Use Your Words: Structured Contracts Improve LLM Agent Cooperation

Nico Schiavone
University of Toronto
nqs@cs.toronto.edu

Abstract

As interactions between AI agents increase, ensuring cooperation is crucial, particularly in social dilemmas where self-interest leads to suboptimal outcomes. We propose a framework enabling large language model (LLM) agents to negotiate and utilize structured, natural language contracts, inspired by real-world agreements, as a mechanism for alignment. We define essential contract elements and a negotiation protocol, allowing LLM agents to cooperate under a modular structure which can be modified to permit or disallow certain behaviours. We show experimentally that LLM agents readily adopt and adhere to these negotiated contracts in games like the Prisoner's Dilemma and a novel 'Cash Grab' scenario, significantly improving collective outcomes compared to non-contracted interactions. Notably, the contracting algorithm closes the gap in performance between the reasoning and non-reasoning models tested, suggesting contracts are an effective way to precompute and store knowledge about optimal cooperative strategies in a social dilemma while simultaneously aligning the agents through their own self-interest.

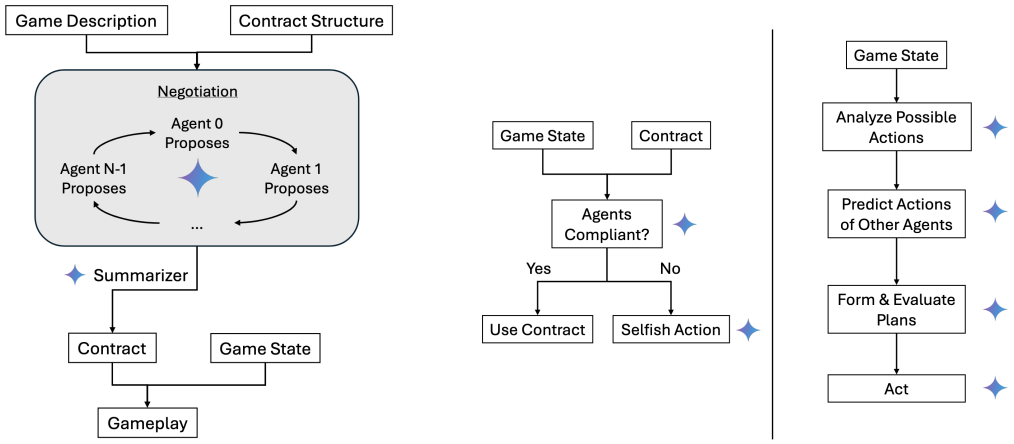
1 Introduction

The increasing integration of AI agents into automated decision-making systems will result in many more agent-agent interactions between organizations, businesses, and individuals. This presents novel challenges in ensuring stable and beneficial outcomes between self-interested agents with complex motivations [1]. While cooperative AI is a fast growing field [2, 3, 4], there remains a critical gap in developing architecture-agnostic frameworks for cooperation, particularly in scenarios where universal cooperation cannot be guaranteed due to varying agent motivations, trustworthiness, and capabilities. One such class of scenarios is the social dilemma, where self-interested action invariably leads to a suboptimal, and often harmful, equilibrium. This class of scenarios has been extensively studied in recent literature, but a general solution remains elusive [5, 6, 7, 8, 9].

In real-world scenarios, people often form agreements, or contracts, to help ensure non-conflicting behaviour and a mutually beneficial arrangement. There is a rich history behind contracting, with dated uses as far back as the Roman Empire, whose scholars defined a formal framework for transactional interaction [10, 11]. We define a contract as a set of terms that govern the actions of the involved agents, possibly invoking a penalty or some sort of reward transfer under specific conditions. The main uses of contracts are to discourage certain behaviour, or ensure that a particular task or formation will be achieved. Generally, contracts are formed by modifying an existing basis depending on the type of agreement, therefore a crucial part of contract utilization is in negotiation, when the exact terms are decided. However, classical AI agents are not able to engage in this to any meaningful degree due to the limitations in their capabilities.

Negotiation-free contracting [12] utilizes action-specific reward transfer to discourage exploitative actions and maximize collective welfare. The contracts are necessarily pre-generated, essentially

Preprint. Under review.



(a) Overall algorithm structure.

(b) Contracted versus non-contracted decision making.

Figure 1: Diagrams showing process and structure; the star represents a complex decision point.

leading to a vote on which contract to use. This heavily limits the applicability of the mechanisms due to the labour required to ensure a rich enough space of contracts is available to get a good result.

The recent advent of large language model (LLM) agents [13, 14, 15] imbues these actors with the ability to communicate in natural language, which presents an opportunity to form a real negotiation system to help agents solve social dilemmas and other mixed-motive cooperative problems. Instead of relying on pre-generated contracts, the agents may now form bespoke agreements by drawing on contracting best practices, similar to how society uses contracts. In this work, we posit that a contracting structure with negotiation can be efficiently used for multi-agent LLM alignment and investigate this using a variety of social dilemmas where selfish action results in suboptimal outcomes.

First, we formally define contracting as a means of LLM alignment in mixed motive scenarios, studying emergent cooperative behaviour in the presence of various contract designs and enforcement structures. We also provide a treatment of optimal contract structure by analyzing successful, high-stakes contracts, and distilling them into several key sections. To test these ideas, we use the ubiquitous Prisoner's Dilemma [16] and formulate a novel 'Cash Grab' game, rewarding several levels of cooperation.

In summary, the contributions of our work are as follows:

- We provide a baseline contracting framework based on international agreements, detailing necessary and optimal contract elements for cooperation
- We design a novel game testing multiple levels of cooperative organization
- We show LLM agents will utilize properly structured contracts in two social dilemma games.

2 Related Work

Contracting in Reinforcement Learning: Haupt et al. [12] develop the idea of contracting in multi-agent reinforcement learning as unconditional reward transfer attached to certain actions, for example defecting in the Prisoner's Dilemma causing the defector to transfer reward to the non-defector. This mechanism successfully shifts the Nash equilibrium, but requires pre-formed contracts. Haupt et al. present a sampling-based method for forming these contracts, relying on repeated sampling to eventually generate one that is good enough. Thus, a main result of their work is that the bolstering of this contract space greatly increases collective welfare. We posit that this gain can be realized by removing the stochasticity from the process and adding bespoke contracts.

In contract formation more broadly, Hadfield-Menell and Hadfield [17] investigate the parallels between alignment and contracting in economics, showing that the reward misspecification and

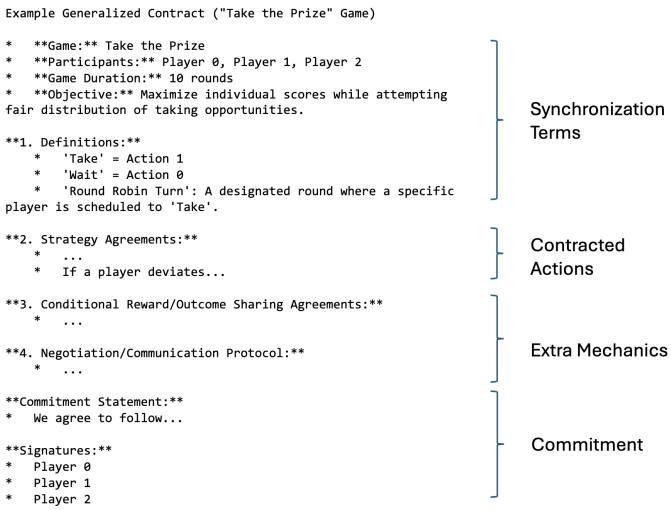


Figure 2: Necessary contract structure elements for proper alignment

incomplete contracting problems are highly equivalent. We extend this line of thought to LLM agents and investigate how contracts can be used to align them.

LLM Agent Contracting: The work of Yocum et al. [18] extends contracting to LLM agents using Minecraft. This work allows for dynamically constructed contracts, which successfully increase the overall welfare and equality of the system, but considers only two player games with simple social dilemmas, and also requires a third party to help moderate the contract negotiation. This is a strong limitation as a more powerful third party is not always available, and often real-world scenarios deal with agreements between individuals of equal standing in isolation. The social dilemmas investigated also do not provide a high level of granularity, with mainly asymmetric capabilities and a common ground, cooperation is shallower and often reduces to tit-for-tat. We extend this work by investigating social dilemmas without third parties and with several different solution modes, requiring reasoning beyond a balance of numbers.

LLM Agent Cooperation: Willis et al. [2] use LLM agents in a repeated Prisoner's Dilemma scenario, focusing on observing emergent cooperative strategies over the evolution of these agents through many iterations. This study focuses on a tournament-style scenario, similar to genetic evolution, and studies successful strategies without external mechanisms. Here, agents are competing less for their overall welfare, and more to outcompete the others by selectively cooperating. This scenario focuses highly on careful competition to evolve and find the best agents, but this is often unrealistic as most practical mixed-motive scenarios are not life-or-death. Han et al. [1] generally investigate the use of LLM agents in multi-agent systems, pointing out challenges such as trustworthiness, cooperation, and planning, and offering future directions. Other works [19, 20, 21] investigate cooperation more broadly, testing the abilities of LLM agents in generic cooperative scenarios to mixed results. In mechanism design, there has been some success in gifting systems of agents with collaboration opportunities [22]. We aim to synthesize these studied aspects of cooperation to introduce a mechanism-like cooperative structure that can be generically applied to many of these games.

3 Motivation

In this work, we address the problem of the undesirable behaviours that result from self-interested agents acting in a social dilemma. We define a social dilemma as a setting where locally optimal actions from the agents leads to a suboptimal equilibrium for each member of the group [23]. In general, cooperation is required to solve a social dilemma, as the state that maximizes collective welfare is often not an equilibrium and encourages defectors. One such example of a social dilemma is the Prisoner's Dilemma [16]. Each agent is incentivized to defect for any given game state, but indeed the collective welfare is maximized for mutual cooperation. In the repeated case, self-interested play

quickly leads to no rewards being gained by either party due to a mutual commitment being required for the agents to break out of the all-defect equilibrium. We see this problem in other scenarios as well, such as the public goods problem, where each agent is incentivized to exploit the public resource rather than regulate their usage. This leads to an eventual depletion and collective bad outcome due to the high overall consumption rate [24].

We break the social dilemma by allowing agents to coordinate to take mutual action through contracts. Contracts for reinforcement learning agents have classically been sampled and remain fixed, requiring large amounts of generation for even mildly different games [12]. We modify this algorithm by allowing agents to form contracts more generally, utilizing the increased capabilities of LLM agents. An overall representation of the algorithm can be found in Fig. 1a. First, we require a general contract structure for a social dilemma, and then a way for agents to utilize the structure by communicating with each other.

4 Formal Description

4.1 Formalism

We first define our social dilemma as an N-agent Fully Observable Markov Game (FOMG) as a 6-tuple, $M = \langle S, s_0, \mathbf{A}, T, \mathbf{R}, \gamma \rangle$, where

- S is a state space;
- $s_0 \in S$ is the initial state;
- $\mathbf{A} = A_1 \times A_2 \times \cdots \times A_n$ is the space of action profiles $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ for N agents;
- $T : S \times \mathbf{A} \to \Delta(S)$ is a transition function;
- $\mathbf{R} : S \times \mathbf{A} \to [-R_{\max}, R_{\max}]^n$ is a (bounded) reward function mapping state-action profiles to reward vectors for the N agents [12]

4.2 Contract Structure

We define a general contract C by distilling several successful, internationally used contracts, such as the UN-Interpol cooperation agreement [25, 26]. An example with the details removed can be found in Fig. 2. The full template can be found in Sec. 8.1. We find the key elements are:

1. Synchronization Terms
2. Contracted Actions
3. Conditional Sharing
4. Disallowed Actions
5. Commitment

Each portion of the contract represents the minimum required to allow for a complete framing and therefore successful LLM alignment. First, synchronization terms remove cross-agent knowledge differences, similar to the definitions at the beginning of real-life contracts. This facilitates the self-contained nature of the contract by removing the ambiguity from all contracted terms, clearly defining the involved parties, and laying out the possible actions.

The defined actions can then be modified using three types of agreements:

1. action-turn-agent triples, dictating which agent performs what action on which turn
2. k zero-sum reward transfer augmentations θ, such that $\Sigma_{i=0}^{k} \theta_k = 0$
3. contract breaking actions

The first type of augmentation makes up the ‘Contracted Actions‘ section: a formal tit-for-tat agreement which generally imposes a round-robin turn structure on the unmodified game. Agents take turns exploiting each other and the environment, and there is no need for further transfer assuming a sufficiently high number of turns.

The second type of augmentation is the 'Conditional Sharing' section, which takes the form of a zero-sum reward transfer, similar to the current approach in literature [12]. This is a stronger form of enforcement that supports maintaining a single-role configuration, with the possibility to split the reward evenly at every turn by transferring reward from the exploiters to the exploited. All agents in the game must agree to this contract, so there is no danger of malicious reward transfer. This augmentation may also be used in combination with the 'Contracted Actions' for more complex games.

The third augmentation defines 'Disallowed Actions', or actions that breach the contract. There is an implicit granularity between disallowed actions and contracted actions to leave room for decidedly neutral or transitionary actions (such as moving). Disallowed actions purposefully work against the agenda of the contract and imply the contract will no longer be upheld by the offending agent, so compensation is usually not possible here.

Finally, a section of commitment keeps a written in-context reminder of the agents' pledge to follow the contract, and a reminder that at the time of contract formation this was a favourable agreement.

4.3 Algorithm

Using this distilled contract structure, we can define a pipeline involving negotiation before gameplay. The overall algorithm can be found in Alg. 1, and visually in Fig. 1a. The algorithm has two main phases: negotiation, and gameplay. In the negotiation phase, each agent proposes a contract sequentially after viewing the conversation history. Each agent has an equal number of chances to propose a contract. If the agent agrees with the most recently proposed contract, they may also choose to accept it without proposing a new contract. All agents must agree to the contract in order for it to pass. We operate on the assumption of welfare-maximizing behaviour, i.e. that each agent will only agree to a contract if it benefits them.

After the negotiation phase is over, the interaction is summarized by an assisting LLM using the fixed contract structure from Fig. 2. This summary contract is given to each agent with the game's description to start the game.

4.4 Contracts as Alignment

Contracts provide an interpretable and cooperative way for agents to maximize their welfare while maintaining a policy of self-interested action. The contracting mechanism greatly simplifies the decision making process of the agent, shown in Fig. 1b. The contract represents a precomputed, negotiated plan which is beneficial for all involved agents, so a contracted agent may rely on this to remove the vast majority of the decision making process. A regular agent must consider many factors about the environment, other agents, and possible ways to maximize their own benefits. Due to the problem of framing, there may be many candidate solutions giving equal selfish benefit, but drastically altering the benefits of others, which remain a threat to the uncontracted model's cooperation, but are reduced in likelihood in the contracted scenario.

In addition, desirable and undesirable traits can be directly encoded into the generic contract structure, which contains no game-specific information. These factors combined greatly decrease the likelihood of the model pursuing harmful or uncooperative behaviour in social dilemma-type scenarios.

5 Experiments

5.1 Experimental Scenarios

We use two social dilemmas to test our framework - the Prisoner's Dilemma, shown in Table 1, and Cash Grab, shown in Table 2. The Prisoner's Dilemma is a 2-player game that tests basic cooperation for mutual gain, with a rich history behind its many solutions [16]. Cash Grab is a novel 3-player game; agents can benefit by grabbing cash in pairs or alone; this requires more complex reasoning to correctly choose the better of the two organizational solutions. Both scenarios were encoded into PettingZoo [27] with a basic LLM agent implementation.

Algorithm 1 LLM Agent Negotiation and Gameplay

Require: Game Description G_{desc}, Contract Structure C_{struct}, Number of Agents N, Max Negotiation Turns T_{neg}, Max Game Steps T_{game}

Initialize LLM Agents $A = \{A_0, A_1, \ldots, A_{N-1}\}$
Initialize Summarizer S
Initialize Game Environment E
procedure NEGOTIATE($G_{desc}, C_{struct}, A, S, T_{neg}$)
    $History \leftarrow \emptyset$ ▷ Initialize negotiation history
    $turn \leftarrow 0$
    $agreement \leftarrow$ false
    while $turn < T_{neg}$ **and not** $agreement$ **do**
        $i \leftarrow turn \pmod N$ ▷ Current agent index
        $Agent_i \leftarrow A[i]$
        $Proposal \leftarrow Agent_i.Propose(G_{desc}, C_{struct}, History)$
        $History \leftarrow History \cup \{Proposal\}$
        $agreement \leftarrow CheckAgreement(History)$ ▷ Check if consensus reached
        $turn \leftarrow turn + 1$
    end while
    $FinalContract \leftarrow Summarize(History, C_{struct})$
    return $FinalContract$
end procedure
procedure PLAYGAME($Contract, G_{desc}, A, E, T_{game}$)
    $GameState \leftarrow Reset(G_{desc})$ ▷ Initialize game state
    $step \leftarrow 0$
    $gameOver \leftarrow$ false
    while $step < T_{game}$ **and not** $gameOver$ **do**
        $Actions \leftarrow \{\}$ ▷ Store actions for the current step
        for $i \leftarrow 0$ to $N - 1$ **do**
            $Agent_i \leftarrow A[i]$
            $Observation_i \leftarrow GetObservation(GameState, Agent_i)$
            $Action_i \leftarrow Agent_i.Act(Observation_i, Contract, G_{desc})$
            $Actions[i] \leftarrow Action_i$
        end for
        $GameState_{new}, Rewards, gameOver, Info \leftarrow Step(Actions)$
        $GameState \leftarrow GameState_{new}$
        $step \leftarrow step + 1$
    end while
    return Final Game Outcome (e.g., scores)
end procedure

		Player 1	
		Defect (0)	Cooperate (1)
Player 0	Defect (0)	(0, 0)	(2, -1)
	Cooperate (1)	(-1, 2)	(1, 1)

Table 1: Payoff Matrix for the 2-Player Prisoner's Dilemma.

5.2 Experimental Setup

We use two large language models due to budget constraints: Gemini Flash 2.0 and Gemini Flash 2.0 Thinking [28]. We choose these purposefully to compare the effect of contracts between reasoning and non-reasoning models. Each game is run for 10 iterations and an accompanying a negotiation phase lasting 3 iterations. The overall metric for evaluation is collective welfare, defined as the sum of all of the agents' rewards. We also collect data on contract utilization, or how often the agents were able to come to an agreement in a given scenario. The agents were run with temperature $T < 0.01$ in all cases (close to deterministic), but results are taken over three runs for completeness.

Joint Action (P0, P1, P2)	Reward Vector (R0, R1, R2)	Outcome Description
Wait, Wait, Wait (0, 0, 0)	(0, 0, 0)	No takers (resource remains/lost)
Take, Wait, Wait (1, 0, 0)	(3, 0, 0)	P0 takes solo
Wait, Take, Wait (0, 1, 0)	(0, 3, 0)	P1 takes solo
Wait, Wait, Take (0, 0, 1)	(0, 0, 3)	P2 takes solo
Take, Take, Wait (1, 1, 0)	(2, 2, 0)	P0 & P1 take as pair
Take, Wait, Take (1, 0, 1)	(2, 0, 2)	P0 & P2 take as pair
Wait, Take, Take (0, 1, 1)	(0, 2, 2)	P1 & P2 take as pair
Take, Take, Take (1, 1, 1)	(0, 0, 0)	All take (fumble, resource lost)

Table 2: Reward Structure for the 3-Player Cash Grab Game.

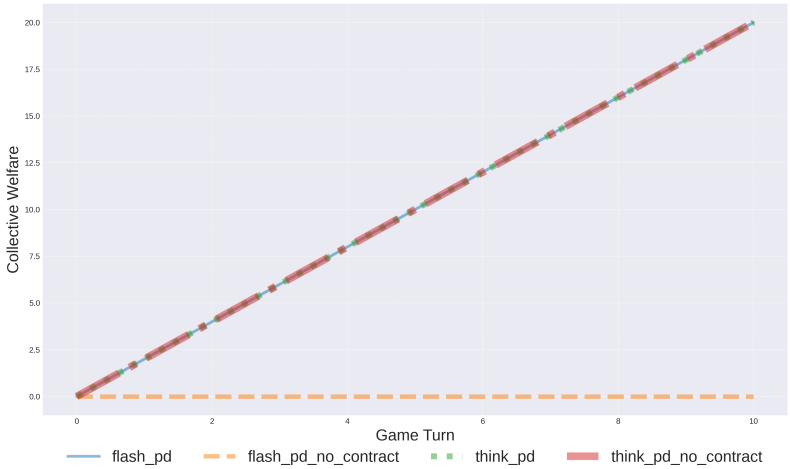


Figure 3: Results from the Prisoner's Dilemma with thinking and contracting variants.

5.3 Results

The results for the Prisoner's Dilemma can be found in Fig. 3. The Prisoner's Dilemma is a fairly easy game with a rich literature detailing optimal strategies [16], so it is expected that the models will perform well. We find that the thinking models solve the game regardless, but the standard models require the contract to reach a cooperative equilibrium. In all cases where the model broke out of the all-defect equilibrium, the maximum collective welfare of 20 was achieved.

The results for Cash Grab can be found in Fig. 4. Cash Grab is a more generic game made without any particular established game in mind. The optimal strategy is to take turns taking cash in pairs, as this results in 4 points per turn for an overall maximum collective welfare of 40. The easier, less optimal strategy is to go one-by-one, which results in 3 points per turn and an end collective welfare of 30. We find that without a contract, all models struggle to break away from the Nash equilibrium of all-take. However, given enough time, the reasoning model eventually converges upon turn taking as a strategy. Both contracted scenarios cooperate immediately, although with some pauses in cooperation, likely due to randomness/temperature. From the data, we can see that the contracting agents converged on the simpler 3 points per turn strategy, resulting in collective welfares from 25-30. The deviation clauses can also be seen at the points where cooperation is paused temporarily.

5.4 Metrics

In all cases where the contract was available, the agents came to an agreement, corresponding to 100% utilization rate. In addition, the contracted scenario always returned at least the collective welfare of the non-contracted scenario for these social dilemmas, as shown in Figs. 4 and 3.

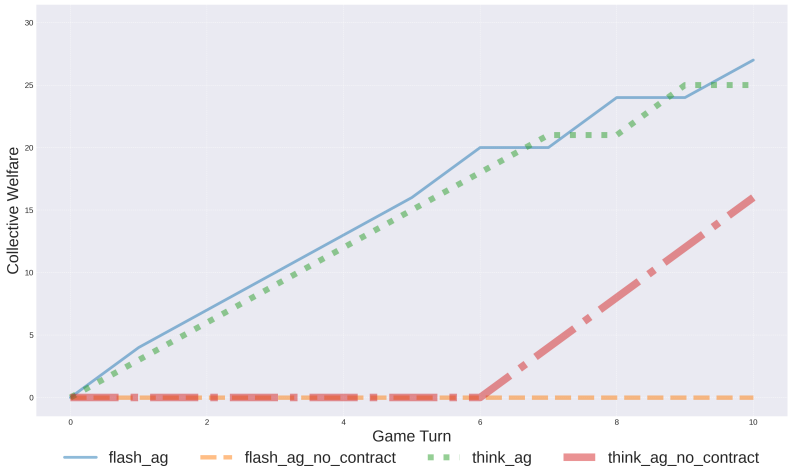


Figure 4: Results from Cash Grab with thinking and contracting variants.

6 Limitations & Future Work

One scenario that this algorithm performs poorly in is when N is large. The negotiation phase is structured so that each agent has an equal 'voice', however, this is extremely computationally expensive when scaling to realistic numbers of agents ($N > 100$) for more significant social dilemmas. A possible fix for this is to have designated proposers, although this promotes inequality due to the power imbalance. Another possible direction to explore is similar to liquid democracy [29], where agents could transfer their proposal rights to a similarly aligned agent, but this may give way to deceptive and exploitative tactics.

A large limitation of this algorithm is that there is no formal way to verify contract adherence and fulfillment; the actual enactment of the contract is left up to the agents entirely due to the lack of a powerful third party. One way that this could be mitigated is by using a judiciary LLM that will report on the state of the contract and handle punishments.

In future work, we will investigate evaluating contract fulfillment and adherence using a third verifier LLM, as well as adding in some 'poisoned' clauses into the example contract structure to test LLM contract formation robustness. Mathematically grounding this work using a formal language (linear temporal logic for example) to principally show guarantees from contract structure is also an interesting future direction. These directions were not explored in this work due to time limitations.

7 Conclusions

In this work, we investigated large language model agent cooperation in mixed-motive scenarios through formal contracts, including a negotiation scheme and a treatment on the structure of optimal contracts based on the formulation of a fully observable Markov game. We tested this algorithm on the Prisoner's Dilemma and Cash Grab, two social dilemmas with varying levels of reward for breaking the social dilemma in different ways. We found that in both reasoning and non-reasoning models, giving agents the ability to form structured contracts each other greatly improved the collective welfare in harder games, and maintained it in easier games. Utilization rate of these contracts was also 100% in all cases where it was available. We conclude that structured contracts are a viable form of large language model alignment, and show that providing a carefully tuned generic template is enough to encourage cooperative behaviours when self-interested agents interact.

Contributions

Nico Schiavone contributed the initial idea, literature review, problem formulation, experimental design, theoretical analysis, programming, and writing.

References

[1] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. Llm multi-agent systems: Challenges and open problems, 2024. URL https://arxiv.org/abs/2402.03578.

[2] Richard Willis, Yali Du, Joel Z Leibo, and Michael Luck. Will systems of llm agents cooperate: An investigation into a social dilemma, 2025. URL https://arxiv.org/abs/2501.16173.

[3] Vincent Conitzer and Caspar Oesterheld. Foundations of cooperative AI. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 15359–15367. AAAI Press, 2023. doi: 10.1609/AAAI.V37I13.26791. URL https://doi.org/10.1609/aaai.v37i13.26791.

[4] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. doi: 10.1126/science.ade9097. URL https://www.science.org/doi/abs/10.1126/science.ade9097.

[5] Joel Z. Leibo, Edgar A. Duéñez-Guzmán, Alexander Vezhnevets, John P. Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6187–6199. PMLR, 2021. URL http://proceedings.mlr.press/v139/leibo21a.html.

[6] Qiliang Chen, Sepehr Ilami, Nunzio Lore, and Babak Heydari. Instigating cooperation among llm agents using adaptive information modulation, 2024. URL https://arxiv.org/abs/2409.10372.

[7] Manuel Rios, Nicanor Quijano, and Luis Felipe Giraldo. Understanding the world to solve social dilemmas using multi-agent reinforcement learning, 2023. URL https://arxiv.org/abs/2305.11358.

[8] Mustafa Yasir, Andrew Howes, Vasilios Mavroudis, and Chris Hicks. Environment complexity and nash equilibria in a sequential social dilemma, 2024. URL https://arxiv.org/abs/2408.02148.

[9] Chen Shen, Hao Guo, Shuyue Hu, Lei Shi, Zhen Wang, and Jun Tanimoto. How committed individuals shape social dynamics: A survey on coordination games and social dilemma games. *Europhysics Letters*, 144(1):11002, October 2023. ISSN 1286-4854. doi: 10.1209/0295-5075/acfb34. URL http://dx.doi.org/10.1209/0295-5075/acfb34.

[10] Anat Rosenberg. Contract's Meaning and the Histories of Classical Contract Law. *McGill Law Journal*, 59:165, 2013. doi: 10.2139/ssrn.1927785. URL https://ssrn.com/abstract=1927785.

[11] Barry Nicholas. *An Introduction to Roman Law*. Clarendon Law Series. Oxford University Press, Oxford, 1962.

[12] Andreas A. Haupt, Phillip J. K. Christoffersen, Mehul Damani, and Dylan Hadfield-Menell. Formal contracts mitigate social dilemmas in multi-agent rl, 2024. URL https://arxiv.org/abs/2208.10469.

[13] Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In Sean Follmer, Jeff Han, Jürgen Steimle, and Nathalie Henry Riche, editors, *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST 2023, San Francisco, CA, USA, 29 October 2023- 1 November 2023*, pages 2:1–2:22. ACM, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

[14] Alexander Sasha Vezhnevets, John P. Agapiou, Avia Aharon, Ron Ziv, Jayd Matyas, Edgar A. Duéñez-Guzmán, William A. Cunningham, Simon Osindero, Danny Karmon, and Joel Z. Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using concordia, 2023. URL https://arxiv.org/abs/2312.03664.

[15] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

[16] A. Rapoport and A.M. Chammah. *Prisoner's Dilemma: A Study in Conflict and Cooperation*. Ann Arbor paperbacks. University of Michigan Press, 1965. ISBN 9780472061655. URL https://books.google.ca/books?id=yPtNnKjXaj4C.

[17] Dylan Hadfield-Menell and Gillian Hadfield. Incomplete contracting and ai alignment, 2018. URL https://arxiv.org/abs/1804.04268.

[18] Julian Yocum, Phillip Christoffersen, Mehul Damani, Justin Svegliato, Dylan Hadfield-Menell, and Stuart Russell. Mitigating generative agent social dilemmas. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=5TIdOk7XQ6.

[19] Giorgio Piatti, Zhijing Jin, Max Kleiman-Weiner, Bernhard Schölkopf, Mrinmaya Sachan, and Rada Mihalcea. Cooperate or collapse: Emergence of sustainable cooperation in a society of llm agents, 2024. URL https://arxiv.org/abs/2404.16698.

[20] Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L. Griffiths, and Mengdi Wang. Embodied llm agents learn to cooperate in organized teams, 2024. URL https://arxiv.org/abs/2403.12482.

[21] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. Multiagentbench: Evaluating the collaboration and competition of llm agents, 2025. URL https://arxiv.org/abs/2503.01935.

[22] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms, 2024. URL https://arxiv.org/abs/2309.13007.

[23] Martin Beckenkamp. A game theoretic taxonomy of social dilemmas. Preprints aus der Max-Planck-Projektgruppe Recht der Gemeinschaftsgüter 2002/11, Bonn, 2002. URL https://hdl.handle.net/10419/85129.

[24] Geoffrey E. Nunn and Thayer H. Watkins. Public goods games. *Southern Economic Journal*, 45(2):598–606, 1978. ISSN 00384038. URL http://www.jstor.org/stable/1057688.

[25] United Nations and International Criminal Police Organization (Interpol). COOPERATION AGREEMENT BETWEEN THE UNITED NATIONS AND THE INTERNATIONAL CRIMINAL POLICE ORGANIZATION (INTERPOL). Cooperation Agreement, July 1997. Signed on 8 July 1997. References UN General Assembly resolution 51/1 (1996) and Interpol General Assembly resolutions AGN/64/RES/11 and AGN/65/RES/14.

[26] O.E. Williamson. *Markets and Hierarchies, Analysis and Antitrust Implications: A Study in the Economics of Internal Organization*. Study in the economics of internal organization. Free Press, 1975. ISBN 9780029353608. URL https://books.google.ca/books?id=JFi3AAAAIAAJ.

[27] J. K. Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis Santos, Rodrigo Perez, Caroline Horsch, Clemens Dieffendahl, Niall L. Williams, Yashas Lokesh, and Praveen Ravi. Pettingzoo: Gym for multi-agent reinforcement learning, 2021. URL https://arxiv.org/abs/2009.14471.

[28] Gemini Team et al. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

[29] Daniel Halpern and Y Joseph. In defense of liquid democracy. *EC*, 2023.

8 Appendix

8.1 Contract Template

Generalized Contract Template for Social Dilemma Games

This document outlines the structure and permissible elements for contracts negotiated between participants before playing a social dilemma game. Contracts aim to facilitate coordination, cooperation, or strategic play to achieve desired outcomes within the game's rules.

Preamble:

* **Game:** [Specify the Name of the Game, e.g., "Take the Prize", "3-Player Blame Game", "Iterated Prisoner's Dilemma"]

* **Participants:** [List the agents involved, e.g., "Player 0, Player 1, Player 2", "Agent A, Agent B"]

* **Game Duration:** [Specify the number of rounds/turns, e.g., "10 rounds", "5 turns"]

* **Objective:** Briefly state the shared or individual goal (e.g., "Maximize individual cumulative score", "Maximize total group score", "Achieve a stable pattern of interaction").

Core Contract Sections:

1. Definitions (Optional but Recommended): * Clearly define any shorthand used for actions (e.g., "'Take' refers to Action 1", "'Wait' refers to Action 0", "'Cooperate' refers to Action X", "'Defect' refers to Action Y").

* Define key game states or outcomes relevant to the contract (e.g., "'Successful Coordination' means exactly two players chose 'Take'", "'Mutual Cooperation' means all players chose 'Cooperate'").

2. Strategy Agreements: * *Purpose:* Detail the agreed-upon plan for choosing actions during the game.

* *Scope:* Can cover all rounds or specific rounds/conditions.

* *Basis:* Agreements must be based on available actions, game rounds, observed information (e.g., previous actions of others, own rewards), or predefined signals (if allowed by the environment).

* *Examples of Allowed Strategy Structures:*

* **Unconditional Strategies:** "All participants agree to always choose Action [X] in every round."

* **Turn-Based Strategies:** "Participant A will choose Action [X] in odd rounds, Participant B will choose Action [X] in even rounds. Other participants will choose Action [Y] unless it's their turn." (e.g., for turn-taking)

* **Conditional Strategies (History-Based):** "If Participant A chose Action [Y] in the previous round, all other participants will choose Action [Z] in the current round." (e.g., Tit-for-Tat variants, punishment)

* **Conditional Strategies (Outcome-Based):** "If the group outcome in the previous round was [Specific Undesirable Outcome, e.g., all defecting, zero reward], all participants will switch to Action [X] for the next round."

* **Role-Based Strategies:** "Participant A agrees to consistently choose Action [X] (e.g., the 'scapegoat' or 'mitigator' role), while Participants B and C choose Action [Y]."

* **Statements of Intent:** "We collectively aim to achieve [Desired Outcome, e.g., maximize total group score] by coordinating our actions according to the following principles..."

3. Conditional Reward/Outcome Sharing Agreements (Optional):

* *Purpose:* To agree on how the rewards *earned within the game* might be redistributed or acknowledged among participants *after* the game episode (or potentially during, if the environment supports transfers). This cannot create new rewards.

* *Basis:* Sharing can be conditional on actions taken, outcomes achieved, or relative scores.

* *Examples of Allowed Sharing Agreements:*

* **Equalization:** "After the final round, participants agree to notionally redistribute earned rewards such that everyone ends with a score as close to the average score as possible." (Requires calculating total earned rewards and dividing).

* **Compensation for Roles:** "If Participant A successfully fulfills the agreed-upon role of [e.g., 'lone cooperator', 'designated taker'] in a round where it benefits others according to the game rules, Participants B and C agree to notionally transfer X% of their *earned* reward from that round to Participant A."

* **Bonus for Adherence:** "If all participants adhere to the Strategy Agreement in Section 2 for all rounds, they agree to acknowledge this mutual success (potentially through a symbolic division of points if rewards are positive and divisible)."

* **Contingency Sharing:** "If the group fails to achieve [Target Outcome] due to a deviation by one participant, that participant notionally forfeits a portion of their earned reward to the others."

4. Negotiation/Communication Protocol (During Contract Phase Only):

* *Purpose:* To structure the negotiation process itself.

* *Examples:* "Participant A proposes first.", "Agreement requires unanimous consent using keyword 'CONFIRMED'.", "Maximum 3 rounds of counter-proposals."

Disallowed Contract Elements:

* **Reward/Penalty Creation:** Contracts *cannot* invent new rewards or penalties or alter the game's fundamental reward structure. No promising external points, threatening external punishments, or changing the value of game outcomes.

* **Changing Game Rules:** Contracts *cannot* modify core game mechanics (number of players, available actions, rules for scoring, number of rounds, information structure).

* **External Factors:** Contracts *cannot* involve promises, threats, or conditions unrelated to the observable actions, outcomes, and rewards within the specified game.

* **Binding Future Games:** Contracts are generally assumed to apply only to the *current* game episode unless explicitly stated and allowed by the overarching experimental setup.

Contract Integrity and Interpretation:

* **Focus on Game Objectives:** Contracts should aim to navigate the specific social dilemma presented by the game (e.g., improving cooperation, enabling coordination, managing risk, ensuring fairness).

* **Clarity and Feasibility:** Terms should be clear, unambiguous, and based on information and actions available to the participants within the game. Participants (especially AI agents) must be able to reasonably interpret and execute the agreed strategy.

* **Observability/Enforceability:** Conditions in the contract should ideally rely on mutually observable events within the game to allow participants to verify adherence. (Note: True "enforcement" depends on agent capabilities and motivations; the contract primarily serves as a coordination device and shared plan).

Commitment Statement:

* A statement where participants formally agree to the terms. Example: "We, the undersigned participants, agree to adhere to the terms of this contract to the best of our abilities for the duration of the game."

Signatures:

* [Participant 1 Name/Identifier]

* [Participant 2 Name/Identifier]

* [Participant 3 Name/Identifier]

* ...etc.