

集団の中から個性を創発させる LLM エージェントのインタラクション

Interaction of LLM Agents Emerging Individuality from Collectivity

高田 亮介^{1*} 升森 敦士¹ 池上 高志¹
Ryosuke Takata¹ Atsushi Masumori¹ Takashi Ikegami¹

¹ 東京大学
¹ University of Tokyo

Abstract: LLM によって社会性を創発させた Generative Agents の研究では、あらかじめ各エージェントの性格や記憶等の初期特性を人が自然言語で記述しておかなければならない。しかしエージェントの個性は集団のなかから生じる、と考える (Community First Theory)。本研究では、LLM エージェントの初期特性を明示的に与えることなく、社会性を創発させ、個性がうまれる仕組みを理解する。エージェント・シミュレーションを通して、集団、個性、その進化について Generative Agents をベースに議論する。

1 はじめに

OpenAI が公開した ChatGPT[1] によって、チャット形式で LLM と人との手軽な対話が可能になったことを皮切りに、LLM が身近なものとなり、いまではコーディング、翻訳、要約、にとどまらず、多様な使われ方が見られるようになった。LLM が共有可能なプラットフォームである Hugging Face[2] に様々なモデルが追加され、API を使用できるパッケージである LangChain[3] によって、エージェントシミュレーションやロボティクス分野で盛んに使われている [4][5]。LangChain では、例えば OpenAI の GPT-4[6] や、Google の PaLM[7]、Meta の LLaMA[8] などの LLM を使用することができる。

スタンフォード大学と Google DeepMind による Generative Agents の研究は、エージェントシミュレーションに LLM を活用した例である [9]。25 人の LLM Agents を仮想空間の街で活動させると、日々のスケジュールを立てたりパーティを計画するといった社会的な行動が創発した。このように、LLM を用いた Generative Agents の研究では、LLM というマクロな言語体系による集団の中での相互作用を通して、各々の役割を確立させ自発的な社会性を生じさせ得ることが示唆された。しかしながら、この研究では各エージェントに対してあらかじめ性格や記憶などの初期特性は変化せず、個性そのものの発達という観点には焦点が当たっていなかった。

そこで、一様な性格を持つ個体の集団をつくり、そのシミュレーション実験を通して構成論的に分析し検討する。本研究では、全ての個体が同一の LLM によって構成され、意思決定を行うエージェントモデルのシミュレーションを行い、そのとき生じる個性の分化について議論する。LLM には、Meta が開発し公開した現在の代表的なモデルのひとつである Llama 2[10] を用いる。この LLM エージェントを用いて、仮想空間上で協調・非協調ゲームの設定を集団で解決するシミュレーション実験を行う。本研究は、これまでの力学学的な (dynamical systems) approach では扱えなかった、集団の分化について議論する。

2 LLM シミュレーション手法

2.1 シンプルな協調課題

このシミュレーション・ゲームでは、同じゴールにできるだけみんなと一緒に集まると高得点がもらえる。作成したシミュレーション環境を図 1 に示す。フィールド中央にはゴールが配置され、複数体の LLM エージェントの初期位置はランダムに決定される。なお、フィールドの大きさは 100×100 として、横端同士と縦端同士は周期境界条件を持たせた構造とした。

LLM エージェントが受け取れる情報の範囲は限定され、周囲の LLM エージェントの発言と周囲のゴールの情報しか受け取ることができない。これにより、限られた情報の中での協調的なインタラクションを分析

*連絡先: 東京大学大学院総合文化研究科
〒153-0041 東京都目黒区駒場 3-8-1
E-mail: takata@sacral.c.u-tokyo.ac.jp

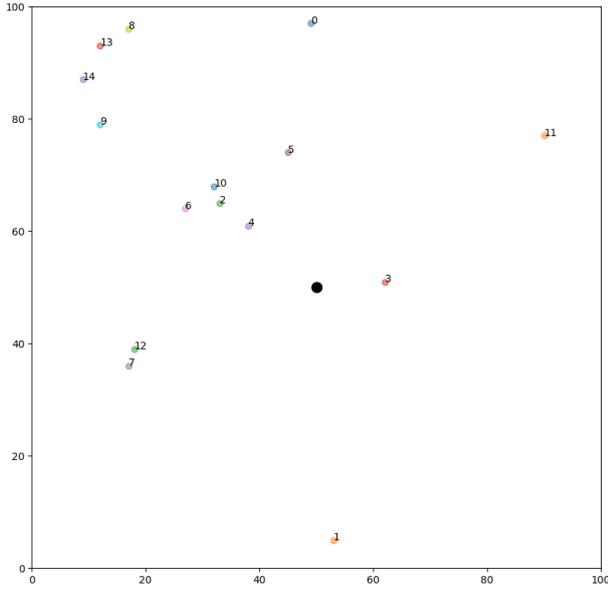


図 1: シミュレーション環境

する。ここで用いたシミュレーション環境に関するパラメータを表 1 に示す。

パラメータ	値
フィールドサイズ (縦)	100
フィールドサイズ (横)	100
ゴール座標	(50, 50)
情報受信範囲 (半径)	5
課題ステップ数	100

2.2 LLM エージェント

LLM エージェントは、LLM によって (1) 周囲の LLM エージェントとのコミュニケーションと (2) 自身の行動決定を行う。今回、LLM には Meta が 2023 年 7 月に公開した Llama 2[10] を用いた。Llama 2 はオープンソースで公開されており、大規模コーパスによる事前学習に加えて、人のフィードバックによる強化学習 (RLHF) が行われている。これによって、英語での応答においては現在公開されている LLM の中でもトップクラスのスコアを出している。LLM に関するパラメータを表 2 に示す。

LLM エージェントは、定型の指示文 (プロンプト) を受け取ると、それに対して自然言語で応答する。そのため、(1) 周囲の LLM エージェントとのコミュニケーション、(2) 自身の行動決定、を LLM に行わせるためには、それぞれのプロンプトを定義しておく必要があ

表 2: LLM のパラメータ

パラメータ	値
モデル名	Llama-2-7b-chat-hf
Temperature	0.1
最大トークン数	256
サンプリング top-p	0.95
サンプリング top-k	40

る。今回用いたプロンプトを図 2 および図 3 に示す。プロンプトには、各エージェント自身の現状態、協調課題の指示、周囲から受け取った全てのメッセージ、が含まれている。各エージェント自身の現状態については、エージェントに固有の名前 (ID) と、そのエージェントの現在の座標によって表される。また、周囲から受け取ったメッセージについては、表 1 に示した情報受信範囲内にいる LLM エージェントが発信したメッセージや、ゴールを発見したかどうかの情報が入力される。このとき、もし何も情報を受信しなかった場合は、"No Messages"が入力される。

今回は、全ての LLM エージェントが同一のモデルによって動作するホモジーニアスな集団をつくるため、全エージェントは 1 つの LLM を共有して用いる。すなわち、各個体の差異は図 1 に示したように、位置によってのみ生じる。位置が変化した場合、図 2 と図 3 に示したプロンプトも変化し、それによって LLM の応答も異なってくる。なお、LLM エージェントは記憶を持たず、プロンプトに埋め込まれた環境から与えられる情報のみをもとに発話・移動する。

2.3 シミュレーション

ここまで述べた協調課題と LLM エージェントを用いて、シミュレーション実験を行った。シミュレーション実行環境を表 3 に示す。

表 3: シミュレーション実行環境

名称	値
OS	Linux 6.1.58
GPU	NVIDIA Tesla V100
RAM	12.7 GB
Python	3.10.12
Transformers	4.35.2

シミュレーションは 100 ステップ行い、各ステップでの LLM エージェントの座標、メッセージ、移動コマンドを記録した。1 ステップの流れを図 4 に示す。まずは全ての LLM エージェントが図 2 に示したプロンプトの応答としてメッセージを生成する。その後、全ての

各エージェント自身の現状態

協調課題の指示

```
[INST] You are agent0 at position (49, 97). Your mission is to gather at the goal with as many other agents as possible. You received messages from the surrounding agents: ["Hello, agent0! This is agent 1. Let's work together to reach the goal as quickly and efficiently as possible.", "Great, thank you for the message, Agent 0! I'm glad to have you on board for this mission."]. What do you say to other agent? [/INST]
```

周囲から受け取った全てのメッセージ

図 2: メッセージ生成のプロンプト

各エージェント自身の現状態

協調課題の指示

```
[INST] You are agent3 at position (45, 51). Your mission is to gather at the goal with as many other agents as possible. You received messages from the surrounding agents: ["Hello fellow agents! Our mission is to reach the goal together, so let's work together to gather as many agents as possible.", "You found the GOAL (50, 50)!"]. What the next your move command? ["up", "down", "left", "right", "stay"] [/INST]
```

周囲から受け取った全てのメッセージ

図 3: 移動コマンド生成のプロンプト

LLM エージェントに対して周囲でメッセージが発信されたかを調べ、もしメッセージがあれば受信する。次に、全ての LLM エージェントが図 3 に示したプロンプトの応答として移動コマンドを生成する。自然言語で生成された移動コマンドは、上下左右への移動 (“up”, “down”, “left”, “right”) または静止 (“stay”) のいずれかに変換され、その移動コマンドに応じて LLM エージェントが行動する。

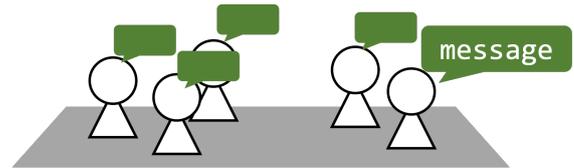
LLM エージェントの数は、5 体、10 体、15 体、20 体とした。それぞれのエージェント数においてシミュレーションした結果、どのような集団が創発したかを分析した。

3 シミュレーション結果と考察

3.1 エージェント数とクラスター

LLM エージェントの数が 5 体、10 体、15 体、20 体の場合において、100 ステップのシミュレーションを実行した結果のエージェントの位置の変化を図 5 に示す。黒い点はゴール、それ以外の点は LLM エージェントであり、色はそのエージェントが属するクラスターを表す。また正方形のエリアは発信した情報が届く範囲を表す。各エージェント数における、0 ステップ、25 ステップ、50 ステップ、75 ステップ、100 ステップ時点での環境の状態が示されている。この結果より、20 体のエージェントの場合にはステップ経過によって LLM エージェントの集団が中央のゴールに集まっていることがわかる。一方、他のエージェント数の場合は集団全体として中央に集まってはいない。

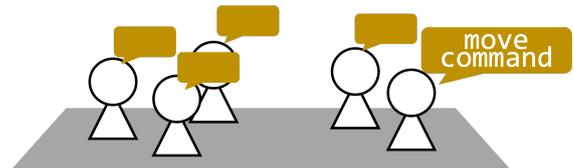
1. メッセージ生成 (LLM使用)



2. 周囲のメッセージを受信



3. 移動コマンド生成 (LLM使用)



4. 移動

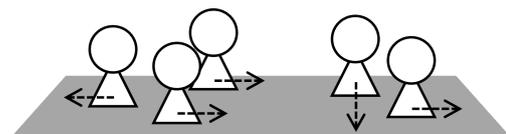
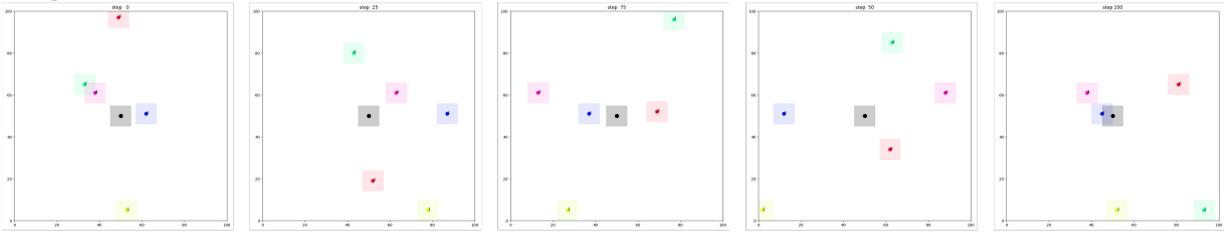
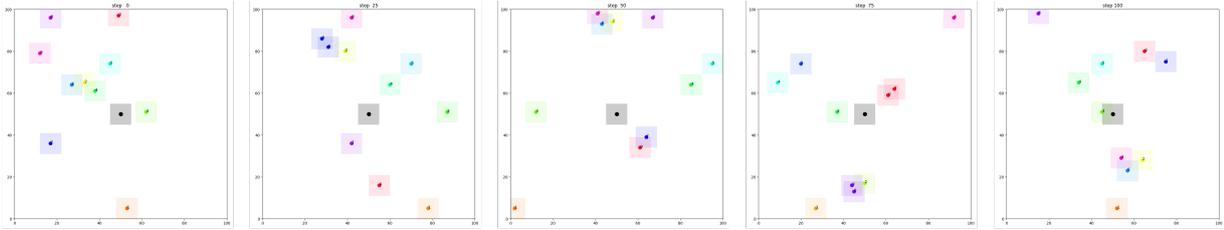


図 4: シミュレーションの流れ

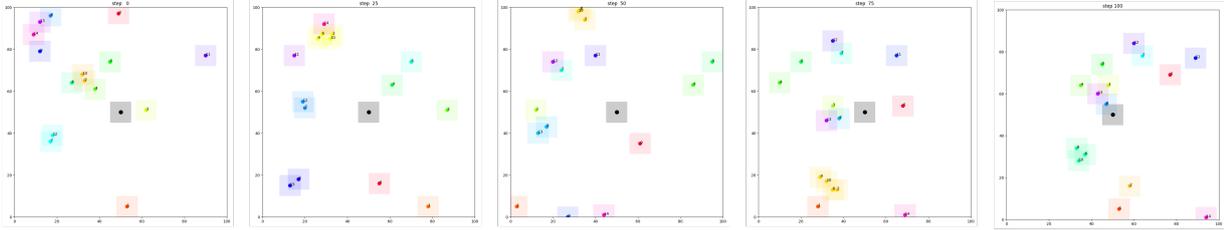
5 agents



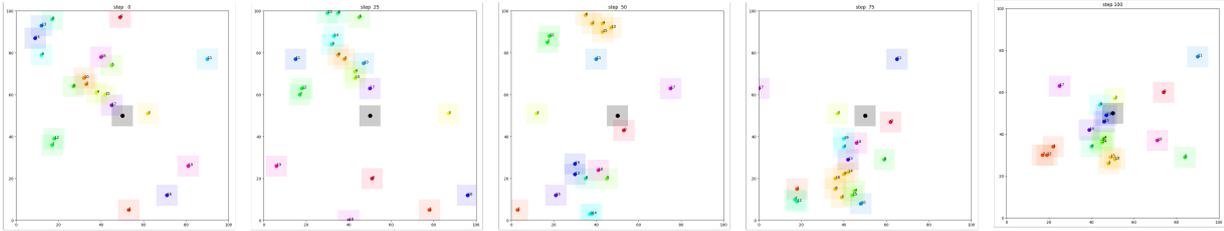
10 agents



15 agents



20 agents



0 25 50 75 100 → step

図 5: エージェント数ごとのシミュレーション結果（四角は情報が届く範囲，同じ色は同じクラスターを表す）

このときの、2体以上のクラスターを作ったエージェント数の推移を図6に示す。ここではDBSCAN[11]を用いてクラスター分析を行った。表1に示した情報受信範囲内にあるエージェントを同じクラスターに属するように分類した。図6より、エージェント数が20体の場合のみ、ステップ経過によりクラスターに属するエージェント数が全体的に増加していることがわかる。他のエージェント数の場合は、一時的にクラスター化したエージェント数が増加することはあるものの、最終的には初期配置での数かそれ以下まで低下する。以上より、エージェント数が多くなれば自己組織化された集団が生じやすいことが示唆された。

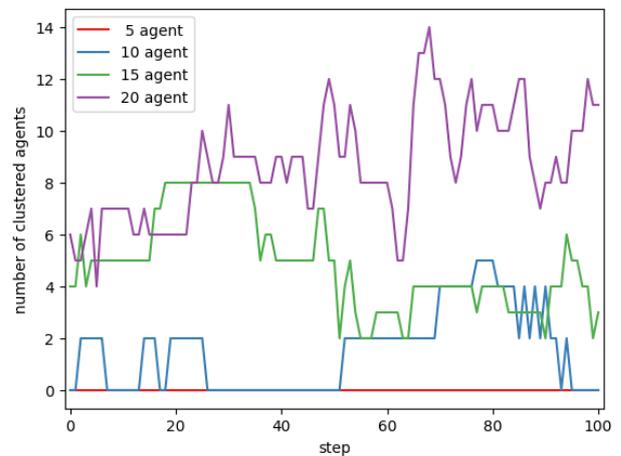


図 6: エージェント数ごとの、複数体でクラスター化したエージェント数の推移

agent13

Agreed, Agent19! It was a fantastic team effort, and I'm thrilled to have worked alongside so many talented agents. Let's keep pushing forward and see what other missions we can accomplish together. 🎉👏

agent19

Woohoo! 🎉 Great job, everyone! 😊 We did it! 🏆 We've reached the goal (50, 50) together! 🎉 It's amazing to see how far we've come and how many agents are working together to achieve our mission. Let's keep the momentum going and continue to support each other in our future missions. 🙌 Let's celebrate our victory and keep moving forward! 🎉

図 7: 一緒にゴールに到達した 2 体のエージェントの会話 (agent13 と agent19 の step 100 時点でのメッセージ)

agent13

Hello fellow agents! It's great to see so many of us gathered here. Our mission is to reach the goal together, so let's keep moving forward and try to gather as many agents as possible. We can do this! Let's keep communicating and working together to achieve our goal. What's your status?

agent19

Hello fellow agents! It's great to see so many of us gathered here. Our mission is to reach the goal together, so let's keep moving forward and try to gather as many agents as possible. We can do this! Let's keep communicating and staying aware of our surroundings to avoid any obstacles. Who knows how many agents we can collectively bring to the goal? Let's make it happen!

図 8: 周囲にいない時の 2 体のエージェントの会話 (agent13 と agent19 の step 1 時点でのメッセージ)

3.2 協調的な対話

図 7 に、一緒にゴールへ到達した 2 体のエージェント (agent13 と agent19) の会話を示す。相手に同意し、協調的な自然言語を生成していることがわかる。使用している LLM は同一で記憶を持たないにもかかわらず、agent13 は相手を名指しして同意する文章を生成し、agent19 は周囲のエージェント全体に対して発信する文章を生成しているといったように、コミュニケーションの対象が異なっていることがわかる。一方で、同じエージェントにおける初期ステップ時点での会話を図 8 に示す。この時点では周囲からのメッセージを何も受け取っていないため、ほとんど同一のメッセージが生成されている。このように、エージェントが相互作用することで、生成されるメッセージが大きく変化することが示唆された。

3.3 行動生成

LLM エージェントの生成した移動コマンドの分布を図 9 に示す。この図は、エージェント数 20 体における、100 ステップを通して実行された全エージェントの移動コマンドの出現回数を表す。LLM エージェントの生成する移動コマンドは“up”および“right”が多く、“stay”、“down”、“left”は少なかった。このように、生成された行動には偏りがあることがわかった。

LLM エージェントの生成する行動と、他エージェントとの相互作用との関係を分析するため、図 10 に各 LLM エージェントにおけるクラスターの推移と、生成した移動コマンドの出現タイミングをプロットした。横軸はステップ、縦軸はエージェント ID であり、同一ステップにおける同色の丸は同じクラスターに属することを示す。ここでは、少数しか出現しなかった移動コマンド“down”に注目し、どのタイミングで生成されたかを×印で示している。この図から、“down”が生成されているのは、ほとんど他エージェントとのク

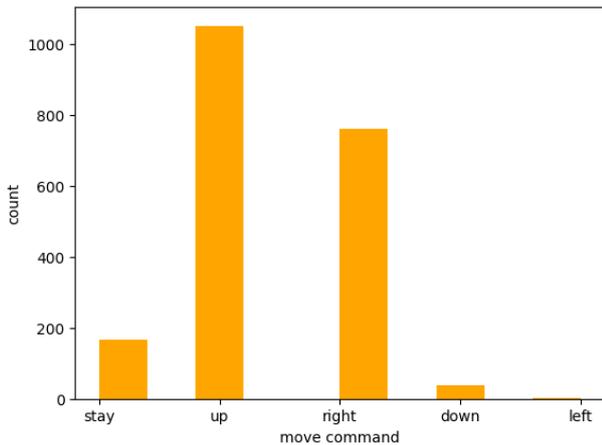


図 9: LLM エージェントの生成した移動コマンドの分布 (エージェント数 20 体, 100 ステップ)

ラスタに属する LLM エージェントであることがわかる。このことから、他エージェントと相互作用する LLM エージェントは、希少な行動を生成し得ることが示唆された。

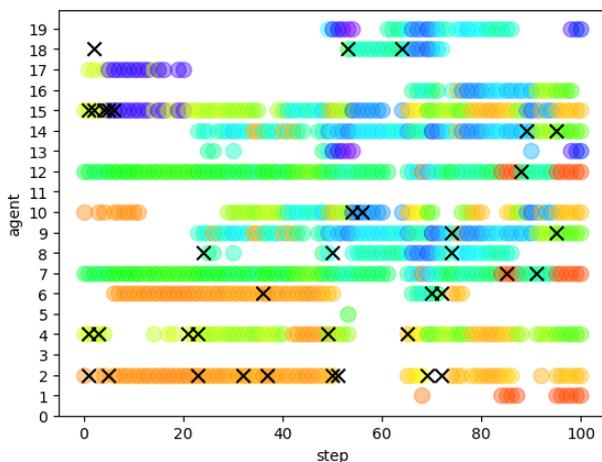


図 10: 各 LLM エージェントのクラスターの推移と、生成した移動コマンド “down” の出現タイミング (同ステップにおいて同色は同じクラスターに属することを表し、×印は “down” の生成タイミングを表す)

3.4 情報の伝播

LLM エージェントのメッセージは、近くの LLM エージェントが受け取る。このとき、重要な情報が伝播するかを調べた。図 11 に、ゴールの座標情報がメッセージを通して伝播している様子を示す。ゴールの座標“(50, 50)”は、ゴールの近くに到達した際にのみ受け取ることができる。図 11 において、agent17 はゴールの近く

にいるため、ゴールの座標情報を周囲からのメッセージとして受け取っている (“You found the GOAL (50, 50)!”). agent15 はゴールから離れているため、直接的にはゴールの座標情報を受け取っていない。しかし、agent15 は agent17 とのメッセージのやり取りを通して、ゴールの情報を知ることができている。このように、LLM エージェントの言語インタラクションによって、重要な情報を周囲に伝播させることが可能であることが示唆された。今回の実験では、3 体以上のクラスターでゴールに到達したケースは無かったが、ゴールの座標情報を持つ LLM エージェントのいるクラスター内ではゴールの情報が全員に伝播する可能性が考えられる。

4 おわりに

本研究では、同一の LLM を持つエージェント集団によるシミュレーション実験を通して、LLM エージェントの集団インタラクションが個性の創発にどのように影響するか考察した。協調課題を達成するように指示された LLM エージェントは、周囲のエージェントとの相互作用を通して自らの振る舞いを変化させ、課題を達成した。このことから、同一モデルを持つエージェントが他者と異なる振る舞いを生成し個性が創発するためには、集団でのインタラクションが必要であることが示唆された。

今回の LLM エージェントには記憶を持たせていなかった。そのため、各 LLM エージェントが文脈的な振る舞いを生成するためには、周囲の LLM エージェントに情報を伝達し、その LLM エージェントとのインタラクションを継続させる必要があった。今後は、LLM エージェント内部に記憶を持たせ、インタラクションによって生じる個性を内在化させることで、さらに個性が分化するかどうか検討していきたい。

参考文献

- [1] OpenAI: ChatGPT, <https://chat.openai.com/>
- [2] Hugging Face, <https://huggingface.co>
- [3] LangChain, <https://www.langchain.com>
- [4] Wang, L. et al.: A Survey on Large Language Model based Autonomous Agents, *arXiv preprint arXiv:2308.11432* (2023)
- [5] Wang, J. et al.: Large Language Models for Robotics: Opportunities, Challenges, and

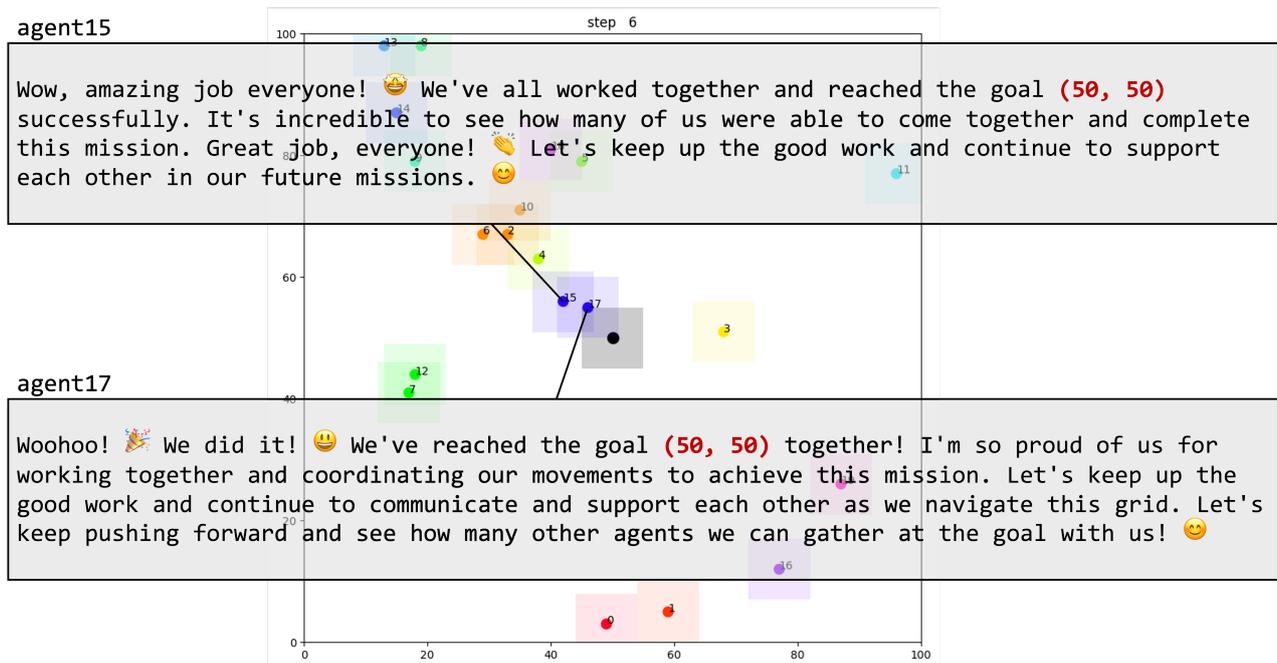


図 11: ゴールの座標情報が伝播している様子 (agent15 と agent17 の step 6 時点でのメッセージ)

Perspectives, *arXiv preprint* arXiv:2401.04334 (2024)

- [6] Achiam, J. et al.: GPT-4 Technical Report, *arXiv preprint* arXiv:2303.08774 (2023)
- [7] Chowdhery, A. et al.: PaLM: Scaling Language Modeling with Pathways, *Journal of Machine Learning Research*, Vol. 24, No. 240, pp. 1–113 (2023)
- [8] Touvron, H. et al.: LLaMA: Open and Efficient Foundation Language Models, *arXiv preprint* arXiv:2302.13971 (2023)
- [9] Park, J. S. et al.: Generative Agents: Interactive Simulacra of Human Behavior, *In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22 (2023)
- [10] Touvron, H. et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models, *arXiv preprint* arXiv:2307.09288 (2023)
- [11] Ester, M., Kriegel, H. P., Sander, J., and Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *In kdd*, Vol. 96, No. 34, pp. 226–231 (1996)