

## APPENDIX

### A PROMPTS

Here we provide all the prompts we designed and utilized in this paper:

System Prompt:

You are a helpful assistant that strictly follows the user's instructions.

Head Prompt:

Tic Tac Toe is a two-player game played on a grid.

Players take turns marking a space with their respective symbol. The goal is to get multiple of one's own symbols in a row, either horizontally, vertically, or diagonally, before the opponent does. If all nine squares are filled and no player has three in a row, the game is a draw.

The Tic Tac Toe game is played on a {grid\_size} by {grid\_size} grid, with the winning length as {win\_len}. Each move is represented by a string consisting of two parts: the column (C) and the row (R), in that order. For instance, C1R2 means the movement at the position of the first column and the second row of the grid. You are playing this game with the user (opponent).

Direct Prompt (first go):

You are the first to go. The legal positions are {legal\_positions}. Choose one move from these legal positions to set up advantages.

Your output should be of the following format:

Move:

Your move wrapped with <>, e.g., <C1R1>, <C1R2>, <C1R3>

(non first go)

Now, your opponent has finished moves: {opponent\_moves}.

You have finished moves: {agent\_moves}. The legal positions are {legal\_positions}. Choose one move from these legal positions to set up advantages.

Your output should be of the following format:

Move:

Your move wrapped with <>, e.g., <C1R1>, <C1R2>, <C1R3>

Chain-of-Thought:

Now, your opponent has finished moves: {opponent\_moves}.

You have finished moves: {agent\_moves}.

The legal positions are {legal\_positions}. First think about your current situation, then choose one move from legal positions to set up advantages.

Your output should be of the following format:

Thought:

Your thought.

Move:

Your move wrapped with <>, e.g., <C1R1>, <C1R2>, <C1R3>

Situation Assessment:

Given this situation, analyze your winning rate and provide a concrete value (from 0 to 100). Conclude in the last line "The winning rate is {s}", where s the integer id of the choice.

Reward Assessment:

Assume {side} will take {next\_move} as the next move. What is the advantage score of {side} for this move? Use a score on a scale of 0 - 100 to represent this score. Conclude in the last line "The advantage score for {side} is {{s}}", where s is the score.

## B TEXAS HOLDEM POKER

The explanations of Texas Holdem Poker actions:

1. FOLD: You decide not to play the hand and discard your cards.
2. CHECK: Declining the opportunity to bet. It's like saying 'I'm still in the game, but I don't want to bet right now.
3. CALL: Matching the current highest bet to stay in the hand.
4. RAISE.3BB: Raising the bet to three times the big blind amount.
5. RAISE.HALF.POT: Raising to an amount equal to half the current pot size.
6. RAISE.POT: Raising to an amount equal to the current pot size.
7. RAISE.2POT: Raising to an amount equal to twice the current pot size.
8. ALL\_IN: Betting all your chips.
9. SMALL.BLIND: A forced bet that's typically half the size of the big blind. It rotates around the table.
10. BIG.BLIND: A forced bet that sets the initial pot amount and action. It's typically twice the size of the small blind and rotates around the table.

## C MAJOR ADAPTIONS TO AUTONOMOUS AGENTS

For CoT, CoT-SC, and Prompt, please refer to A for which prompts are utilized. It is because prompts are the most important part for these methods.

For ReAct, we follow the prompts from their official codebase and utilize the first-think-then-action procedures. However, one of the major challenges is that we need to design search spaces for our tasks. For example, in Yao et al. (2022), the action space defined for the Hotpot QA dataset is SEARCH[entity], LOOKUP[entity], and FINISH. To do that, we design the following actions for Tic-Tac-Toe:

- (1)Defensive Action, which means to block the potential winning of your opponent (e.g., block your opponent from forming sequences of 3).
- (2)Offensive Action, which means to win the game (e.g., create forks, control the center, play ahead).

In terms of Texas Hold'em poker, we reuse the action presented in Appendix B.

For ToT, we follow the implementation of the text generation task. Specifically, follow the 2-step ToT manner, i.e., 1) generate plans; 2) vote for the plan; 3) generate action according to plan; 4) vote for action. The prompts used in this process are shown as follows:

FIRST GO Prompt:

You are the first to go. The legal positions are {legal\_positions}.  
First think about your current situation,  
then choose one move from legal positions to set up advantages.

Your output should be of the following format:

Thought:

Your thought.

Move:

My move is <CxRy> ('<' and '>' are mandatory), where x and y are  
the index of column and row, respectively.

Vote Prompt:

Conclude in the last line "The best choice is {s}",  
where s is the integer id of the choice.