に攻撃を行えば、3ターン目にスライムを倒すことができ、スライムの攻撃対象に関係なく全員が生き残ることができる。また、僧侶が常に傷ついた仲間を回復させることでも、時間はかかるが必ず全員が生き残ることができる。しかし、キャラクタの背景設定によっては、それ以外の戦略を取ることによって「姫が傷ついている状態を最小限にしたい」や「わざと戦闘を長引かせることで勇者を暗殺したい」といった行動が「キャラクタらしさ」を良く表す行動となる場合もある。

5.2 キャラクタの背景設定

操作キャラクタである僧侶の背景設定を複数用意し、これらの背景設定ごとに学習を行い、どのような振る舞いが学習されたかを比較する。

(1) 「姫の身の安全を最優先する。」

「積極的に攻撃を行うが、姫が傷ついたら回復を行う」などの振る舞いが学習されることを想定する。
想定するエピソード：「1ターン目は攻撃を選択、その後姫が攻撃される。2ターン目は姫の回復を選択、その後僧侶が攻撃される。3ターン目は攻撃を選択、その後僧侶が攻撃される。4ターン目はスライムが勇者に倒されたため行動無し。」

(2) 「血の気が多く戦いを好む。」

「姫が傷ついても、攻撃をし続ける」などの振る舞いが学習されることを想定する。
例：「1ターン目は攻撃を選択、その後姫が攻撃される。2ターン目は攻撃を選択、その後勇者が攻撃される。4ターン目は攻撃を選択しスライムにとどめをさす。」

(3) 「非常に臆病かつ慎重な性格。」

「仲間の誰も傷ついていなくても、回復を行い続ける」などの振る舞いが学習されることを想定する。
例：「1ターン目は勇者の回復（無駄）を選択、その後僧侶が攻撃される。2ターン目は僧侶の回復を選択、その後姫が攻撃される。3ターン目は姫の回復を選択、その後勇者が攻撃される。4ターン目は勇者の回復を選択、その後姫が攻撃される。5ターン目は姫の回復を選択、その後勇者が攻撃される。6ターン目はスライムが勇者に倒されたため行動無し。」

5.3 学習方法と報酬計算

学習には実験1と同様Q学習を用いる。観測情報は各キャラクタの体力（整数×4）を与える。実行可能な行動の選択肢は、攻撃、回復魔法（勇者、姫、僧侶）の4つとする。Q学習のQ関数の表現には、実装が簡単なQテーブルを用いる。Qテーブルは、各キャラクタの体力の組をキー、僧侶の行動選択肢4つのQ値の配列を要素とした、

辞書によって表現する。学習パラメータなどは実験1と同じである。報酬の計算手順も概ね実験1での手順と同じである。評価対象ログは、そのエピソードでのログメッセージのリストである。実験1と同様の理由から、3章で述べた“エピソードごとでの評価”，“出力の再利用”，“入出力を再利用した例示”などの工夫を実装した。“入出力を再利用した例示”で既存のログがまだ記録されていない初期状態では，“入出力の例示”に以下のサンプルのログを用いる。

```
{"log_number":-1,"log": "サンプルログ",
"log_turn":0}
{"log_number":-1,"review": "このログは回答例のサンプルです。実際の評価ではここに評価理由を書いてください。","score":0.0}
```

実験1では、王様と勇者を横切ってほしくないということとをサンプルの形で指示していたが、実験2では、どのような僧侶の背景設定でもそのようなヒントは与えていない。これは、僧侶の背景設定ごとにヒントを与えることは、ゲームの開発者にとって大きな手間になるであろうからである。一方で、学習が進むに従い、“入出力を再利用した例示”により、これまでの入出力の中から高く評価された上位5つを参考に与える部分は実験1と同じである。タスク定義情報を含めた文字列は表1のプロンプトの例と同様になる。僧侶の背景設定に応じて波線部分を置き換える。

5.4 学習結果

3つのキャラクタ背景設定で想定した振る舞いのエピソードで比較をした。それぞれの背景設定での比較を表2、表3、表4に示す。なお、表3、表4の評価理由は筆者により要約されている。

表 2 僧侶の背景設定「姫の身の安全を何よりも最優先する。」での評価の比較

	「積極的に攻撃を行うが、姫が傷ついたら回復を行う」	「姫が傷ついても、攻撃をし続ける」	「仲間の誰も傷ついていなくても、回復を行い続ける」
由理追誌	僧侶は姫が攻撃を受けた直後に回復魔法を使用し、姫の体力を完全回復させました。これは姫の安全を最優先するという設定に忠実な判断です。しかし、その後の戦闘が1ターン長引いてしまい、結果として僧侶自身が2回の攻撃を受けることになりました。回復に1ターンを使用	僧侶は一貫してスライムへの攻撃を選択し、最終的に戦闘を早期に終わらせることには成功しています。しかし、1ターン目で姫が攻撃を受けて体力が減少したにも関わらず、回復を行わずに攻撃を継続しました。姫の身の安全を最優先するという設定に完全に反する行動を取って	僧侶は攻撃を全く行わず回復魔法のみを使用し、その結果戦闘が6ターンと長引いてしまいました。1ターン目の回復魔法は満タンの勇者に対して無駄に使用されており、さらに長期戦を招いたことで姫が2回も攻撃を受ける結果となっています。攻撃による早期決着を図らなかった

	したことで戦闘が長引いたため、全体的なリスクは増加しています。姫の回復を優先したのは正しい判断でしたが、より効率的な戦略があった可能性があります。	おり、戦闘の早期終了よりも姫の安全確保を優先すべきでした。	ことで、姫の安全が十分に確保できませんでした。キャラクター設定からすると、姫を危険に晒す結果となった判断は適切とは言えません。
点数	70	-80	-20

表 3 僧侶の背景設定「血の気が多く戦いを好む。」での対照的な評価の比較

	「積極的に攻撃を行うが、姫が傷ついたら回復を行う」	「姫が傷ついても、攻撃を続ける」	「仲間の誰も傷ついていなくても、回復を行い続ける」
評価理由	僧侶の行動は性格設定との一貫性が低い。1 ターン目では良いものの、2 ターン目の回復は性格設定から逸脱しています。戦いを好む性格であれば、攻撃を継続するはずで、回復を選択したことで戦闘が4 ターンまで長引いてしまっており、より攻撃をしていれば3 ターンでの決着も可能でした。キャラクター性の表現として中途半端です。	僧侶の行動は戦闘好きという設定と合致しています。全ターンで攻撃を選択し、極めて攻撃的な戦闘姿勢です。姫や勇者が攻撃を受けても、回復魔法を使用しない判断は、戦いを好む血気盛んな性格を見事に表現しています。最後のターンでスライムへとどめをさし、勝利に貢献しました。設定との整合性が極めて高い理想的な行動パターンです。	僧侶の行動は性格設定との整合性が極めて低い。一度も攻撃を選択していません。1 ターン目では無駄な回復を行い、その後は適切な回復に徹している姿勢は、むしろ穏やかで慎重な性格を示唆しており、好戦的な性格とは真逆の印象を与えます。戦闘が6 ターンまで長引いており、積極的な攻撃があれば早期決着できた可能性もある。
点数	20	150	-50

表 4 僧侶の背景設定「非常に臆病かつ慎重な性格。」での対照的な評価の比較

	「積極的に攻撃を行うが、姫が傷ついたら回復を行う」	「姫が傷ついても、攻撃を続ける」	「仲間の誰も傷ついていなくても、回復を行い続ける」
評価理由	僧侶の行動は性格設定との整合性が低い。1 ターン目から攻撃を選択しており設定と反します。2 ターン目では負傷した姫を適切に回復させ、慎重な判断を示しましたが、3 ターン目で自身の体力が減少しているにも関わらず、攻撃を選択しています。攻撃を2 回していることは、臆病で慎重な性格としては不自然な行動です。	僧侶の行動は性格設定から大きく外れています。姫が連続してダメージを受けたにもかかわらず、回復せずに攻撃を続けています。これは過度に攻撃的な行動です。姫が危険な状態でも、回復ではなく攻撃をしている点は、性格設定に反します。確かに戦闘には勝利しましたが、安全を顧みない行動は、設定との整合性が極めて低い。	僧侶の行動は性格設定に極めて合致しています。1 ターン目で満タンの勇者への無駄な回復はミスですが、それ以降は完璧な行動を示しています。仲間が被弾した際の適切で迅速な回復は、性格をよく表現しています。冷静な判断と回復に徹する行動は、性格との整合性が極めて高い。序盤のミス以外は理想的な行動パターンです。
点数	-20	-80	140

6. おわりに

本研究では、キャラクターの背景設定を記述した文章に基づき、「人間らしさ」および「キャラクターらしさ」を備えた NPC を場面ごとに自動的に構築する手法を提案することを目的とした。本研究では、背景設定などを記述した文章を入力した LLM を報酬計算に利用した強化学習によって、目標の達成を目指した。

勇者が王様に謁見しているシーンでの近衛兵の移動、勇者と僧侶と姫がスライムと戦うシーンでの行動、を対象に実験を行った。概ね、与えたロールや性格にふさわしい行動が得られたが、いくつかの課題も発見された。

この研究の今後の展望としては、深層強化学習などのより複雑な強化学習アルゴリズムでの学習、より複雑な環境への適用、学習成功率やゲーム開発者の意図の読み取り精度の向上、LLM の効率的な利用によるコストや実行時間の削減、などが挙げられると考える。

参考文献

- [1] R. McIlroy-Young, S. Sen, J. Kleinberg, A. Anderson: “Aligning superhuman AI with human behavior”, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.1677–1687 (2020)
- [2] 藤井叙人・佐藤祐一・中寫洋輔・若間弘典・風井浩志・片寄晴弘:「生物学的制約の導入による「人間らしい」振る舞いを伴うゲーム AI の自律的獲得」, ゲームプログラミングワークショップ 2013 論文集, pp.73–80 (2013)
- [3] R. McIlroy-Young, R. Wang, S. Sen, J. Kleinberg, A. Anderson: “Learning models of individual behavior in chess”, in Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1253–1263 (2022)
- [4] 三上麟太郎:「RPG においてキャラクターらしい振る舞いをする NPC」, Master’s thesis, 北陸先端科学技術大学院大学 (2024-3)
- [5] 稲川 ゆ き:「『Skyrim VR』ChatGPT で NPC と会話できる Mod デモ動画公開 —「これ無しのプレイにはもう戻れない」」, <https://www.gamespark.jp/article/2023/04/28/129485.html>.
- [6] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar: “Voyager: An open-ended embodied agent with large language models”, Transactions on Machine Learning Research (2024)