Large Language Models as Agents in Two-Player Games

Yang Liu[*]   Peng Sun[*]   Hang Li

ByteDance Research
{yang.liu01, wanhesong, lihang.lh}@bytedance.com

Abstract

By formally defining the training processes of large language models (LLMs), which usually encompasses pre-training, supervised fine-tuning, and reinforcement learning with human feedback, within a single and unified machine learning paradigm, we can glean pivotal insights for advancing LLM technologies. This position paper delineates the parallels between the training methods of LLMs and the strategies employed for the development of agents in two-player games, as studied in game theory, reinforcement learning, and multi-agent systems. We propose a reconceptualization of LLM learning processes in terms of agent learning in language-based games. This framework unveils innovative perspectives on the successes and challenges in LLM development, offering a fresh understanding of addressing alignment issues among other strategic considerations. Furthermore, our two-player game approach sheds light on novel data preparation and machine learning techniques for training LLMs.

1. Introduction

Large language models (LLMs) (Radford et al., 2019; Brown et al., 2020b; OpenAI, 2023; Team, 2023; Touvron et al., 2023) have emerged as powerful tools for building human-level intelligence. It is crucial to understand the underlying principles and mechanisms in a more scientific way, for future development of the technologies and beyond. One question we want to address is whether it is possible to formalize the typical training and inference processes of LLMs in a single and unified framework, including pre-training, self-supervised fine-tuning (SFT) (Ziegler et al., 2019; Dodge et al., 2020), reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022a), in-context learning (Min et al., 2022; Lampinen et al., 2022), and chain-of-thought

[*]Equal contribution

reasoning (Zhou et al., 2022a; Kojima et al., 2022; Wei et al., 2022; Wang et al., 2022a). For example, pre-training is inherently language modeling, whereas RLHF is grounded in reinforcement learning, belonging to separate paradigms within machine learning.

In this position paper, we propose an agent perspective to unify the methodologies for training and improving LLMs. Our inspiration partly stems from a straightforward observation: LLMs like ChatGPT typically behave like an agent when responding to human users' requests, such as answering questions, providing advice, solving math problems, engaging in creative writing, or assisting with task completion. The interactions between users and LLMs bear strong resemblance to a two-player game where player one (corresponding to the human users) and player two (corresponding to the LLMs) alternate in taking actions, with each attempting to maximize their individual internal goal. Indeed, this is a well-established subject that has been extensively studied in game theory (GT), reinforcement learning (RL), and multi-agent systems (MAS) with regards to creating agents that learn to play games. One can refer to, e.g., the studies of learning-in-game in traditional GT literature (Fudenberg et al., 1998; Cressman, 2003), as well as the contemporary work that employs deep RL as the learning tools in multi-player games (Heinrich, 2017; Brown et al., 2020a; Schmid, 2021), which is essentially multi-agent reinforcement learning (MARL) (Shoham & Leyton-Brown, 2008; Lanctot et al., 2017).

In our setting, there are two agents or players who interact with each other by playing a language-based game over multiple turns. In each turn of the interaction, one player generates a sequence of tokens in multiple steps, based on the interactions between the two players up to that point. More concretely, we consider an RL formulation. Suppose that *time* t represents the step of generation made by player-one (user) or player-two (LLM); *action* a_t corresponds to generating the next token e_{t+1}; *state* s_t corresponds to the state of the interactions so far, which can simply be the sequence of tokens generated up to step t: $s_t = \{e_i\}_{i \leq t}$; the *transition function* is defined to represent the transition

	Reinforcement Learning	LLM or Sequence Modeling
Step	Time t	Token t
State	s_t	$\{e_i\}_{i \leq t}$
Action	a_t	e_{t+1}
Reward	r_t	$\log \Pr(e_{t+1}|e_{i \leq t})$

Figure 1. LLMs can be viewed as agents participating in language-based games in the framework of reinforcement learning.

to state s_{t+1} given state s_t and action a_t; *reward* r_t corresponds to the loss incurred at generation of token e_{t+1}. Player-one and player-two each have their own policies for taking actions or generating token sequences to maximize their expected cumulative rewards or expected returns. Player one and player two can each be formalized as an RL agent. We summarize this formulation in Figure 1.

In our framework, we conceptualize pre-training from a vast text corpus as behavior cloning of a sub-optimal policy of an "average" player-two from a large amount of log data of two-player language-based games (after proper processing). We regard SFT and RLHF as the respective methods for behavior cloning and policy learning aimed at developing an optimal policy for player-two. Within our framework, we offer explanations for various phenomena, including the learning of multiple tasks, chain-of-thought reasoning, prompting, hallucination, and in-context learning. We highlight that this new framework enables a comprehensive explanation and analysis of critical issues concerning the alignment of LLMs, such as vulnerabilities to attacks, leveraging the capabilities of LLMs, and the pursuit of superhuman intelligence. Additionally, our approach illuminates potential strategies for augmenting the capabilities of LLMs (player-two) through the refinement of data preparation and the advancement of learning methodologies.

Section 3 defines the terms and notations necessary for our discussions. Section 4 elaborates on how each training process of an LLM can be interpreted through the lens of a two-player game. Section 5 explores the implications arising from our framework established in Section 4. The same section also delves into new opportunities, challenges, and open questions. Finally, Section 6 concludes our paper.

2. Related Work

Most relevant to us is the growing family of GPT models that have built the foundations of LLMs (Radford et al., 2019; Brown et al., 2020b; OpenAI, 2023; Team, 2023; Touvron et al., 2023). Thanks to advancements in technologies such as Transformer (Vaswani et al., 2017) and language model training, the abundance of data, and the large scale of

models, the current LLMs, with GPT-4 being a prominent example, have truly showcased human-level intelligence in numerous tasks (Bubeck et al., 2023). There has been tremendous interest in understanding the training mechanisms. Our position paper aims to spur further discussions on the research on LLMs.

SFT and RLHF (Christiano et al., 2017; Ziegler et al., 2019; Dodge et al., 2020; Liu et al., 2022; Ouyang et al., 2022; Bai et al., 2022a) are two prevalent training methods that steer LLMs to better align with human instructions or to generate outputs more in tune with human preferences. It is straightforward to consider the fine-tuning stage as behavior cloning and RLHF as reinforcement learning training. However, there has been no discussion about viewing pre-training as behavior cloning of a sub-optimal policy, to the best of our knowledge. Our discussions in this paper may provide profound insights into alignment techniques. For example, recent studies have investigated the possibility of self-aligning LLMs with guidance from another AI agent (see (Lee et al., 2023; Guo et al., 2024)). We are poised to offer novel guidelines for advancing these technologies.

Relevant to our scope are also methodologies that enhance LLMs through improved prompting, which include a broader discussion of prompt engineering (Arora et al., 2022; Chung et al., 2022; White et al., 2023), chain-of-thought (CoT) (Wei et al., 2022; Wang et al., 2022a; Zhou et al., 2023; Kojima et al., 2023), and in-context learning (Xie et al., 2021; Min et al., 2021; 2022; Lampinen et al., 2022; Dong et al., 2022), among others. As we will demonstrate, our formulation offers a novel perspective on these techniques.

Our study is closely related to the GT, RL, and MAS literature on building agents to play multi-step games. In this body of work, the goal is to design learning algorithms that enable agents to achieve optimal policies, or, equivalently, to implement iterative equilibrium-finding procedures. The type of game can be either perfect information game (Tesauro, 1995; Baxter et al., 2000; Silver et al., 2017b), or imperfect information game, which may arise from simultaneous actions (Littman, 1994; Hu & Wellman, 2003) or partial observability (Vinyals et al., 2019; Brown et al., 2020a). In this study, we consider the interactions between humans and LLMs as games with action patterns similar to strategy card games (Kowalski & Miernik, 2023), where efficient learning algorithms exist (Xu, 2016; Grill et al., 2020; Xi et al., 2023).

It has been discussed in the RL literature on its connection to the sequence modeling (Wen et al., 2022). For instance, a decision Transformer model is built for RL tasks (Chen et al., 2021). Offline RL algorithms using sequence modeling techniques are proposed by (Janner et al., 2021). Our paper provides a complementary view to this connection by
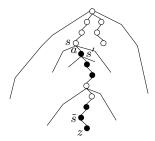
Figure 2. The Game Tree representation for the LLM formulation. Hollow circles: player one states, solid circles: player two states. On the trajectory (episode) ending at terminal state z, the representative states s, s', $\tilde{s}$ and action a are depicted, where $s \sqsubset \tilde{s}$, $(s, a) \sqsubset \tilde{s}$, etc. The active players are denoted as $P(s) = 1$, $P(s') = 2$, $P(\tilde{s}) = 2$. The edge (s, a) leads to s' according to the transition function $s' = T(s, a)$. The visiting probability $d^\pi(\tilde{s}|s, a)$, starting from the edge (s, a) and reaching the node $\tilde{s}$, is given by equation 2. Similarly, $d^\pi(z)$ is given by equation 1.

establishing how the relevant RL literature on two-player games can offer new insights to the training of LLMs.

3. Formulation

We now go through the preliminaries for a new framework explaining LLMs based on game theory, RL, and MAS.

Extensive Form Game and Game Tree We model the language interactions between a human and an LLM (agent) as an Extensive Form Game, where the two players interact in multi-turn and multi-step. For ease of discussion, we focus on the perfect information game setting. Later we will discuss the extension to the partially observable setting. Formally, let the game be $G = \langle \mathcal{N}, \mathcal{S}, \mathcal{Z}, \mathcal{A}, P, T, r \rangle$, which can be represented with a *Game Tree* as illustrated in Figure 2. Denote by $\mathcal{S}$ the set of all possible states and state $s \in \mathcal{S}$ corresponds to a node. The path from the root node to a node forms a trajectory. A node and the trajectory to it are used interchangeably when the context is evident. Denote by $\mathcal{Z}$ the set of *terminal states* and terminal state $z \in \mathcal{Z}$ corresponds to a leaf node. The action $a \in \mathcal{A}(s)$, representing the generation of a token, is taken at s, where $\mathcal{A}(s)$ denotes the vocabulary that is usually in the size of tens of thousands. Thus, (s, a) corresponds to an edge. The *transition function* T can be either probabilistic or deterministic. To model sequential token generation we adopt a deterministic function: $s' = T(s, a) = [s, a]$. Applying it recursively, we can obtain a sequence of tokens and we denote a state s using the sequence generated so far. Let $\mathcal{N} = \{1, 2\}$ denote the set of indices for the two players. A player $i \in \mathcal{N}$ can be either a human or an LLM. At each s, the *player function* $P(s) = i$ decides the *active player*, i.e., it is player i's turn to generate tokens. There exists a special token `<eos>` to indicate the ending of generation for player

i in this turn. Subsequently, it becomes the other player's turn for token generation, and the process continues. The multiple turns form an episode of a game, which is also a session of conversation. To indicate that a node s or an edge (s, a) belongs to the trajectory $\tilde{s}$, we use the notation $s \sqsubset \tilde{s}$ or $(s, a) \sqsubset \tilde{s}$. Finally, denote by $\text{ch}(s)$ the set of trajectories originating from node s, and denote by $\text{ch}(s, a)$ the set of trajectories originating from edge (s, a).

Policy and Visiting Probability At state s, the active player takes an action (generating a token) based on a conditional probability known as the *policy*, denoted as $\pi^i(\cdot|s) \in \Delta(\mathcal{A}(s))$, where the superscript i denotes the active player $i = P(s)$ and $\Delta(\cdot)$ represents the probability distribution defined on the action set. The policy $\pi^i(\cdot|s)$ can be either *deterministic* or *probabilistic*, also referred to as *pure* policy and *mixed* policy in game theory, respectively. Let i represent one player, and $-i$ represent the other player. We refer to $\pi = (\pi^i, \pi^{-i})$ as a *policy profile*. The policy $\pi^i(\cdot|\cdot)$ for player i can be either predefined or learned. Once the policy profile is determined, the traversal of a game tree is given. For a trajectory $s \in \mathcal{S}$ or an episode $z \in \mathcal{Z}$, the *Visiting Probability* at node s or leaf z, denoted by $d^\pi(s)$ or $d^\pi(z)$, is simply defined to be the product of the players' policies along the trajectory:

$$d^\pi(s) = \prod_{\forall(s', a') \sqsubset s} \pi^j(a'|s'), \quad d^\pi(z) = \prod_{\forall(s', a') \sqsubset z} \pi^j(a'|s'),$$
(1)

where the sum is over all the visited edges (s', a') and the active player is determined on-the-fly by those visited states $j = P(s')$. Reloading the notation $d^\pi(\cdot)$, we can also define the visiting probability for an edge (s, a) as $d^\pi(s, a) \equiv d^\pi(s)\pi^j(a|s)$, where $j = P(s)$. Furthermore, we define the conditional visiting probability which gives the probability of traversing $\tilde{s}$ starting from node $s \sqsubset \tilde{s}$ or edge $(s, a) \sqsubset \tilde{s}$:

$$d^\pi(\tilde{s}|s) \equiv d^\pi(\tilde{s})/d^\pi(s), \quad d^\pi(\tilde{s}|s, a) \equiv d^\pi(\tilde{s})/d^\pi(s, a).$$
(2)

Note that in general the visiting probability $d^\pi(\cdot)$ depends on the policy profile $\pi = (\pi^i, \pi^{-i})$ from both players.

Reward, Return and Value Function Let player i receive an *immediate reward* $r^i(s, a) \in \mathbb{R}$ at edge (s, a). Note that each player $i \in \mathcal{N}$ has their own reward function $r^i(s, a)$, no matter whether state s is i's turn or not. For a trajectory $z \in \mathcal{Z}$ and a player $i \in \mathcal{N}$, we define the *return* as his sum-of-rewards: $R^i(z) = \sum_{\forall(s, a) \sqsubset z} r^i(s, a)$. In a slight abuse of notations, we also respectively define the *expected return* and the *state-action value function* as:

$$R^i(\pi) = \mathbb{E}_{z \sim d^\pi(\cdot)}\left[R^i(z)\right] = \sum_{z \in \mathcal{Z}} d^\pi(z)R^i(z). \quad (3)$$
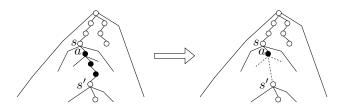
Figure 3. A fixed player $-i$ policy induces an MDP to player i, where the environmental dynamic is $\Pr(s'|s, a) = d^\pi(s'|s, a)$ by using the visiting probability equation 2 and by noting that the underlying transition $T(\cdot, \cdot)$ is deterministic.

$$Q_\pi^i(s, a) = \mathbb{E}_{z \sim d^\pi(\cdot|s,a)} \left[R^i(z) \right]$$
$$= \sum_{z \in \mathcal{Z}, z \in \text{ch}(s,a)} d^\pi(z|s, a) R^i(z). \qquad (4)$$

It is worth mentioning that the expected return or value function of player i also depends on the policy of the other player, i.e., the policy profile $\pi = (\pi^i, \pi^{-i})$.

The relationship between the returns $R^i(z)$ of the two players determines the type of the game. In a two-player game, $R^1(z) + R^2(z) = 0$ represents a zero-sum game, which is purely adversarial; $R^1(z) - R^2(z) = 0$ indicates an identical interest game, which is purely cooperative; $R^1(z) + R^2(z) = c(z)$ represents a general case, which is mixed competitive-cooperative (collaborating to make the cake bigger but conflicting when dividing the cake).

Reinforcement Learning Suppose that we aim to learn the policy π^i for player i while the other player's policy π^{-i} is fixed. In this scenario, a *Markov Decision Process (MDP)* is induced from the perspective of player i, where the *environment* is determined by the other player's policy π^{-i} and the transition function $T(\cdot, \cdot)$, as illustrated by Figure 3. The objective for player i is to maximize

$$\max_{\pi^i} R^i(\pi) \equiv \max_{\pi^i} R^i(\pi^i, \pi^{-i}), \qquad (5)$$

which corresponds to a (single-agent) *Reinforcement Learning* problem for player i.

Solution Concept and Nash Equilibrium When both players' policies are subject to learning, the optimal policy profile $\pi^* = (\pi^{i,*}, \pi^{-i,*})$ is referred to as a *Solution Concept*. One widely used solution concept in game theory is the *Nash Equilibrium* (NE), which represents a *fixed point* in the policy space that satisfies the following condition:

$$R^i(\pi^{i,*}, \pi^{-i,*}) \geq R^i(\pi^i, \pi^{-i,*}), \quad \forall \pi^i, \ \forall i \in \mathcal{N}. \qquad (6)$$

This condition indicates that if both players adopt NE policies, neither player can achieve a higher expected return by unilaterally altering his policy.

Figure 4. Example of token sequence in pre-training data. It can be interpreted as a series of state-action pairs resulting from the game of two RL agents.

4. Interpretations

In this section, we interpret various LLM technologies using the theoretic framework in the previous section.

4.1. Overview

Suppose that in the discussions in section 4.2-4.4, the player-one (*human*)'s policy is fixed such that it reduces to a single-agent RL problem for player-two (*LLM*). We sometimes rewrite the notations by omitting the player index superscript $i = 2$ for simplicity. Denote by $\pi_\theta(a_t|s_t) \equiv \pi^i(a_t|s_t)$ the policy over state $P(s_t) = i$ with parameter θ, denote by $J(\theta) = R^i(\pi) \equiv R^i(\pi_\theta, \pi^{-i})$ the objective function in equation 3, and denote by $Q_{\pi_\theta}(s_t, a_t) \equiv Q_\pi^i(s_t, a_t) \equiv Q_{(\pi^i, \pi^{-i})}^i(s_t, a_t)$ the state-action value in equation 4.

4.2. Pre-training

We regard pre-training of LLM as *behavior cloning* of the average player-two's policy, which is sub-optimal when aiming to construct an ideal player-two. Nonetheless, the data in the form of documents, approximating an extensive collection of logs from two-player games, can be highly valuable for our purpose, primarily due to its vast scale. After pre-training, the LLM becomes a Transformer-based policy model capable of taking actions or generating token sequences for player-two in two-player games.

The data in pre-training is not explicitly formatted to depict the sequential actions of two players. We posit that the inclusion of specific delimiters, akin to tags in HTML documents, could serve as cues to segment a token sequence into two parts. The first part would represent the actions of the first player, who poses a question, while the second part would represent the actions of the second player, who provides a response. The utility of delimiters within pre-training data is pointed out in previous work (Brown et al., 2020b).

Consider a document comprised of a token sequence as depicted in Figure 4, which is divided into two segments: a question and an answer. The corresponding sequence of states and actions for the document can be delineated as shown in Figure 4. Note that there are a total of $T = 16$ actions and $T + 1 = 17$ states.

$s_1 = [\,], \quad a_1 = \text{Who}, \quad P(s_1) = 1$
$s_2 = [\text{Who}], \quad a_2 = \text{is}, \quad P(s_2) = 1$
...
$s_6 = [\text{Who is Harry Potter?}], \quad a_6 = \text{<eos>}, \quad P(s_6) = 1$
$s_7 = [\text{Who is Harry Potter?<eos>}], \quad a_7 = \text{Harry}, \quad P(s_7) = 2$
$s_8 = [\text{Who is Harry Potter?<eos>Harry}], \quad a_8 = \text{Potter}, \quad P(s_8) = 2$
...
$s_{19} = [\text{Who is Harry Potter? ... of novels}], \quad a_{19} = \text{<eos>}, \quad P(s_{19}) = 2$
$s_{20} = [\text{Who is Harry Potter?<eos>Harry Potter is ... of novels <eos>}], \quad a_{20} = \text{Who}, \quad P(s_{20}) = 1$
$s_{21} = [\text{Who is Harry Potter?<eos>Harry Potter is ... of novels <eos>Who}], \quad a_{21} = \text{wrote}, \quad P(s_{21}) = 1$
...

Who is Harry Potter ? <eos>
Harry Potter is a fictional character in a series of novels .<eos>
Who wrote...

Figure 5. Example of token sequence in SFT data. It can be interpreted as a series of state-action pairs resulting from the game of two RL agents.

Specifically, the LLM is trained on a very large corpus of documents $\mathcal{D}$, to predict the next token in an auto-regressive manner by optimizing the log-likelihood (equivalent to minimizing the negative log-likelihood):

$$\hat{\pi} = \arg\max_{\pi} \tilde{\mathbb{E}}_{(s_t,a_t)\sim\mathcal{D}} \left[\log \pi(a_t|s_t)\right] \qquad (7)$$

where $\tilde{\mathbb{E}}$ denotes batch-level averaging. When calculating the pre-training loss for the data depicted in Figure 4, it is necessary to include all pairs of states and actions (s_t, a_t) for each t ranging from 1 to 16.

4.3. SFT

We perceive supervised fine-tuning (SFT) (Ziegler et al., 2019; Dodge et al., 2020) of LLM as behavior cloning of the optimal policy for player-two. Training data is treated as logged interactions between the two players, with player-two being the agent we aim to construct. The questions (data) are interpreted as actions (token sequences) from player-one, while the answers are interpreted as actions (token sequences) from player-two. The goal is for the LLM to learn from the demonstrations of the ideal player-one, which are typically annotated by human experts. The model is trained by maximizing the log-likelihood of the data $\mathcal{D}$

$$\pi^* = \arg\max_{\pi} \tilde{\mathbb{E}}_{(s_t,a_t)\sim\mathcal{D}} \left[\log \pi(a_t|s_t)\right], \qquad (8)$$

where $\tilde{\mathbb{E}}$ stands for batch-level averaging.

Suppose that we are provided with an example of SFT data, as in Figure 5. The states, actions and player functions can be then specified as in the figure. This figure includes sequences of tokens from both player-one and player-two. During player-two's turn, state s_7 serves as the starting point, and state s_{19} marks the end. Player-two engages in a series of actions, which leads to a progression of state and action pairs (s_t, a_t) for $t = 7, 8, ..., 18, 19$. The token generation process adheres to equation 8. The action sequence $[a_1, ..., a_6]$ corresponds to the question, and the subsequent action sequence $[a_7, ..., a_{19}]$ corresponds to the answer in the SFT data.

4.4. RLHF

In RLHF, the LLM undergoes further fine-tuning in a two-stage process (Christiano et al., 2017; Bai et al., 2022b; Ouyang et al., 2022; Lee et al., 2023). A reward model, $r^i(s, a)$ or $R^i(z)$, is first trained based on data from human preferences. The LLM after SFT is then further trained using policy gradient based RL, e.g., the PPO algorithm (Schulman et al., 2017). RLHF is considered as a single agent RL process aimed at enhancing the LLM's capability to take actions or generate tokens, effectively emulating an ideal player-two.

The objective of RL is to maximize the expected return as in equation 5. This can be done by gradient ascent (Sutton et al., 1999) that explicitly gives the gradient:

$$\nabla_\theta J(\theta) \approx \tilde{\mathbb{E}}_{(s_t,a_t)\sim\mathcal{D}} \left[Q_{\pi_\theta}(s_t, a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)\right], \quad (9)$$

where $\mathcal{D}$ is a "temporary dataset" consisting of the fresh rollout data for player two sampled by the current policy profile (π_θ, π^{-i}), i.e., the on-policy RL. The PPO algorithm is similar in essence to equation 9, but incorporates more advanced safe stepping techniques and variance reduction techniques.

4.5. Additional Settings

We explore several other setups of LLMs as two-player games.

Meta Learning In all three-stages of LLM training, the dataset $\mathcal{D}$ in fact contains a large number of subsets that feature different tasks and different transition dynamics ($\mathcal{D}_i, i = 1, 2, \cdots, n$). Therefore, the formulations in sections 4.1-4.3 can also be viewed as a *meta policy learning* process (Nagabandi et al., 2018; Gupta et al., 2018; Rakelly et al., 2019), in which player-two learns a meta policy from the entire dataset of different tasks.

During the three stages of LLM training, the dataset $\mathcal{D}$ actually comprises a multitude of subsets, representing different tasks with their different RL configurations ($D_i, i = 1, 2, \cdots, n$):

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 .... \cup \mathcal{D}_n$$

Consequently, the approaches outlined in sections 4.1-4.3 can be interpreted as a meta policy learning process, in which player-two acquires a meta policy that is applicable across a large number of tasks represented by the entire dataset:

$$\pi^* = \arg\max_{\pi} \tilde{\mathbb{E}}_{G\in\mathcal{D}} \left[\tilde{\mathbb{E}}_{(s_t,a_t)\sim\mathcal{D}(G)} \left[\log \pi(a_t|s_t)\right]\right] \qquad (10)$$

In the notation previously introduced, a game formulation G_i is first selected from a distribution of games, and then a set of episodes is drawn from the episode distribution $\mathcal{D}_i$

of the game G_i. This analogy further clarifies the often-noted zero-shot or few-shot learning capabilities exhibited by LLMs such as GPT-3 and others.

Partially Observed States A natural extension of our framework is to conceptualize the games between two players such that the states of environment are only partially observed. From the perspective of either player-one or player-two, the scenario resembles a partially observed Markov decision process (POMDP). In a competitive setting, player-one might deliberately conceal information during its engagement with player-two. Even in a collaborative or mixed scenario, it is possible that player-one does not adequately communicate all the essential details (Andreas, 2022). Under these circumstances, player-two must deduce the true states of the environment.

The POMDP extension helps us better comprehend the significance of the quality of prompts, i.e., the questions or extra elucidations provided by player-one. A more effectively articulated prompt diminishes the uncertainty and noise within the token sequence from player-one, enabling player-two to precisely discern the states and execute appropriate actions (Zhou et al., 2022b; White et al., 2023).

The extension also enables us to understand the efficacy of chain-of-thought (CoT) reasoning (Wei et al., 2022; Wang et al., 2022a). In this approach, player-two produces a token sequence that reflects its "thought process", that is, the CoT, prior to generating the token sequence that serves as a response to player-one. Player-two can utilize the observation to deduce the states while deciding on the subsequent actions. The information provided by the CoT can offer a clearer indication of the true states and lead to better actions.

The POMDP perspective additionally compels us to contemplate a more comprehensive formulation for the training of LLMs, which we regard as future work. This opens up new prospects for constructing world models for both player-two and player-one, thereby enhancing their capacity for reasoning and planning (Ha & Schmidhuber, 2018; Hao et al., 2023). However, to fully leverage the players' ability to create and use their world models, it would necessitate capabilities for sensing and gathering data across multiple modalities, encompassing image, video, speech, and others.

Hallucination The two-player games in question are "language-based games", where the players must engage using human language. This implies that the token sequences produced by both players are entirely legitimate in human language. Put differently, these token sequences are required to adhere to the syntactic, semantic, and pragmatic norms of the language, which concurrently constitute the rules of the games. An underlying premise of this two-player game formulation is that the rules are not explicitly provided; instead, they must be inferred from the data obtained in the

past games by many different players.

The process of learning, especially pre-training of LLMs, can largely fulfill the goal of discerning the rules of the game from vast amount of text data. It seems that the token sequences generated by LLMs are syntactically and pragmatically impeccable. Nevertheless, the same cannot be said for their semantic accuracy. At times, LLMs may produce token sequences that seem credible yet lack factual substantiation - a phenomenon known as *hallucination* (Zhang et al., 2023b; Li et al., 2023; Ji et al., 2023; Liu et al., 2023). It is evident that the learning mechanisms of LLMs are not fully equipped to address the issue of hallucination effectively.

We posit that the primary cause of hallucination in LLMs is their lack of construction and connection to world models (LeCun, 2022). Although LLMs can generate responses to fact-based questions, showing that they store knowledge in the form of language, they lack world models that could help verify whether their statements are factually correct. Consequently, this leads to the breach of semantic rules pertaining to the veracity of the statements made.

In-Context Learning As outlined in the GPT-3 paper (Brown et al., 2020b), a substantial volume of question-answer pairs can be assembled, annotated, and incorporated into the pre-training of an LLM through what is called in-context learning (Min et al., 2021; Dong et al., 2022). This approach can be regarded as an imitation of average player-two within our framework. Meanwhile, the learning is similar to off-policy reinforcement learning and is conducted in a mini-batch fashion. With this kind of training, an LLM is capable of executing zero-shot or few-shot learning effectively, without the need of additional parameter tuning. During inference, the LLM can be immediately utilized in an in-context learning scenario to enhance the capability of player-two (Zhu et al., 2023a), where a handful of question-answer examples are given as a prompt.

4.6. Alignment

To improve the capabilities of LLMs (player-two), it is advantageous to also consider the capabilities of the environment (player-one) and the overall game configuration. This will foster the creation of novel alignment algorithms that go beyond the reinforcement learning from human feedback (RLHF). We can explore various two-player game settings as outlined below, which lead to solutions for a range of complex alignment challenges. These include adversarial attacks on LLMs, harnessing the full potential of LLMs, and advancing LLMs towards superhuman intelligence, corresponding with adversarial player, cooperative player, and the Nash equilibrium, respectively.

The alignment research is poised for significant advance-

ment through the development of new algorithms within our framework, moving beyond the current PPO algorithm and the recently introduced direct preference optimization (DPO) algorithm (Rafailov et al., 2023). We also believe that the large collection of learning algorithms developed for games, including those tailored for specific games (Guo et al., 2021), as well as those created for imperfect games (Kahn et al., 2017; Wang et al., 2020; 2021; Perolat et al., 2021; An et al., 2021; Sokota et al., 2022; Zhang et al., 2023a), can also inspire the design of new player-two algorithms that could significantly enhance the alignment of LLMs (Wang et al., 2022b; Sun et al., 2023; Guo et al., 2024).

4.7. Adversarial Players

Enhancing the trustworthiness of LLMs is crucially important in the development of LLMs, including alignment. From the viewpoint of a two-player game, all the issues arise due to an adversarial player-one. These include but are not limited to reliability, safety, robustness, fairness, adherence to social norms, and resistance to misuse (Liu et al., 2023; Wang et al., 2023). The key question here is how to model player-one. One possible approach is to train another LLM as player-one, and let player-one and player-two play against each other. In learning, they can enhance their capabilities through a large number of game-plays, analogous to the self-play RL in AlphaZero (Silver et al., 2017b).

Recent research on "red teaming" (Perez et al., 2022; Ganguli et al., 2022) is based on this presumption. Within our game-theoretic framework, player-one and player-two are participants in a zero-sum game $R^1(z) + R^2(z) = 0$, indicating a purely competitive interaction (cf., section 3). With adequate training, player-one becomes adept at creating a range of challenging questions designed to stump player-two, while player-two simultaneously learns to respond to these questions safely and effectively. Their skills, or their policy profile, improve over time, ultimately converging to a Nash equilibrium as in equation 6.

4.8. Cooperative Players

The majority of interactions between an LLM and human users can be seen as cooperative games involving player-two and player-one (Nash, 1953; Claus & Boutilier, 1998). This encompasses interactions like responding to queries, offering guidance, providing emotional health assistance, entertaining, and helping with various tasks by the LLM.

A critical technique in employing LLMs involves the art of prompting. It is broadly recognized that effective prompting can remarkably improve the quality of responses by LLMs (player-two). This can be viewed as augmenting player-one's questions with additional explanations that act as context, enabling player-two to respond more appropri-

ately. The reason can be interpreted as that without proper prompting, the intrinsic states of player-one are not fully exposed in its generated tokens.

An important question is how to enhance prompting (additional actions of player-one) via a learning mechanism. In (Saunders et al., 2022; Bai et al., 2022b), a predetermined policy is applied to player-one, whereas a policy for player-two is formulated through learning. It is also feasible to consider a two-player identical interest game $R^1(z) = R^2(z)$, as discussed in section 3. In such a cooperative scenario, the player-one policy should be carefully initialized and adjusted, serving as a dedicated prompt provider, and both the player-one and player-two policies are learnable, being refined over time and ultimately approaching NE as in equation 6.

4.9. Superhuman Intelligence

A recently defined objective within the community, including organizations like OpenAI, is to align LLMs for tasks that require superhuman intelligence. Currently, there are many open questions on the feasibility of accomplishing this formidable goal. OpenAI's newly published work (Burns et al., 2023) explores the concept of employing less powerful models to guide a much more stronger LLMs in executing tasks of superhuman complexity. Further investigation is essential for this challenging issue.

If we consider alignment involving two agents capable of self-play and simultaneous improvement, it appears to be a more natural approach to attaining superhuman intelligence. This concept is not novel; we have already seen AlphaZero's triumph over top human Go players through self-play between two AI agents.

If we consider both players as learning agents progressing towards an equilibrium strategy, we must ponder what this equilibrium would mean for an LLM. Would the equilibrium strategy lead to a superhuman level of policy? Exploring these questions will yield both theoretical and empirical insights for the creation of LLMs.

5. Insights and Open Questions

In this section, we offer several insights derived from our framework and pose several questions for future research.

5.1. Data Preparation

Our framework offers valuable insights for data preparation. For example, the game approach to the pre-training phase suggests that data pre-processing methods which convert unstructured text data into a "question-answer" (Q-A) structure reflecting the actions of two players can facilitate the training of the learning player-two (LLM), as indicated by

(Brown et al., 2020b).

Furthermore, examination of chain-of-thought (CoT) suggests that the pre-training data may already include a "question-reasoning-answer" (Q-C-A) structure from which the model can learn. It is possible that pre-trained LLMs have acquired the CoT capability, which can be effectively brought out through proper prompt engineering.

These two observations lead to the critical question of whether explicitly structuring data into Q-A or Q-C-A formats could enhance the training process even further. We leave this question to the research community.

Moreover, it is apparent that the success of SFT and RLHF is contingent on the policy of player-one. The prevalent method of data gathering for SFT and RLHF resembles the process of obtaining samples from the policy of an "average" player-one, which is essentially data sourced from typical users. We anticipate that our framework will motivate a re-evaluation of the dataset preparation for SFT and RLHF. This could involve a more deliberate modeling and training of player-one, aimed at producing higher quality data (Zhu et al., 2023b). This data would then be used by player two for learning of an optimal policy. The concept shares similarities with red teaming; however, we consider here a more principled approach for the collection of superior data.

5.2. Training Methods

Reward Functions The two-player game approach also provides fresh insights into reinforcement learning of LLMs, including RLHF. By substituting the reward function with one that represents the two-player games, we can expand the versatility of reinforcement learning. For example, by devising a reward function that embodies a zero-sum game, it is conceivable to train a model to learn to refuse to respond to particular queries. This area of study, recently gaining interest under the theme of "LLM unlearning" (Yao et al., 2023; Eldan & Russinovich, 2023), highlights the potential of reward function design in reinforcement learning.

Value Functions Incorporation of value functions into the training of LLMs and the users they interact with can also be explored. They could enhance the learning of policies for both LLMs and the users, corresponding to player-two and player-one respectively. Given our framework, investigating this topic is of significant importance.

Our framework also underscores the significance of formalizing and employing value functions for long-term planning. Although the learning of LLMs have inherently included the learning of language-based games within their training regime, they lack the notion of a long-term value function, which is a standard component in the training of reinforcement learning agents for long-term planning (Guo et al., 2014; Gupta et al., 2019; Schrittwieser et al., 2020). The

explicit computation of long-term value functions could aid the LLMs exploit "long chains of reasoning," thereby reducing the effect of hallucination and enhancing the models' reasoning and planning abilities.

Multi-Agents The recent surge in interest centers on collaborations among multiple LLMs, with the objective of enhancing the overall performance of the models. Research on LLM debates, for instance, has offered supporting evidence for this collaborative approach (Du et al., 2023; Chan et al., 2023). Such results highlight the prospective benefits of utilizing multi-agent technologies during the training of LLMs to boost their collective capabilities.

Learning from Scratch Recent research demonstrates that with a simple win-or-lose reward for terminal states, an agent can master the board games like Go, Chess, and Shogi. This success is attained a unified approach to multi-agent reinforcement learning (MARL) that learns from the scratch (Silver et al., 2017a;b). Consequently, one might naturally ask: given an omniscient reward function/model, can the players learn, *tabula rasa*, human-level language-based "game" abilities? It is noteworthy that in our formulation of a perfect information game for two-players, a communication channel with unlimited capacity among the players is readily feasible. Furthermore, the upper bound of the gaming capability is subject to the reward function/model. Is it possible to establish a reward function or model that facilitates the acquisition of human-level abilities?

Learning in World Environment It is widely held that natural language has evolved from interactions among humans (Pinker, 2007). Initial efforts to replicate this phenomenon have been made in controlled laboratory settings. It is stated in (Mordatch & Abbeel, 2018): "*...teach AI agents to create language by dropping them into a set of simple worlds, giving them the ability to communicate, and then giving them goals that can be best achieved by communicating with other agents*" (quoted from (Mordatch, 2017)).

The conceptualization of LLMs as players in a game, or more broadly, as agents, might allow us to expand the range of AI agents. This expansion could take us from agents that learn language alone to those that learn language within a multimodal world environment, and from agents that solely learn a language model to those that learn an integrated system of language and multimodal models (LeCun, 2022).

6. Concluding remarks

This position paper presents a two-player game framework to re-evaluate the training, tuning, and alignment processes of LLMs. We provide a detailed argument for viewing each training process of an LLM through the lens of a two-player game. Our formulation provides implications and insights that may explain the current successes of LLMs,

including the celebrated SFT, RLHF, and in-context learning paradigms. Our formulation suggests potential future developments, including possible ways to more effectively align an LLM to follow human preferences, and potential ways to improve reasoning by more explicit modeling of long-term values. We hope that our paper will inspire more discussions between the LLM and game theory, RL and MAS communities.

7. Impact Statements

This paper presents a two-player game formulation to explain some of the underlying mechanisms of LLMs. We believe our paper has the potential to unveil the inner workings behind the observed successes and failures of LLMs. The comprehensive understanding and insights derived from our paper could enhance the trustworthiness of LLMs. For example, we provide recommendations for comprehending the hallucination aspect of LLMs. Additionally, our paper offers suggestions for enhancing the capabilities of LLMs. For instance, our discussion on developing reward functions that embody different zero-sum games has the potential to inspire new alignment algorithms and applications.

References

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Andreas, J. Language models as agent models. *arXiv preprint arXiv:2212.01681*, 2022.

Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatia, K., Chami, I., Sala, F., and Ré, C. Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*, 2022.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Baxter, J., Tridgell, A., and Weaver, L. Learning to play chess using temporal differences. *Machine learning*, 40: 243–263, 2000.

Brown, N., Bakhtin, A., Lerer, A., and Gong, Q. Combining deep reinforcement learning and search for imperfect-information games. *Advances in Neural Information Processing Systems*, 33:17057–17069, 2020a.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020b.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., Chen, Y., Ecoffet, A., Joglekar, M., Leike, J., et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate, 2023.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022.

Claus, C. and Boutilier, C. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2, 1998.

Cressman, R. *Evolutionary dynamics and extensive form games*, volume 5. MIT Press, 2003.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate, 2023.

Eldan, R. and Russinovich, M. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

Fudenberg, D., Drew, F., Levine, D. K., and Levine, D. K. *The theory of learning in games*, volume 2. MIT press, 1998.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Grill, J.-B., Altché, F., Tang, Y., Hubert, T., Valko, M., Antonoglou, I., and Munos, R. Monte-carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*, pp. 3769–3778. PMLR, 2020.

Guo, H., Yao, Y., Shen, W., Wei, J., Zhang, X., Wang, Z., and Liu, Y. Human-instruction-free llm self-alignment with limited samples, 2024.

Guo, W., Wu, X., Huang, S., and Xing, X. Adversarial policy learning in two-player competitive games. In *International Conference on Machine Learning*, pp. 3910–3919. PMLR, 2021.

Guo, X., Singh, S., Lee, H., Lewis, R. L., and Wang, X. Deep learning for real-time atari game play using offline monte-carlo tree search planning. *Advances in neural information processing systems*, 27, 2014.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. *Advances in neural information processing systems*, 31, 2018.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning, 2019.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hao, S., Gu, Y., Ma, H., Hong, J. J., Wang, Z., Wang, D. Z., and Hu, Z. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Heinrich, J. *Reinforcement learning from self-play in imperfect-information games*. PhD thesis, UCL (University College London), 2017.

Hu, J. and Wellman, M. P. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., and Levine, S. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners, 2023.

Kowalski, J. and Miernik, R. Summarizing strategy card game ai competition. *arXiv preprint arXiv:2305.11814*, 2023.

Lampinen, A. K., Dasgupta, I., Chan, S. C., Matthewson, K., Tessler, M. H., Creswell, A., McClelland, J. L., Wang, J. X., and Hill, F. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*, 2022.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

LeCun, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. 2022.

Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., and Rastogi, A. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Li, J., Cheng, X., Zhao, W. X., Nie, J.-Y., and Wen, J.-R. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *EMNLP*, pp. 6449–6464, 2023.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965, 2022.

Liu, Y., Yao, Y., Ton, J.-F., Zhang, X., Cheng, R. G. H., Klochkov, Y., Taufiq, M. F., and Li, H. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2023.

Min, S., Lewis, M., Zettlemoyer, L., and Hajishirzi, H. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Mordatch, I. Learning to communicate, 2017. [Online; accessed 18-Jan-2024].

Mordatch, I. and Abbeel, P. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Nash, J. Two-person cooperative games. *Econometrica: Journal of the Econometric Society*, pp. 128–140, 1953.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*, 2022. URL https://arxiv.org/abs/2203.02155.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Perolat, J., Munos, R., Lespiau, J.-B., Omidshafiei, S., Rowland, M., Ortega, P., Burch, N., Anthony, T., Balduzzi, D., De Vylder, B., et al. From poincaré recurrence to convergence in imperfect information games: Finding

equilibrium via regularization. In *International Conference on Machine Learning*, pp. 8525–8535. PMLR, 2021.

Pinker, S. *The stuff of thought: Language as a window into human nature*. Penguin, 2007.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340. PMLR, 2019.

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., and Leike, J. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Schmid, M. *Search in imperfect information games*. PhD thesis, Charles University, 2021.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017b.

Sokota, S., D'Orazio, R., Kolter, J. Z., Loizou, N., Lanctot, M., Mitliagkas, I., Brown, N., and Kroer, C. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *The Eleventh International Conference on Learning Representations*, 2022.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y., and Gan, C. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*, 2023.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Team, G. Gemini: A family of highly capable multimodal models, 2023.

Tesauro, G. Td-gammon: A self-teaching backgammon program. In *Applications of neural networks*, pp. 267–285. Springer, 1995.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv preprint arXiv:2306.11698*, 2023.

Wang, J., Liu, Y., and Li, B. Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6202–6209, 2020.

Wang, J., Guo, H., Zhu, Z., and Liu, Y. Policy learning using weak supervision. *Advances in Neural Information Processing Systems*, 34:19960–19973, 2021.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022a.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wen, M., Kuba, J., Lin, R., Zhang, W., Wen, Y., Wang, J., and Yang, Y. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

Xi, W., Zhang, Y., Xiao, C., Huang, X., Deng, S., Liang, H., Chen, J., and Sun, P. Mastering strategy card game (legends of code and magic) via end-to-end policy and optimistic smooth fictitious play. *arXiv preprint arXiv:2303.04096*, 2023.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Xu, Z. Convergence of best-response dynamics in extensive-form games. *Journal of Economic Theory*, 162:21–54, 2016.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

Zhang, X., Chen, J., Wang, H., Xie, H., Liu, Y., Lui, J. C., and Li, H. Uncertainty-aware instance reweighting for off-policy learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023b.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022a.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., and Chi, E. Least-to-most prompting enables complex reasoning in large language models, 2023.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022b.

Zhu, Z., Lin, K., Jain, A. K., and Zhou, J. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023a.

Zhu, Z., Wang, J., Cheng, H., and Liu, Y. Unmasking and improving data credibility: A study with datasets for training harmless language models. *arXiv preprint arXiv:2311.11202*, 2023b.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.