Inferring Relationship using Theory of Mind in Press Diplomacy

Hyeonchang Jeon [1 *] **Wonsang You** [1 *] **Songmi Oh** [2 *] **Hoyoun Jung** [2] **Kyung-Joong Kim** [2]

Abstract

Diplomacy is a turn-based, non-cooperative multiplayer game. In the Press version, the relationships among players change dynamically depending on both the public situation and private communications. To negotiate better with others, an agent should infer the mental states of others to identify relationships that are not explicit. In this paper, we propose the Graph-based Theory of Mind Network (GToMnet) that focuses on understanding relationships using the Theory of Mind (ToM). We add graph neural networks (GNNs) to the ToM neural network (ToMnet) to embed trust. To evaluate the GToMnet, we use it to predict agent responses. If successful, the agents can understand relationships with others to predict the acceptance of the negotiation. Our work is also applicable to other multi-agent reinforcement learning (MARL) problems featuring complex relationships, such as sequential social dilemmas.

1. Introduction

Diplomacy is a turn-based, non-cooperative game in which seven players compete to expand their territories and occupy a majority of supply centers. To conquer many regions effectively, players should cooperate as well as compete, so they negotiate with others depending on the situation and relationships. Since the relationship between players is not revealed explicitly, inferring the internal states of others, such as their unique characteristics or accumulated trust, is key when deciding with whom and when to ally and when to end it.

Previous Press Diplomacy (Fabregues et al., 2010; Ferreira et al., 2015) studies focused on trust between agents. However, these works did not consider how much other agents trust oneself. To this end, it is essential to comprehend the internal states of others and their needs; this is figured out by the Theory of Mind (ToM) which is the ability to infer the desires, beliefs, and intentions of others (Premack & Woodruff, 1978).

ToM-inspired models (Baker et al., 2011; Baker & Tenenbaum, 2014; Baker et al., 2017; Rabinowitz et al., 2018) that infer the internal states of agents have been extensively studied. Especially, the Theory of Mind network (ToMnet) (Rabinowitz et al., 2018) understands the internal state of artificial agents like our approach. Recently, the ToM has been applied to multi-agent reinforcement learning in fixed relations (Wang et al., 2021). Here, we infer dynamically changing relationships by identifying the amount of trust accumulated by each agent and the unique characteristics of the agent.

To infer the agent's internal relationships, we try to predict the response of sent messages in the negotiation. Since the relationship between agents is changed through messages, we choose the Press Diplomacy game which allows communication between the agents in the negotiation phase. The inferred agents are DipBlue agents, which change the strategy depending on parameters related to the ratio of trust.

In this paper, we propose the Graph-based Theory of Mind Network (GToMnet) using graph neural network(GNN) and ToMnet (Rabinowitz et al., 2018) to predict the responses of sent messages. Using GNN, GToMnet embed the public relationships effectively. The GToMnet consists of two networks: $Character\ Network$ and $Prediction\ Network$ as in ToMnet. $Character\ Network$ embeds the fixed characteristics of the DipBlue agent through GNN modules in past episodes. $Prediction\ Network$ infers the response of others, understanding accumulated trust in the current episode and embedded features from $Character\ Network$. Overall process detail is shown in Fig 1. Our method that infers the mental state of trust-based agents in Diplomacy contributes to the agent-based modeling community.

2. Background

Diplomacy Diplomacy is a zero-sum board game in which seven powers take over each other's provinces prior to World War I. The objective of the game is to rule Europe by occupying 18 supply centers located in specific provinces. Each

[1] Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology(GIST), Gwangju, South Korea [2] School of Integrated Technology, Gwangju Institute of Science and Technology(GIST), Gwangju, South Korea. Correspondence to: Kyung-Joong Kim <kjkim@gist.ac.kr>.

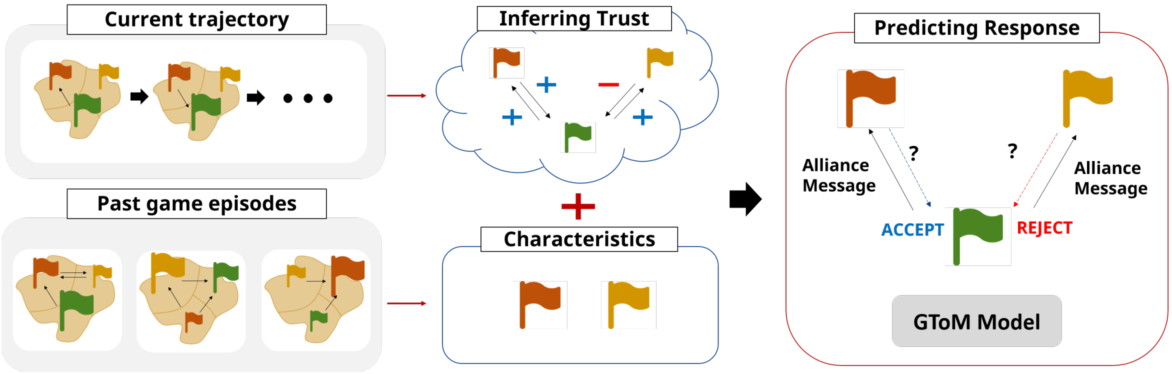

Figure 1. **Graph-based ToM process** : Description of how the GToMnet infers the response of the target agent in current game. The GToMnet understands the fixed characteristics of the target agent by observing the past game episodes. Then, the GToMnet tracks the accumulated trust using previous steps in the current episode. Totally, using accumulated trust in the current trajectory and fixed characteristics in past episodes, the GToMnet predicts the response of the target agent.

power makes an order to the $Army$ and $Fleet$ units to conquer or protect the territory at each turn and can own as many units as the number of supply centers it has.

The game lasts up to 100 years, and each year proceeds in 5 phases: Spring Movement, Spring Retreat, Fall Movement, Fall Retreat, and Winter Adjustment. In the Movement phase, the available orders are $Hold$, $Move$, $Support$ and $Convoy$. The players can protect their region by holding units, conquer a province by moving units to the province and sometimes support other units to defeat their enemies at their destinations. The $Army$ units can get across the water through $Convoy$ orders by $Fleet$ units. Only two orders are available in the Retreat phase: $Retreat$ and $Disband$. If a unit is defeated, the unit can retreat to an unoccupied adjacent location, and the unit with nowhere to go will be dislodged. In the Adjustment phase, the ownership of supply centers changes if the new owner occupies the location. In this phase, the player can choose $Build$, $Disband$, or $Waive$ according to the number of owned supply centers. Players build or disband their units according to the number of supply centers they own and can also waive the chance of $Build$ orders. The game ends if one player occupies more than 18 supply centers or all players agree to draw.

Level 1 Negotiation DipGame (Fabregues et al., 2010) defines a new communication syntax termed “L Language.” There are eight levels of communication, including arguing, explaining, and negotiating a deal. The complexity of expression increases as the level becomes higher. Here, we use the simplest level when negotiating a deal. Level 1 language features three sub-levels: $Negotiation$, $Deal$, and $Offer$. During $Negotiation$ level, the options indicate sending ($Propose$), responding ($Accept$ or $Reject$), and ignoring ($Withdraw$). The $Deal$ level shows the type of content of the suggested deal being exchanged. At the $Offer$ level, the player sends the core content of the messages. For example, “Germany accepts agrees with England that the two countries are allied against Russia.” The $Negotiation$ word is $Accept$, the $Deal$ word is $Agree$, and the $Offer$ word is $Alliance$.

The DipBlue Agent DipBlue (Ferreira et al., 2015) agent is a Diplomacy artificial intelligence that negotiates with opponents and exploits trust reasoning to win. In Press version Diplomacy, players have time to negotiate before making orders. During this phase, DipBlue agent employs three negotiation skills through L Language level 1 communication: $Peace\ agreement$, an $Alliance\ against\ enemies$, and $Request\ for\ unit\ orders\ to\ allies$. DipBlue agent uses trust reasoning to react to opponent betrayals. The DipBlue agent calculates the trust ratio for the opponents, taking into account who attacks whom and who are the enemies of its alliance, and updates that value continuously throughout the game. DipBlue agent consists of a negotiator and five advisers, $MapTactician$, $FortuneTeller$, $TeamBuilder$, $AgreementExecutor$, and $WordKeeper$. The negotiator handles negotiation messages between oneself and the other players, and each adviser evaluates the values of orders using its criteria. The final values of orders are determined by summing the weighted values of the advisers.

DipBlue agent implementation is very modular. Thus, the characteristics or strategy of DipBlue can be varied by customizing the adviser types or weight. In our experiments, we only adjusted the weights of $AgreementExecutor$ and $WordKeeper$ so that there were agents with varying degrees of importance for trust values.

3. Method

Input Representation In this section, we describe our GToMnet input. We used a public board state $s^t \in S$ and a private message state $m^t_{i,j}$ for sender i and re-

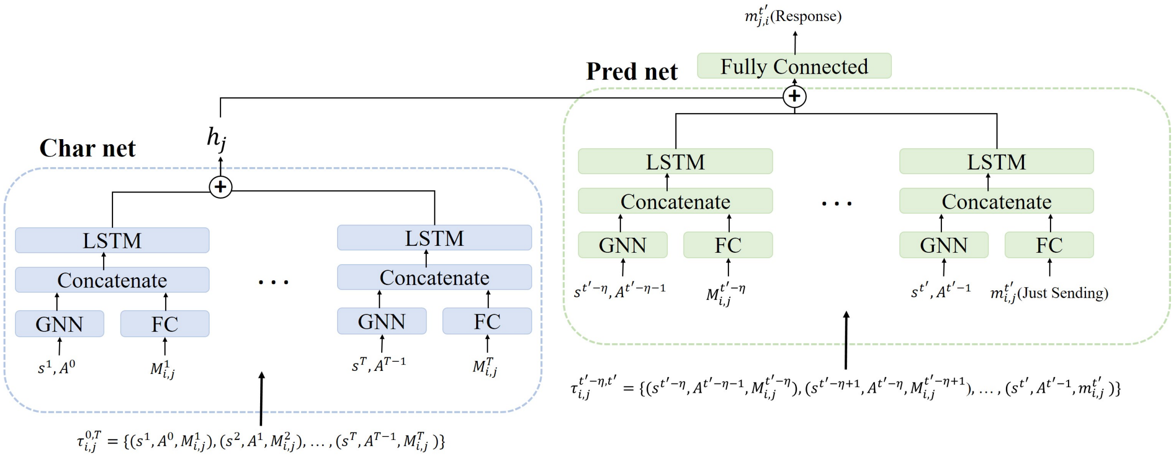
$m_{j,i}^{t'}$(Response)
Fully Connected
Pred net
Char net
h_j
LSTM
Concatenate
GNN
FC
s^1, A^0
$M_{i,j}^1$
s^T, A^{T-1}
$M_{i,j}^T$
$s^{t'-\eta}, A^{t'-\eta-1}$
$M_{i,j}^{t'-\eta}$
$s^{t'}, A^{t'-1}$
$m_{i,j}^{t'}$(Just Sending)
$\tau_{i,j}^{0,T} = \{(s^1, A^0, M_{i,j}^1), (s^2, A^1, M_{i,j}^2), \dots, (s^T, A^{T-1}, M_{i,j}^T)\}$
$\tau_{i,j}^{t'-\eta,t'} = \{(s^{t'-\eta}, A^{t'-\eta-1}, M_{i,j}^{t'-\eta}), (s^{t'-\eta+1}, A^{t'-\eta}, M_{i,j}^{t'-\eta+1}), \dots, (s^{t'}, A^{t'-1}, m_{i,j}^{t'})\}$

Figure 2. **Overview of the GToMnet Architecture** The past full trajectories $\tau_{i,j}^{0,T}$ include the board state s^t, the previous order state A^{t-1}, and the message state $m_{i,j}^t$. $Character\ Network$ embeds past trajectories within agent characteristics. Next, $Prediction\ Network$ infers the acceptance of message $m_{j,i}^{t'}$ using the current partial trajectories $\tau_{i,j}^{t'-\eta,t'}$ and the sent message $m_{i,j}^{t'}$.

ceiver j at time t. Message pairs at time t are $M_{i,j}^t = (m_{i,j}^1, m_{j,i}^1, ..., m_{i,j}^k, m_{j,i}^k)$ for k messages. The n agents simultaneously choose a joint action and modify to previous action features $A^t = g(a_1^t, ..., a_n^t)$ (actually, submitting action is not simultaneously but processed simultaneously in the game). All actions are determined by rule-based policy $a_i^t \sim \pi_i(a_i^t | s^t, M_{i,j}^t)$ and proceed to the next states via a transition function $s^{t+1} = f(s^t, A^t)$. Trajectory τ is $\tau_{i,j}^{p,k} = \{(s^t, A^{t-1}, M_{i,j}^t)\}_{t=p}^k$ for sender i and receiver j. Past trajectories are $\tau_{i,j}^{0,T}$, where T is a terminal step. Using the past trajectories, we understand the unique characteristics of the target agent j. Current partial trajectory is $\tau_{i,j}^{t'-\eta,t'}$, but there is only one sent message in t'. Through the LSTM network in $Prediction\ Network$, our network understands the current trust of the agent.

The Character Network The inputs of $Character\ Network$ are the past trajectories $\tau_{i,j}^{0,T}$. There are two forms of information: public information $s^t \in \mathbb{R}^{81\times35}$, $A^t \in \mathbb{R}^{81\times40}$ as in DipNet (Paquette et al., 2019) and private information $M_{i,j}^t \in \mathbb{R}^{10\times76}$ for maximum 10 message pairs. To separate the public and private information that agent i has on agent j, we divide the graph into an explicit graph $\mathcal{G}_i^{exp}$ and an implicit embedding. The former contains the public information of all agents as revealed by the board state and the previous order state. To embed the node information, we use the graph convolution network (GCN) (Kipf & Welling, 2016) of DipNet (Paquette et al., 2019) where Adj is the adjacent matrix of provinces.

On the other hand, the latter implicit network consists of two fully connected layers and embeds the private messages between i and j, e_{ij}^{imp}. Then, we concatenate both outputs of graphs and pass through LSTM (Hochreiter & Schmidhuber, 1997). Lastly, we extract the h_j representing the attributes of agent j.

The Prediction Network To consider the trust between agents, we use the GNN and LSTM to encode the current partial trajectory $\tau_{ij}^{t-\eta,t}$ in $Prediction\ Network$. As for the past trajectories in $Character\ Network$, we employ both public and private information. The current partial trajectory is first embedded by the GNN modules and then passed through LSTM to discover how relationships change during the current episode. The LSTM output is concatenated with that of $CharacterNetwork$ and the current sent message $m_{i,j}^t$. Lastly, the concatenated output is passed to a fully connected layer with a sigmoid activation function to predict the response $m_{j,i}^t$.

4. Experimental Settings

To evaluate GToMnet, we set the problem that the network predicts whether the DipBlue agent will accept or reject an alliance request another DipBlue agent makes during the Diplomacy Negotiation phase. To train the GToMnet, we collected gameplay data such as board states and messages with DipBlue agents that considered trust differently.

The Agents We implemented the DipBlue agent using the Diplomacy game engine of DipNet (Paquette et al., 2019). Although all of the five advisers of DipBlue agent can be adjusted by their weights, we only adjusted weights of two advisers, $AgreementExecutor$ and $WordKeeper$.

During the negotiation phase, bots send requests to others and decide whether to accept the suggestions of others based on how much they trust them. Eventually, how much the bots consider the trust is determined by the

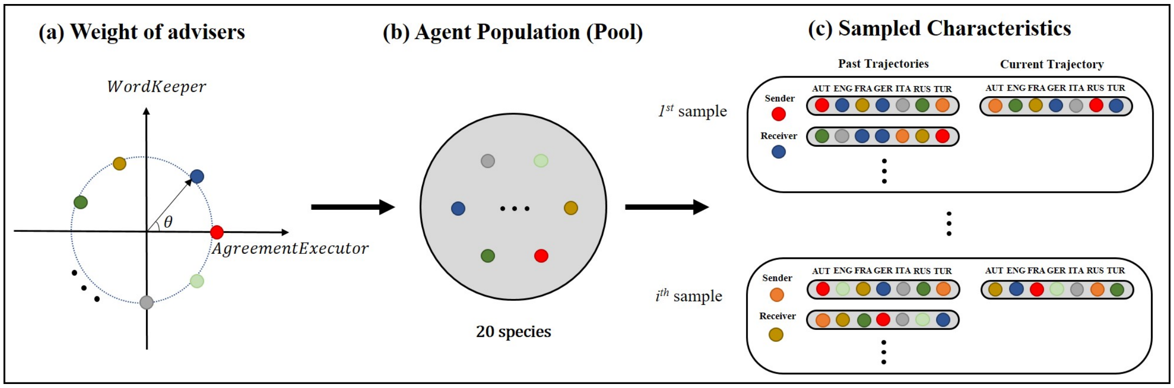
(a) Weight of advisers
WordKeeper
θ
AgreementExecutor
(b) Agent Population (Pool)
20 species
(c) Sampled Characteristics
Past Trajectories
Current Trajectory
1st sample
Sender
Receiver
AUT ENG FRA GER ITA RUS TUR
AUT ENG FRA GER ITA RUS TUR
ith sample
Sender
Receiver
AUT ENG FRA GER ITA RUS TUR
AUT ENG FRA GER ITA RUS TUR

Figure 3. **Description of Data Collection** : **a)** Agent characteristics can be weighted. The trustworthiness of both oneself and others is considered. The horizontal axis is the extent to which I consider how much others trust me (this parameter is employed in the $AgreementExecuto$r). The vertical axis is how much I trust others and is used in the $WordKeeper$. **b)** Twenty different coordinates (the angles θ) are sampled from the unit circle to create a population pool. **c)** Sampling of seven species from the pool (replacement is permitted). The sampled species are assigned to seven adjusted DipBlue agents.

weights of the two aforementioned advisers Fig 3 (a). $AgreementExecutor$ adviser considers how much the agent trusts others, while $WordKeeper$ considers how much the opponent agents trust the agent. For example, an agent who does not care what anyone else thinks will very likely not perform what was negotiated. Also, an agent that gives importance to trust in others will not attack the opponents with a high trust ratio and believe that such behavior will be reciprocated. We use the angle θ to control the weights of the two advisers; such adjustment finally changes their responses.

Data collection We make a population pool with 20 characteristics by adjusting the weights of $AgreementExecutor$ and $WordKeeper$, and sampling with replacement is employed to extract 7 for one sample data. 500,000 samples were used for training and 10,000 samples for testing.

We considered 20 characteristics and used 500,000 samples for training and 10,000 samples for testing. Sampling with replacement was employed to extract 7 characteristics. One sample data consists of several episodes and one current trajectory with the same agents. To match the $Character\ Network$ to the characteristics of specific agents, we shuffle the powers of all past trajectories to avoid the character networks embedded in those powers. Current trajectories are collected over 40 steps, and seven characteristics of countries are extracted and then randomly shuffled.

5. Potential Value and Future Work

We explored the capability of inferring accumulated trust between rule-based agents 'DipBlue' in a communicable multi-agent environment, Press Diplomacy. Our work is expected to be a good milestone for agent-based modeling of real-world problems. People shift between cooperative and competitive behavior in many interactions such as negotiations, auctions, and international diplomacy. Also, some people do not help others who are useless and sometimes betray others when the opportunity cost of sustaining the relationship is greater than that of betrayal. For an agent to adapt well to such a complex situation, it is most important to recognize changing relationships, predict the behaviors of others, and then decide how to optimize one's own position. We plan to demonstrate that the trust of agents depicting this tendency of humans can be inferred by observing interactions through the experiment.

In a subsequent study, the usefulness of the proposed model will be shown by extending outside of the realm of diplomacy. First of all, more general trust-based agents will be designed for applying our model to other environments out of the DipBlue in Press Diplomacy. For generality, we will implement a data-driven model using supervised learning or reinforcement learning. Furthermore, other social interaction environments will be added to demonstrate the effectiveness of our GToMnet architecture. The candidates are environments where cooperation and competition are mixed so that agents can move based on trust.

Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the MSIT(2021R1A4A1030075)

References

Baker, C., Saxe, R., and Tenenbaum, J. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science* *society*, volume 33, 2011.

Baker, C. L. and Tenenbaum, J. B. Modeling human plan recognition using bayesian theory of mind. *Plan, activity, and intent recognition: Theory and practice*, 7:177–204, 2014.

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

Fabregues, A., Navarro, D., Serrano, A., and Sierra, C. Dipgame: A testbed for multiagent systems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pp. 1619–1620. Citeseer, 2010.

Ferreira, A., Cardoso, H. L., and Reis, L. P. Dipblue: A diplomacy agent with strategic and trust reasoning. In *ICAART 2015-7th International Conference on Agents and Artificial Intelligence, Proceedings*, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Paquette, P., Lu, Y., Bocco, S. S., Smith, M., O-G, S., Kummerfeld, J. K., Pineau, J., Singh, S., and Courville, A. C. No-press diplomacy: Modeling multi-agent gameplay. *Advances in Neural Information Processing Systems*, 32, 2019.

Premack, D. and Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4): 515–526, 1978.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Wang, Y., Xu, J., Wang, Y., et al. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. 2021.