Unravelling Complex Sequential Social Dilemmas: In a Risky World with A2C Decision Transformers

Ashish Panchal
Indian School of Business
Hyderabad, India
ashish_panchal@isb.edu

Vandith Pamuru
Indian School of Business
Hyderabad, India
vandith_pamuru@isb.edu

ABSTRACT

Real-world decision-making often involves complex sequential social dilemmas (SSD), characterized by semi-structured environments, dynamic interactions, and multiple agents with conflicting goals. Therefore AI must be agile and adaptive to be useful. According to recent research, existing RL algorithms, despite success in structured environments, struggle with SSD complexities found in games like Civilization and Diplomacy. Dynamic player populations and unbounded action sequences per turn lead to high non-stationarity in large state spaces, significant world changes between turns, and persistent uncertainty hinder effective strategy learning. As these simulated games are still too complex, we developed a miniature "Risk"-inspired environment to better understand and foster explainable AI in complex SSDs. This environment features various players, long action sequences with stochastic outcomes, and the potential for temporary alliances and deceit, mirroring real-world complexities and requiring agents to adapt to unpredictable situations. Consistent with prior work, traditional RL algorithms (A2C, DDQN) under-perform against simple pseudo-random rule-based bots in our environment. We posit that large gaps between agent turn and the vast state space necessitate understanding opponent intentions through action sequences rather than isolated snapshots. Furthermore, opponents' multiple, hidden sub-goals complicate transition probability estimation. Aligning with this hypothesis, we created an encoder-only Decision Transformer, leveraging A2C, which significantly outperforms traditional methods in this simulated setting, potentially highlighting the importance of temporal context in complex social interactions. Future work will explore the model's potential in dynamic relationship-building, including cooperation, collaboration, and deceit, in an environment with multi-modal interaction. This research aims to take initial steps toward highlighting the utility of RL-DT like models to bridge the gap between simplified environments and the complexities of real-world SSDS, which permeate diverse domains, from large-scale scenarios like international negotiations and market competition to smaller-scale examples such as medical treatment planning and team sports practice.

KEYWORDS

Sequential social dilemma, Multi-agent reinforcement learning, Decision Transformer

ACM Reference Format:
Ashish Panchal and Vandith Pamuru. 2025. Unravelling Complex Sequential Social Dilemmas: In a Risky World with A2C Decision Transformers. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Most systems in this universe can be defined as isolated environments following a set of rules. Multiple elements follow set rules inside these systems and interact with each other. By design, these elements might have goal/s that may or may not align with others. A dilemma in the system arises when the interests of most or all of the elements are at odds.

This type of dilemma occurs frequently and can be observed across a wide range of interactions, from a macro perspective, in international diplomacy (including economic interactions, development cooperation, military alignments, and environmental cooperation) to the micro context in firm-level interactions or even in a still smaller scope such as interaction of players in a sports team or individual treatment plans for chronic health conditions that involve multiple medications over time. Multi-agent reinforcement learning mirrors human interaction in society that has been used to successfully address parts of such larger problems [9, 25, 27, 32, 35], in most cases where the properties of the environment could be defined more precisely, such as bounded state and action spaces. More specifically, while research has shed some light on the emergence of cooperation in a competitive environment such as the tragedy of commons, though good at studying the behavior of said agents, traditional MARL has only seen limited success [37], where human beings thrive. Nonetheless, on a very simplified simulation, PPO-like algorithms have shown promise [39], i.e., agents can learn to sustain together in a predator-prey situation by developing collaborative strategies like herding. This was extended later by Leibo et al. [26] who introduced two SSDs, Harvest and Wolfpack hunting game, modeling agents that learned policies to implement their strategic intentions, i.e. cooperation and defection, under different environmental characteristics, governing the ratio between the two strategies, they shed some light on when and how can cooperation emerge in a competitive world. Hughes et al. [22] later focused on a specific property of the agent rather than the environment, such as inequity aversion, that promoted cooperation specifically in intertemporal social dilemmas. This was particularly interesting as the players had a state-action value function depicting a specific kind of moral value resembling human behavior, such as globally advantageous or disadvantageous aversion, resulting in temporary "all vs. one" team-up to resolve certain dilemmas and maintaining high values.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), A. El Fallah Seghrouchni, Y. Vorobeychik, S. Das, A. Nowe (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

Following this, Jaques et al. [24] worked on modeling agents with reward functions based on the social influence of their actions on other agents quantifying alignment with their intrinsic motivation, either by modeling their actions based on directly visible opponent's actions or via dedicated channel communication between agents, where they used A3C model for the agents, along with modeling other agents directly using LSTM policy network for each opponent agent. Where they showed influence-based rewards coupled with modeling opponents resulted in higher overall returns. Although they do not particularly quantify cooperation and temporal change in the agent's attitude. While all these works broadened our understanding of solving SSD using MARL, many of these follow conventional definitions of state action space and environmental dynamics of decentralized partially observable games with limited reduction abilities to real work problems. However, real-life systems and situations could be a little more asynchronous, i.e., while one player takes 10 actions, another player takes 1 due to reasons like different external constraints or long-term strategies.

Additionally real life situations are more dynamic, i.e. it might take many temporal observations to understand what an opponent is trying to do, that may change mid-way, therefore there is no fixed horizon which can be used to completely capture this picture. This results in a state-action space, which sometimes is impossible to completely estimate, and by extension, the transition and reward functions as well. For humans, these events are quite common, and as a species, we have shown resilience and identified solutions for the toughest of such problems. Board games like "Diplomacy" and "Civilization" have been known to capture combinatorial state action space complexities, which make them generalist and reducible to other real-life social systems. In some of the initial works, Philip et al. [36] proposed diplomacy as a new SSD test bed to model real-life problems, with a GCN-based "dipnet" model, with supervised pretraining on a large human data followed by A2C-based finetuning using self-play. The model did well against predefined bots and showcased potential for dynamic coalition formation. However, it still failed to show any skill improvement and, by extension, no incentive optimization. Anthony et al. [4] tried to address this problem by proposing a best response and fictitious play-based policy iteration BRPI algorithm utilizing a combination of GNN encoder and LSTM decoder, coupled with human action imitation similar to [36], winning 27.3% times against dipnet A2C where the latter only won 1.3% times against the former. However, they do not shed light on the cooperative nature of the model.

In their subsequent work, Jonathan et al. [21] showcased a one-step lookahead search agent (Searchbot) trained on human data along with regret minimization, outperforming BRPI and securing 2%ile rank against humans on an online platform. Bakhtin et al. [7] continued to develop DORA using double oracle to approximate Nash Equilibrium for RL-based action exploration. They not only beat Searchbot but also reached superhuman performance in 2 player diplomacy game without any human data pretraining,. However, they were the first to present that self-play-based models converge to drastically different equilibria than human players in a 7-player full setting, performing slightly worse than the previous SOA, indicating that self-play might be insufficient for superhuman performance. Briefly, they also showed using a Transformer model instead of a GCN in prior work [4, 21, 36] resulted in improved performance. Jecob et al., in their work, answered the gap of human alignment [23] by introducing a regret minimization algorithm regularized by the KL divergence from an imitation learning (IL) policy trained from human data. Searchbot was found to improve over the base IL score. Anton et al. [8] generalized this algorithm by replacing the fixed regularization parameter with a probability distribution over these parameters, resulting in the top 2 out of 3 players in a mixed human diplomacy competition of 48 players. Bakhtin et al. [17] focused on human alignment with natural language negotiation and coordination with players using LLM to reach human-level proficiency by modeling the likely actions of the players. Most of the later work emphasizes that AI from self-play cannot learn human-aligned strategies without human data. A similar research approach was followed in the game of civilizations, a strategy game where players lead civilizations from the Stone Age to the modern era, competing to build empires, explore the world, and interact with each other through various means with a run time of few hours to a couple of days.

In a more recent effort, Siyuan et al. [38] proposed a minigame and experiment with Mastaba - hierarchical organization of LLM agents. Using pre-trained GPT 3 models, each responsible for a subpart of the game governed by an advisor LLM, not limited to the players to just dialogues but also logically recommending actions, postulated the transferability of human alignment of these models to a game mirroring society. The model showed only moderate results compared to humans and failed to learn defensive strategies. While these results may cast doubts on the utility of the LLMs while using long-term in-context information effectively in the gameplay, since the models were not specifically trained for the environment, the actual capability of these models remains unclear.

Although an AI model must understand and align with human values, it is not clear why the policies learned by AI should be similar to those of humans to cooperate with them. As a decision support system, the only requirement AI has is to optimize the utility of actions in an environment interacting with other AI agents and human agents that may or may not be cooperative all the time, as is the case in human deliberations. However, it is evident from the mentioned research above that there is a certain alluring human element that is not present in the current Agent models, which might be crucial to solving the demanding challenges of our society. Additionally, the complexity of the environments above might be too high to learn from, evaluate, and understand the reasoning of actions in a social interaction.

Even though the initial investigations considered environmental and player's intrinsic characteristics as the driving factor for the identification of reward optimizing behaviour controlling the dynamics of cooperation and deceit, the research later focused more on algorithm optimization techniques and model architecture to directly optimize the overall returns, however, still failed. Larger models trained on large data for a prolonged time on large computing units have been shown to perform significantly well in learning tasks. Arguably, these might still be insufficient due to the lack of social learning as a human factor, that is, learning with society rather than learning in society.

Evolutionary and computational biologists (Cosmides et al. [14] and Michael et al. [31]) talk about two competitive theories of human intelligence with much support, Social exchange theory and the Cultural Brain Hypothesis. The former elicits the presence of human cognitive architecture incorporating specialized reasoning, shaped by natural selection to solve adaptive problems like social interaction and avoiding dangers, with an example of a cheater-detection system that is highly precise, activating only when a social exchange potentially involves cheating. The latter posits that the brain has evolved to acquire, store, and manage adaptive knowledge gained through social and individual learning. The analysis in the study shows that larger groups and inherited knowledge favor social learning. In contrast, scarce knowledge favors individual learning, explaining observed correlations between brain size, group size, and development.

Based on this evidence and arguments presented in the above paragraphs, three possible hypotheses can be formulated. Firstly, the current environments in which to study cooperations are either too complex, like the game of diplomacy and civilization, with many moving parts and extremely long game horizons, or are too simple to be generalized to real-life SSDs. Secondly, initial research in studying cooperation in simpler environments and evidence from the evolution of human intelligence warrants further exploration of algorithms and models that can make use of social learning inherently modeling social exchanges and having an adaptive policy with and without past human inferences where the latter would be useful in identifying potentially new and unseen strategies, even in risky environments with the cost of collecting human data is high. Lastly, following [14, 31] and based on the recent work [6, 41], the impact of complex and large social groups on the evolution of intelligence cannot be rejected. It can be hypothesized that humans learn complex social strategies from larger life experiences which can then be transferred to small-scale simulated games like diplomacy and civilization and, by extension, smaller problems. Therefore, it is imperative to validate if such skills can be developed in AI agents in large and densely populated learning environments and transferred to smaller environments.

Our efforts start with answering the 1st hypothesis. The turn-based board game of Risk [1], while restricting the diplomatic setting to world conquest, offers a complex environment with an unbounded sequence of actions, potential for dynamic relationship building, highly delayed rewards, and high non-stationarity [20]. This presents grounds for intragroup negotiation study [5, 29]. Even with simpler dynamics, MCTS and Expert iteration models like Alphazero with self-play, that defeated previous state-of-the-art and human champions in 2-player strategic games like Chess, go and shogi [40] which are found to be EXPTIME complete [2, 19, 40, 45, 48], did not produce adequate results in the game of risk in their initial explorations [11, 12, 33, 34, 44]. Inherently, games like risk and diplomacy involving unbounded actions and by extension unbounded number of negotiation deals result in the estimation of Nash equilibria and value estimation problem to be NP-hard [15, 16, 18] .There is reason to believe that the exponentially increasing branching factor, resulting in a huge state-action space, similar to diplomacy and civilization, inhibits the model from effectively learning complex strategies like the average human player. It is challenging, as the intention of the players is visible only after many iterations of action and turn cycles, which limits the analyzer's ability to validate strategies learned by the model. To facilitate analysis and visualization of the model's strategies, we have created a simplified version of the game, Tiny-Risk. It contains all the elements that make the game complicated but with constrained dynamics, further explained in the next section

Our initial experiments suggest that even with this minimized setting, well-known RL models like A2C and DDQN fail to learn complex strategies, resulting in a low win rate even against a random legal action bot.

To validate the second hypothesis, we take our initial steps to create an advantage actor critic-based variation of a decision transformer model (A2C-DT) that optimizes policy gradient in a much more efficient and stable fashion coupled with a target network.

We postulate that this model, based on temporal trajectories of the state changes by agents' as well as opponents actions (without externally modelling opponents' policy) can intrinsically model and identify their strategies. This gives the agent the power to dynamically adjust its relationships with other players and seek potential collaboration and defection by design, even without prior human gameplay data. Our initial experiments with this agent against the same random agents as before elicits its superiority in deciphering potential state transitions resulting in an improved winning rate.

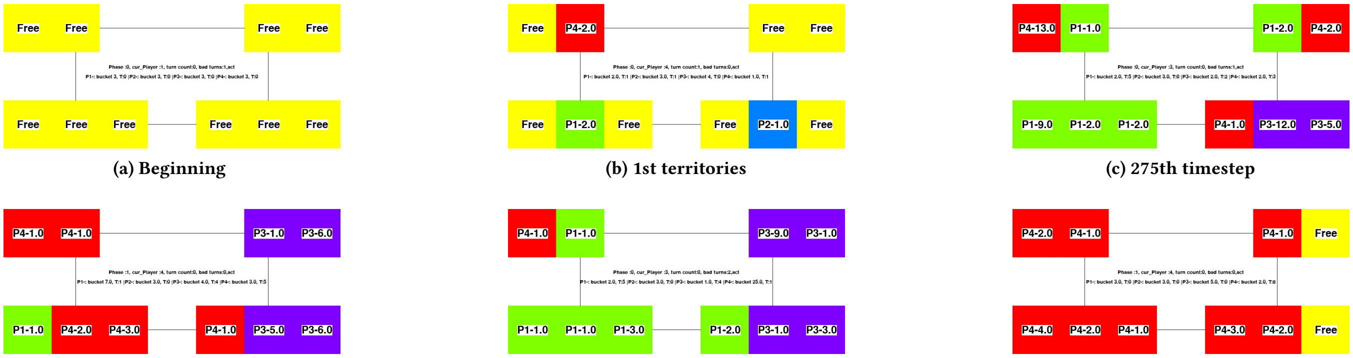
We further postulate that such a model will perform better when trained against an actor with personality rather than a random agent. Given such actors would make actions logically following a temporal pattern. Also, as seen from the recent success of the transformer models in language understanding tasks there is reason to believe A2C-DT could identify such patterns. As the DT is coupled with an attention mechanism, it could be argued that it can estimate the horizon to approximate opponents' ongoing and changing strategy. Furthermore, models like Rwkv and Mamba promise a longer look ahead and retrospection than core transformer models and therefore can be expected to facilitate improved in-context modelling of the opponents, thus a higher potential for complex social behaviour. Future studies will delve deeper into specific analysis and incorporation of social interaction to decide a meta strategy like cooperation and defection along with venue of strategy in the model's value function along with validation of the third hypothesis.

2 TINY-RISK: ENVIRONMENT DESCRIPTION AND WORKFLOW

Risk is a turn-based game where the player aims to conquer the world map by eliminating all other players. Players achieve this by controlling territories and strategically deploying their armies. In the process, players can make temporary alliances by directly communicating with other players on dedicated channels or by supporting actions in the game. To reduce the complexity, the game setting is restricted to no-press.

Setup and Game Dynamics

The game board is divided into 4 continents, each with directly connected territories (10 territories in total). Each continent is connected with 2 other continents by a path through specific territories. fig. 1 In each cycle of the game, players take sequential turns having 3 phases:


Free
Free
Free
Free
Phase :0, cur_Player :1, turn count:0, bad turns:1,act
P1-: bucket 3, T:0 |P2-: bucket 3, T:0 |P3-: bucket 3, T:0 |P4-: bucket 3, T:0
Free
Free
Free
Free
Free
Free
Free
P4-2.0
Free
Free
Phase :0, cur_Player :4, turn count:1, bad turns:0,act
P1-: bucket 2.0, T:1 |P2-: bucket 3.0, T:1 |P3-: bucket 4, T:0 |P4-: bucket 1.0, T:1
Free
P1-2.0
Free
Free
P2-1.0
Free
P4-13.0
P1-1.0
P1-2.0
P4-2.0
Phase :0, cur_Player :3, turn count:0, bad turns:1,act
P1-: bucket 2.0, T:5 |P2-: bucket 3.0, T:0 |P3-: bucket 2.0, T:2 |P4-: bucket 2.0, T:3
P1-9.0
P1-2.0
P1-2.0
P4-1.0
P3-12.0
P3-5.0
P4-1.0
P4-1.0
P3-1.0
P3-6.0
Phase :1, cur_Player :4, turn count:0, bad turns:0,act
P1-: bucket 7.0, T:1 |P2-: bucket 3.0, T:0 |P3-: bucket 4.0, T:4 |P4-: bucket 3.0, T:5
P1-1.0
P4-2.0
P4-3.0
P4-1.0
P3-5.0
P3-6.0
P4-1.0
P1-1.0
P3-9.0
P3-1.0
Phase :0, cur_Player :3, turn count:0, bad turns:2,act
P1-: bucket 2.0, T:5 |P2-: bucket 3.0, T:0 |P3-: bucket 1.0, T:4 |P4-: bucket 25.0, T:1
P1-1.0
P1-1.0
P1-3.0
P1-2.0
P3-1.0
P3-3.0
P4-2.0
P4-1.0
P4-1.0
Free
Phase :1, cur_Player :4, turn count:0, bad turns:0,act
P1-: bucket 3.0, T:0 |P2-: bucket 3.0, T:0 |P3-: bucket 5.0, T:0 |P4-: bucket 2.0, T:8
P4-4.0
P4-2.0
P4-1.0
P4-3.0
P4-2.0
Free

(a) Beginning

(b) 1st territories

(c) 275th timestep

(d) 330th timestep

(e) 690th timestep

(f) End of game

Figure 1: Tiny Risk: Gameplay and map, showcasing different stages of the game, with players' occupied territory and troops

A. Troop Reinforcement: In this game, each player starts with 3 troops. In each cycle, the player gets a fixed number of troops, unlike risk, where troops are proportional to territories owned. This forces the player to try strategies like collaborations and positioning.

Start of the game: Players start by placing troops in unoccupied territories. While the first player has more choices to start, other players have a better chance to defend strategically. This, unlike random territory initialization in Risk, forces reasoning over positional benefit. The utility of card-to-troop exchange is removed to focus on cooperative behavior. Attack phase outcomes primarily introduce stochasticity.fig. 1b

After setting the 1st territory: fig.(1c, 1d, 1e), Actions are bounded only by the number of spare troops and player strategy. Therefore, players can reinforce occupied territories by placing spare troops. If they have troops left after one action they can take multiple actions and spread out through territories or concentrate strategically for their next move. This flexibility increases game complexity due to an almost unbounded number of actions. With each action in a phase, reachable states grow exponentially, encouraging more thoughtful gameplay.

End of the game for the player: fig. 1f, Once all the occupied territories of a player are lost, they lose the game and cannot place new troops, even in the empty territories, even if they have spare troops. The game progresses with the remaining players.

B. Attacking: Following 1st phase, players can attack adjacent territories owned by opponents or those that are still unoccupied.

Attack dynamics: Attackers can attack adjacent territory if they have at least one troop in the current territory. During the attack, the player selects territory, proportion of troops, and attack position. The outcome of combat is determined by 2 probability distributions inputting attacker and defender troops. If attacking troops are more, a wider normal distribution (0,1) is used, estimating the probability of the "attack-defending troops difference," i.e., "chance of winning," and compared against a randomly generated number between 0 and 1. If this number is smaller than the "chance of winning," then opponent troops are eliminated and the attacked loses an equal amount of troops, remaining troops are sent to the newly captured location, unless only 1 troop is left alive, in which case the previously occupied territory of the opponent is freed. If the player loses, both territories also lose troops equal to the minimum of attacking and defending troops.If the attacker's troops are less than the defender's troops, then a narrower distribution N(0,0.1) is used to estimate the winning chance. Which enables the agent to incorporate taking risks in their strategies, the outcome of this exchange in case of winning the opponent loses all its troops, and the attacker loses all its attacking troops, leaving at least 1 behind in the territory.

Similar to the 1st phase, the actions in the attack phase are also unbounded; that is, the player can take multiple steps and continue to attack sequentially until possible or until they want. It is the only means to acquire new territories.

Old and new sources of Stochasticity: In RISK, stochasticity is derived by the distribution of typically rolling out 3 attacking and 2 defending dice, which is simplified by using 2 normal distributions. The second simplification is sending attacking troops to newly conquered territory, where RISK forces the players to choose the number of troops to be sent.

C. Fortifying: Lastly, players can move armies only between connected territories they control to strengthen their defenses or prepare for future attacks. In RISK, connectivity is defined as any possible path between directly connected conquered territories. To manage complexity, only a transfer to an immediately connected territory is allowed. Unlike the 1st and 2nd phases, players can only make one action, forcing them to find the path in a dynamically changing environment.

Additional Constraints

While dice control, Risk cards, continent bonuses, and acquisition of opponent resources from eliminated opponents from the game are removed as they only added to the theoretical stochasticity of the environment. To enhance gameplay by increasing exposure to starting and ending scenarios, additional restrictions have been implemented

Maximum number of troops in a territory: Although this feature is editable, the experiment limits maximum capacity to 50 per territory. This forces agents to use different techniques after they achieve a certain level of defense. This is specifically to avoid strategies that involve creating a single indefeasible territory that can result in a deadlock when multiple agents try the same, resulting in a draw, which is not the worst but neither the best outcome, but is certain, therefore can result in high confidence, followed by a restricted exploration of social strategies. **Time limit of the game:** While the game can run for days, we created an editable feature, restricting the total number of steps per game episode to 3000 steps in our experiments. On average, a cycle could take 200 steps. Therefore we are forcing agents to learn strategies that can be achieved in 15 game cycles, restricting the search space.

Reward Function and Illegal Moves

In-game rewards: A reward of +1 is given for conquering every new territory. Conquering is defined by the presence of the player's troops at the new location. Additionally, -1 is given for losing a previously conquered territory.

End-game rewards: If a player is eliminated in the middle of a game, a -100 is rewarded, and at the end of the game, a winner is rewarded with a +100, and all losers again get a -100, which means the 2^{nd} best player only gets -100 and previously eliminated players get -200 in total apart from in-game rewards. In case of a draw by reaching the maximum time, all the players are given -100, resulting in all alive players sharing the 2nd place.

Penalties for illegal actions: A changeable variable is defined for taking any illegal move in the game. At every timestep, given the current territory acquisitions, allocated troops, spare troops, and phase, the possible actions that can be taken in the game are restricted. As these configurations change, so do the legal actions. Humans can make visual inferences, however, this is not the case for Agents, therefore the game board identifies and shares these actions as an action mask. However, given the model does not know how to use the mask, it can still make illegal moves. A predefined constant penalty is given for each such action. In our environment, we used -0.01, given a single player plays all the possible moves, for example, if the step limit per game is 3000, then a cumulative reward of -30 is accumulated in addition to the game draw reward.

Phase-Turn skip: Maximum possible illegal actions, as a changeable variable, are defined to ensure the game's progress for every phase of the turn in a given cycle. Once it is crossed, the game automatically moves to the next phase. In our experiments, we allowed the player to make 4 illegal actions per phase.

Environment Design

This subsection defines the state and action space, along with other observables.

State space: A 2D matrix of shape (10,2), each row corresponding to a location on the map, and the 1st column defines ownership of the territory, given by a number dedicated to the player, and the 2nd column represents the number of troops deployed.

Other observations: Specific information like the number of player's spare troops, along with the action mask depicting its legal actions for the timestep, are always visible to the player in their and other's turn. Common information such as the current phase, current agent, and current timestep is visible to all agents.

Hidden information: Information specific to other players is not visible, however is not difficult to estimate. Edges of the game are not directly visible and should be learned, given it is a static information it is not provided separately.

Action space: Each action is a pair of 2 variables, having different meanings during different actions. There are 32 possible action values for the 1st variable per timestep in any given phase; this is done to maintain uniformity of action space throughout the game. The 2nd variable is a continuous float between 0 and 1 inclusive.

-**Positioning actions:** For the 1st variable, the 1st 10 action values, starting with 0, correspond to distinct territories on the map, used in the 1st phase of the action cycle, the 31st action corresponds to moving to end the current phase, and the 32nd action for ending the turn, and actions between 11th and 30th are not legal. The 2nd variable corresponds to the proportion of spare players to be placed. The number of spare players is only bounded by the total number of cycles and can change dynamically. The proposed configuration can handle a large number of cycles.

-**Transitory actions:** Attack and fortification phases require troops to move from one place to another. The former required from conquered territory to opponent territory, for the latter both are conquered territories. For the 1st variable, actions between 11th and 30th inclusive, corresponding dedicated 20 unidirectional edges can be used, mapping to a specific form and to territory, 31st and 32nd action follows the same rules as positioning actions. The 2nd variable corresponds to the ratio of the troops to be sent. In case the proportion is 100%, the environment enforces restrictions to keep a single trooper and tries to send the remaining.

Action mask: The action space defines the legal actions for the agent per phase as a one-hot vector of length 32:

-**Positioning actions:** Its 10 values are set to 1 when the positioning is legal on the corresponding territory, i.e., either the player owns the territory, or the territory is unoccupied, and the player hasn't deployed its first troop, additionally the player has some spare troops, else these are 0. The 11th to 30th values are 0, and the 31st and 32nd actions are always legal, i.e., 1.

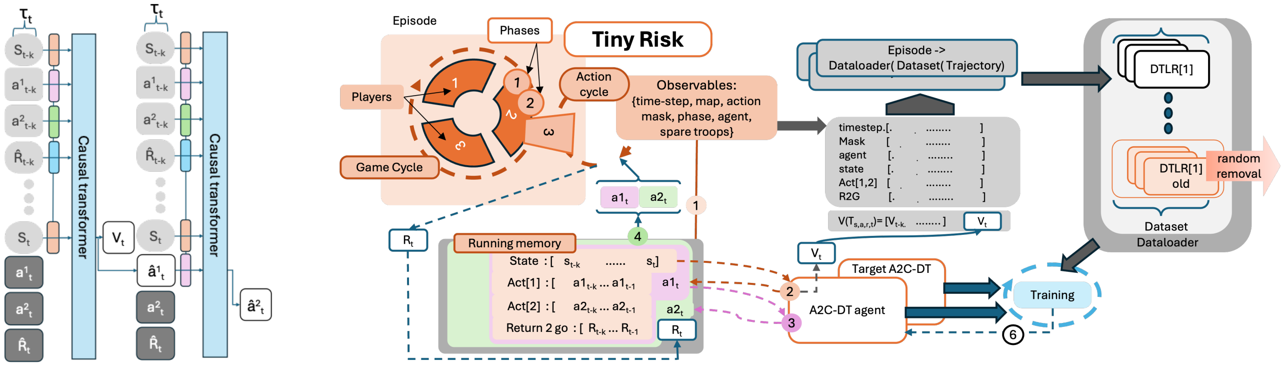
-**Transitory actions:** 1st 10 values are always 0 in the attack phase. Only those edge actions are legal where the "from" territory is owned by the player and has at least 2 troops, along with the "to" territory either occupied by an opponent or empty. In the fortify phase, however, the "to" territory should be self-owned.

Game loop: The game loops through cycles sequentially through each player and all three phases for the players. Until either the time limit is reached or all players but one are eliminated from the map.

3 MODEL ARCHITECTURE AND TRAINING PIPELINE

For the current problem, we used causal transformer models with 3 heads for each time step: two action heads (a_t^1 and a_t^2) and one value function head (V_t). Action 1 head a_t^1 has a SoftMax response of size 32, aligning with the action mask. Whereas, Action 2 a_t^2, given a continuous distribution, has a single value output between 0 and 1. Lastly, the value head gives an unbounded continuous output.

For a trajectory of context length k, a sequence of ordered state s_t, action 1 a_t^1, action 2 a_t^2, and return to go R_t is fed, resulting in $4k - 3$ tokens till s_t, which thereby produces V_t and the distribution of a_t^1. The action with the highest probability is selected, and in the case of multiple outputs with equal probability, a random selection is made among the highest values. To bind a_t^2 with the selected a_t^1,



(a) A2C-DT Autoregressive action prediction

(b) A2C-DT training pipeline with Tiny Risk

Figure 2: Workflow of A2C DT pipeline, depicting different stages training process

a_t^2 is autoregressively produced,fig. 2a, by updating the trajectory T_t and re-feeding it to the model, which is then used for the selected agent to take an action. This process happens only once per time step. At the beginning of every episode, "Return to go" is set to the highest possible achievable return (110 in the case of Tiny Risk).

Notice, unlike DT [13], A2C-DT predicts the value function estimate for the state trajectory instead of Return 2 go [28, 46, 47, 49]. This value function is used later for advantage estimation along with action and critic loss calculation, given the model learns directly from self-play. However, it can be extended to imitation learning if required. Also, note that in the proposed configuration, only the last predicted token is used; i.e., the current state of interest is always the last state token in the sequence.

3.1 Training pipeline:

The training pipeline, fig. 2b follows a cycle of three phases: the experience phase, the data preprocessing phase, and the modeling update. **Experience phase:** N trajectories for N episodes are recorded per cycle. For each episode, each agent maintains its own running memory of context length k and tensors of the complete history of observables for the given episode. Since the agent can still observe the game even when other players are playing, it stores the trajectories of state changes during this time as well. However, it does not store their action masks and actions. This segregates the time steps into two categories: with and without the player's actions, used differently for loss calculation. It can be argued that in such a game, a player can visualize possible opponents' legal moves. Still, we do not include this information, limiting the state-space size and expecting the agent to learn the dynamics independently. While it would be interesting to analyze the gradient map of the agent's predicted actions to understand the influence and dependence of the agent's policy based on opponents' recent actions, we leave this for future work. At each time step, the running memory is updated with normalized observables used for a_t^1, a_t^2, and V_t prediction. The initial cycles of the training pipeline use epsilon-based random legal action exploration, which reduces over episodes. The maintained record for each episode of observables, the agent's actions (predicted or randomly taken during exploration), along with predicted state-trajectory values, is passed to the data processing phase.

Data Processing phase: Three steps are taken in this phase. First, the trajectory of state observables per episode is normalized for training stability. Next, the discounted sum of future rewards until the end of the episode is estimated for each action. Lastly, given that the model requires a sequence of context length k for each prediction, to minimize space utilization, each episode trajectory is converted to a data loader of a single trajectory that produces a sequence of observables, rewards, and recorded predictions. Furthermore, a replay buffer of size X is maintained, where each new trajectory replaces an old one from the $X - N$ past oldest records. This replay buffer further acts as a data loader of data loaders, shuffling and procuring episode data loaders at training time.

Model update: During training, the model goes through all episode trajectories in random order. Each trajectory produces a batch of "Trajectory length × context length" individual elements; therefore, the batch size may vary. To standardize the updates, the batch was further chunked into mini-batches of a fixed size W. To normalize the impact of longer or shorter trajectories, the losses for each mini-batch were divided by the number of mini-batches per episode. For each mini-batch, a target model with a stable copy of the training model weights was used for $V(s_{t+1})$ prediction, which was further used in critic loss estimations. Weights from the training model were copied to the target model after every N episodes. A cosine annealing scheduler with warmup steps was used for the learning rate with the AdamW optimizer and a small weight decay.

3.2 Loss functions:

This subsection covers the definitions and formulas used for Actor and Critic losses. Following slight changes to the definitions of the Advantage actor-critic algorithm, these losses were predicted for each minibatch.

3.2.1 *Actor Loss:*

The actor loss was calculated only for sequences in the mini-batch ending with the agent's actions, whether predicted or randomly sampled during exploration. Therefore, there could be mini instances, but no such instances exist for a given minibatch.

In which case the model only tunes for Critic loss. Given Critic and Actor have a shared network, except the heads, the model still learns improved feature representation during critic updates. The policy gradient in the A2C model can be estimated by the following formula [30, 42, 43], where τ is the trajectory history of length k:

$$\nabla_\theta J(\theta) \approx \sum [\nabla_\theta \log \pi_\theta(a|\tau) \cdot A(\tau, a)]$$

However, the action a is composed of autoregressively predicting a^1 and a^2. The joint probability of taking both actions a^1 and a^2, given trajectory τ, is:

$$\pi_\theta(a = \{a^1, a^2\}|\tau) = \pi_\theta(a^1|\tau) \cdot \pi_\theta(a^2|a^1, \tau)$$

Therefore, the log probability of the action reduces to:

$$\log \pi_\theta(a = \{a^1, a^2\}|\tau) = \log \pi_\theta(a^1|\tau) + \log \pi_\theta(a^2|a^1, \tau)$$

There are two policy gradients that can be estimated: one for $\hat{a}^1$ and $\hat{a}^2_{v2}$ predicted autoregressivly, and the second for the prediction of $\hat{a}^2$ given the actual action taken a^1. Let's denote the predicted actions as $\hat{a}^1$ and $\hat{a}^2$, and the actual actions as a^1 and a^2.

Therefore:

$$LP_1 = \log \pi_\theta(a = \{\hat{a}^1, \hat{a}^2_{v2}\}|\tau) = \log \pi_\theta(\hat{a}^1|\tau) + \log \pi_\theta(\hat{a}^2_{v2}|\hat{a}^1, \tau)$$

$$LP_2 = \log \pi_\theta(\hat{a}^2|a^1, \tau)$$

The total log probability (LP) is a weighted combination:

$$LP = \rho \cdot LP_1 + \beta \cdot LP_2$$

And the policy gradient becomes:

$$\nabla_\theta J(\theta) \approx \sum \left[\nabla_\theta LP \cdot A(\tau, a = a^1, a^2)\right]$$

The log probabilities can be expressed using cross-entropy and binary cross-entropy:

$$\log \pi_\theta(\hat{a}^1|\tau) = \text{CrossEntropy}(\text{Logits}(\hat{a}^1), a^1)$$

$$\log \pi_\theta(\hat{a}^2_{v2}|\hat{a}^1, \tau) = \text{BinaryCrossEntropy}(p(a^2|\hat{a}^2_{v2}, \hat{a}^1), p(a^2|a^2, a^1))$$

$$\log \pi_\theta(\hat{a}^2|a^1, \tau) = \text{BinaryCrossEntropy}(p(a^2|\hat{a}^2, a^1), p(a^2|a^2, a^1))$$

Notice, the gradient of a^1_t are retained a^2_t. The final policy gradient per minibatch is where M is total number of minibatch in an episode, and N is the number of sequences ending in the agent's action.

$$\nabla_\theta J(\theta) \approx \frac{1}{M} \sum_{i=1}^{W} \frac{1}{N_i} \left[\nabla_\theta LP^{(i)} \cdot A(\tau^{(i)}, (a^1, a^2)^{(i)})\right] \qquad (1)$$

Advantage A, for instance, is defined as follows, where the discount return is pre-calculated during the data processing phase.

$$A(\tau^{(t)}, (a^1, a^2)^{(t)}) = \sum_{k=0}^{T-t} \gamma^k r_{t+k+1} - V(\tau_t)$$

3.2.2 Critic Loss: The critic loss minimizes the difference between the predicted value function $V_\theta(s)$ and the actual returns. As even the action cycles of other agents' rewards could be gained, the critic loss is optimized to predict trajectory value at each timestep.

critic loss is defined with the following formula [30]:

$$L_{\text{critic}} = \frac{1}{M \times W} \sum_{i=1}^{W} \left(R(\tau^{(i)}) - V_\theta(\tau^{(i)})\right)^2$$

where $R(\tau^{(i)})$ is the actual discounted cumulative reward. While this is the closest Monte Carlo approximation of the loss, we also incorporated the one-step TD error, which in tandem leads to a stable learning [3, 10], where the target network predicts the value at t+1. Therefore, the following definition of the loss was used, where θ and ϕ represent the training and target models, respectively.

$$\begin{aligned} L_{\text{critic}} = \frac{1}{M \times W} \sum_{i=1}^{W} \Big[\lambda \left(r^{(i)} + \gamma V_\phi(\tau'^{(i)}) - V_\theta(\tau^{(i)})\right)^2 \\ + \eta \left(R(\tau^{(i)}) - V_\theta(\tau^{(i)})\right)^2 \Big] \end{aligned} \qquad (2)$$

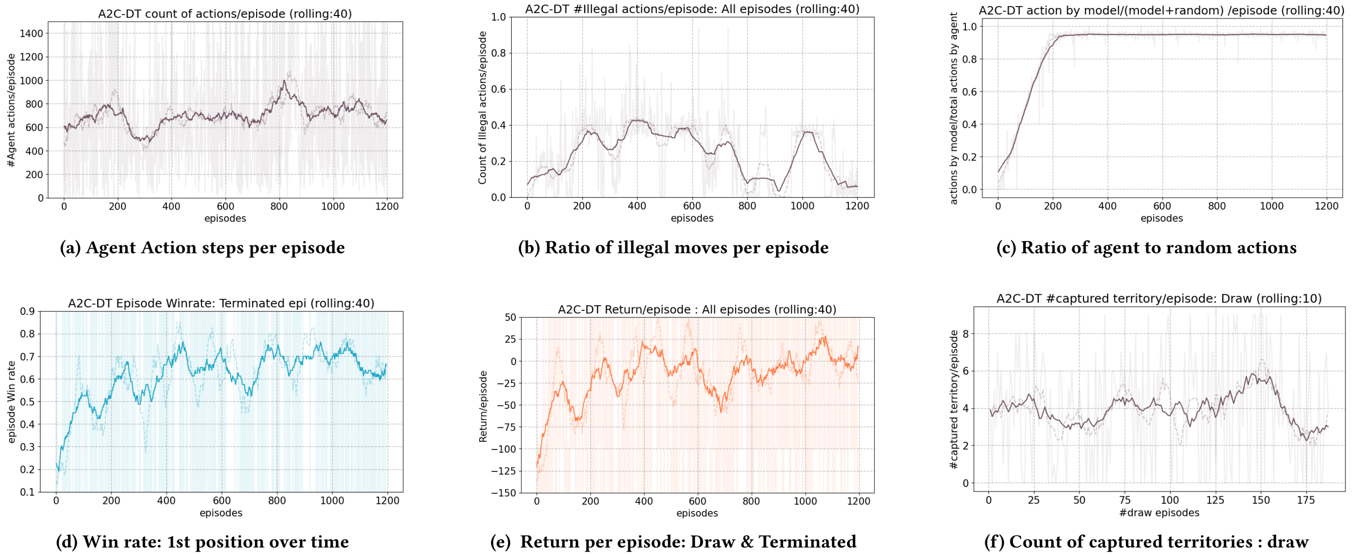
4 EXPERIMENTS AND RESULTS

This section covers the results of our preliminary experiments using A2C-DT on Tiny Risk. Most runs were conducted on a 3-player board with 10 territories. Our initial experiments focus on the learnability of the model, with A2C-DT competing against two random legal action agents with a time limit of 3000 steps per game.

During training, 85% of episodes terminated, and the remaining 15% resulted in a draw. Close to the 200th episode, on average, 95% of the moves were made by the A2C-DT agent fig. 3c. A 5% action stochasticity was introduced to maintain environmental complexity. Note that as the branching factor per action cycle increases exponentially for a phase, the impact of 5% stochasticity per action cycle also increases similarly. As seen in fig. 3b, 3a, even with a penalty of -0.01 per bad action, the model learns to avoid them over time, even with a dynamic legality of the action space, which increases as the model tries to explore and take illegal actions. The number of total actions taken per episode also increases, suggesting the learning and execution of longer-term strategies. As shown in fig. 3d, despite the stochasticity, the model achieves the highest 85% win rate in 1200 episodes against two random legal-action bots. In contrast, vanilla A2C and DDQN models showed no visible improvement even after 5 million episodes and are therefore omitted due to the significant difference in training time and their poor performance. This suggests high sample efficiency for our model, a point we aim to validate in future work.

The model's returns exhibit instability, potentially due to four factors: **Cosine Annealing Learning Rate Scheduler**: While aiding escape from local minima, the cyclical nature of cosine annealing can introduce oscillations in prediction accuracy. Furthermore, cyclical learning rate increases can amplify optimizer momentum, causing overshooting and oscillations around optimal parameter regions. **High Gradient Variance:** The sparsity of rewards in Tiny Risk leads to large gradient fluctuations and, consequently, unstable prediction performance. **Off-Policy Updates:** The mini-batch-based data management and network updates introduce off-policy elements. This can cause instability, as the data collection policy may differ from the updated policy. **Limited Hyperparameter Tuning:** While transformers are generally robust across various configurations, the extensive training time required for validating each parameter set limited our exploration. Due to computational constraints, we used a smaller transformer model (3 blocks, dimension 64) and a replay buffer of only 20 episodes. Given the promising performance of this base model, further tuning is likely to yield improvements.

The results shown in fig. 3e, 3f, the moving average of Return per episode, including draws and terminated episodes, aligns with our



(a) Agent Action steps per episode

(b) Ratio of illegal moves per episode

(c) Ratio of agent to random actions

(d) Win rate: 1st position over time

(e) Return per episode: Draw & Terminated

(f) Count of captured territories : draw

Figure 3: A2C-DT behavioral characteristics and learning performance in 3 player TR, against random legal bots

earlier discussion on the increasing win rate, depicted by the rise in average rewards, alongside the oscillations explained previously. The average reward decreases in later episodes with more steps, resulting in draws. However, even in these drawn episodes, the average number of territories the agent controls increases.

This suggests the agent, aiming for a longer game, is attempting to expand its territorial control. While potentially a viable strategy, draws still result in a -100 reward, only partially offset by the +1 reward per territory occupied. The reward function, while defining the social dynamics of SSDs, also governs the learnability of effective strategies. Balancing these considerations within the reward function is a major design challenge and a key requirement for applying Tiny Risk to real-world problem reduction. In experiments with a 10,000 timestep limit, the model began learning to create strongholds, leading to a higher frequency of draws. This, in turn, resulted in a substantial number of episodes with negative rewards, where the agent exhibited game suicide behavior. This phenomenon may indicate the limitations of a smaller model in navigating a highly complex environment, potentially due to restricted representational power. Larger models and longer training times, in conjunction with reward function redesign, could address this limitation.

5 CONCLUSION AND DISCUSSION

This research introduces Tiny Risk, a simplified variant of Risk designed to study cooperation in complex SSDs. Motivated by the limitations of existing MARL approaches in environments like Diplomacy and Civilization, TR balances complexity and tractability, enabling analysis of emergent strategies and the potential of social learning algorithms. While retaining Risk's core strategic elements, TR constrains dynamics for manageable analysis.

To address the performance gap of existing MARL models in SSDs and explore social learning potential (second hypothesis), we developed A2C-DT, an A2C-based Decision Transformer architecture. A2C-DT learns from state change trajectories influenced by its own and opponents' actions, implicitly modeling opponent strategies without explicit modeling or prior human data. This framework is easily adaptable to incorporate human data in the future. Our results show A2C-DT's high sample efficiency, achieving an 85% win rate against random agents within 1200 episodes. This suggests effective social situation identification through observing opponent-driven trajectory changes, enabling efficient learning in SSDs. The transformer's attention mechanism likely facilitates handling the dynamic nature of strategy initiation.

Despite its success in this limited setting, A2C-DT exhibits performance oscillations, likely attributable to the cosine annealing learning rate schedule, high gradient variance, off-policy mini-batch updates, and computational constraints on hyperparameter tuning.

To fully address the second hypothesis, future work must explore techniques for optimizing model training, including advantage normalization, GAE, and alternative learning rate schedules to mitigate oscillations. Furthermore, we will incorporate algorithmic modifications to better capture the nuances of cooperation and competition in TR, studying dynamic relationship building, the impact of induced moral values in opponents, and dynamically changing environmental factors. This will involve evaluating A2C-DT against more sophisticated opponents, incorporating explicit communication, and analyzing emergent cooperative behaviors.

Experiments with longer game horizons revealed a behavioral bias, highlighting the limitations of smaller models in representing complex long-term strategies. This underscores the importance of representational capacity in solving SSDs, motivating future work with larger models and extended training.

To explore the impact of social group size on agent intelligence (third hypothesis), we will train A2C-DT in larger, denser Risk environments and assess transfer learning to smaller scenarios. Inspired by the cultural brain hypothesis, we hypothesize that experience in complex social settings will enhance performance in simpler ones. This research contributes to the development of more sophisticated, socially intelligent agents.

REFERENCES

[1] [n.d.]. hasbro.com. https://www.hasbro.com/common/instruct/risk.pdf. [Accessed 17-10-2024].

[2] Hiroyuki Adachi, Hiroyuki Kamekawa, and Shigeki Iwata. 1987. Shogi on n× n board is complete in exponential time. *Trans. IEICE* 70 (1987), 1843–1852.

[3] Artemij Amiranashvili, Alexey Dosovitskiy, Vladlen Koltun, and Thomas Brox. 2018. TD or not TD: Analyzing the role of temporal differencing in deep reinforcement learning. *arXiv preprint arXiv:1806.01175* (2018).

[4] Thomas Anthony, Tom Eccles, Andrea Tacchetti, János Kramár, Ian Gemp, Thomas Hudson, Nicolas Porcel, Marc Lanctot, Julien Pérolat, Richard Everett, et al. 2020. Learning to play no-press diplomacy with best response policy iteration. *Advances in Neural Information Processing Systems* 33 (2020), 17987–18003.

[5] Victor Asal, Steve S Sin, Nolan P Fahrenkopf, and Xiaoye She. 2014. The comparative politics game show: Using games to teach comparative politics theories. *International Studies Perspectives* 15, 3 (2014), 347–358.

[6] Benjamin J Ashton, Alex Thornton, and Amanda R Ridley. 2018. An intraspecific appraisal of the social intelligence hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences* 373, 1756 (2018), 20170288.

[7] Anton Bakhtin, David Wu, Adam Lerer, and Noam Brown. 2021. No-press diplomacy from scratch. *Advances in Neural Information Processing Systems* 34 (2021), 18063–18074.

[8] Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. 2022. Mastering the game of no-press Diplomacy via human-regularized reinforcement learning and planning. *arXiv preprint arXiv:2210.05492* (2022).

[9] Wenhang Bao and Xiao-yang Liu. 2019. Multi-agent deep reinforcement learning for liquidation strategy analysis. *arXiv preprint arXiv:1906.11046* (2019).

[10] Andrew Barto and Michael Duff. 1993. Monte Carlo matrix inversion and reinforcement learning. *Advances in neural information processing systems* 6 (1993).

[11] Erik Blomqvist. 2020. Playing the game of Risk with an AlphaZero agent.

[12] Jamie Carr. 2020. Using Graph Convolutional Networks and TD ($lambda$) to play the game of Risk. *arXiv preprint arXiv:2009.06355* (2020).

[13] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[14] Leda Cosmides, H Clark Barrett, and John Tooby. 2010. Adaptive specializations, social exchange, and the evolution of human intelligence. *Proceedings of the National Academy of Sciences* 107, supplement_2 (2010), 9007–9014.

[15] Dave De Jonge and Carles Sierra. 2015. NB 3: a multilateral negotiation algorithm for large, non-linear agreement spaces with limited time. *Autonomous Agents and Multi-Agent Systems* 29, 5 (2015), 896–942.

[16] Dave De Jonge and Dongmo Zhang. 2017. Automated negotiations for general game playing. (2017).

[17] 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, et al. 2022. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science* 378, 6624 (2022), 1067–1074.

[18] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. 2007. Approximate and online multi-issue negotiation. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. 1–8.

[19] Aviezri S Fraenkel and David Lichtenstein. 1981. Computing a perfect strategy for n× n chess requires time exponential in n. In *International Colloquium on Automata, Languages, and Programming*. Springer, 278–293.

[20] David Gordon. [n.d.]. Risk vs Diplomacy — cardboardrepublic.com. https://www.cardboardrepublic.com/classics/risk-vs-diplomacy. [Accessed 17-10-2024].

[21] Jonathan Gray, Adam Lerer, Anton Bakhtin, and Noam Brown. 2020. Human-level performance in no-press diplomacy via equilibrium search. *arXiv preprint arXiv:2010.02923* (2020).

[22] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. 2018. Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in neural information processing systems* 31 (2018).

[23] Athul Paul Jacob, David J Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. 2022. Modeling strong and human-like gameplay with KL-regularized search. In *International Conference on Machine Learning*. PMLR, 9695–9728.

[24] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*. PMLR, 3040–3049.

[25] Hyun-Rok Lee and Taesik Lee. 2021. Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response. *European Journal of Operational Research* 291, 1 (2021), 296–308.

[26] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037* (2017).

[27] Xihan Li, Jia Zhang, Jiang Bian, Yunhai Tong, and Tie-Yan Liu. 2019. A cooperative multi-agent reinforcement learning framework for resource balancing in complex logistics network. *arXiv preprint arXiv:1903.00714* (2019).

[28] Haochen Liu, Zhiyu Huang, Xiaoyu Mo, and Chen Lv. 2022. Augmenting reinforcement learning with transformer-based scene representation learning for decision-making of autonomous driving. *arXiv preprint arXiv:2208.12263* (2022).

[29] Michael P Marks. 1998. Using the Game of" Risk" to Teach International Relations. *International Studies Notes* (1998), 11–18.

[30] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:6875312

[31] Michael Muthukrishna, Michael Doebeli, Maciej Chudek, and Joseph Henrich. 2018. The Cultural Brain Hypothesis: How culture drives brain expansion, sociality, and life history. *PLoS computational biology* 14, 11 (2018), e1006504.

[32] Zepeng Ning and Lihua Xie. 2024. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence* (2024).

[33] Fredrik Olsson. 2005. A Multi-Agent System for playing the board game Risk. https://api.semanticscholar.org/CorpusID:62641882

[34] Jason Osborne. 2003. Markov Chains for the RISK Board Game Revisited. *Mathematics Magazine* 76 (2003), 129 – 135. https://api.semanticscholar.org/CorpusID:11055761

[35] Praveen Palanisamy. 2020. Multi-agent connected autonomous driving using deep reinforcement learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.

[36] Philip Paquette, Yuchen Lu, Seton Steven Bocco, Max Smith, Satya O-G, Jonathan K Kummerfeld, Joelle Pineau, Satinder Singh, and Aaron C Courville. 2019. No-press diplomacy: Modeling multi-agent gameplay. *Advances in Neural Information Processing Systems* 32 (2019).

[37] Julien Perolat, Joel Z Leibo, Vinicius Zambaldi, Charles Beattie, Karl Tuyls, and Thore Graepel. 2017. A multi-agent reinforcement learning model of common-pool resource appropriation. *Advances in neural information processing systems* 30 (2017).

[38] Siyuan Qi, Shuo Chen, Yexin Li, Xiangyu Kong, Junqi Wang, Bangcheng Yang, Pring Wong, Yifan Zhong, Xiaoyuan Zhang, Zhaowei Zhang, et al. 2024. CivRealm: A learning and reasoning odyssey in Civilization for decision-making agents. *arXiv preprint arXiv:2401.10568* (2024).

[39] Fabian Ritz, Daniel Ratke, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. 2021. A sustainable ecosystem through emergent cooperation in multi-agent reinforcement learning. In *Artificial Life Conference Proceedings 33*, Vol. 2021. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ..., 74.

[40] John Robson. 1983. The Complexity of Go. *IFIP Congress Series* 9, 413–417.

[41] Alexandra G Rosati. 2017. Foraging cognition: reviving the ecological intelligence hypothesis. *Trends in cognitive sciences* 21, 9 (2017), 691–702.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[43] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems* 12 (1999).

[44] Barış Tan. 1997. Markov chains and the RISK board game. *Mathematics Magazine* 70 (1997), 349–357. https://api.semanticscholar.org/CorpusID:120096188

[45] John Tromp and Gunnar Farnebäck. 2006. Combinatorics of go. In *International Conference on Computers and Games*. Springer, 84–99.

[46] Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. 2024. Elastic decision transformer. *Advances in Neural Information Processing Systems* 36 (2024).

[47] Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*. PMLR, 38989–39007.

[48] Hongming Zhang and Tianyang Yu. 2020. AlphaZero. *Deep Reinforcement Learning: Fundamentals, Research and Applications* (2020), 391–415.

[49] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *international conference on machine learning*. PMLR, 27042–27059.