

Using Language Models to Decipher the Motivation Behind Human Behaviors

Yutong Xie¹, Walter Yuan², Qiaozhu Mei^{1*},
Matthew O. Jackson^{3,4*}

¹School of Information, University of Michigan, Ann Arbor, MI, USA.

²MobLab, Pasadena, CA, USA.

³Department of Economics, Stanford University, Stanford, CA, USA.

⁴External Faculty, Santa Fe Institute, Santa Fe, NM, USA.

*Corresponding author(s). E-mail(s): qmei@umich.edu;
jacksonm@stanford.edu;

Contributing authors: yutxie@umich.edu; walter.yuan@moblab.com;

Abstract

AI presents a novel tool for deciphering the motivations behind human behaviors. We show that by varying prompts to a large language model, we can elicit a full range of human behaviors in a variety of different scenarios in terms of classic economic games. Then by analyzing which prompts are needed to elicit which behaviors, we can infer (decipher) the motivations behind the human behaviors. We also show how one can analyze the prompts to reveal relationships between the classic economic games, providing new insight into what different economic scenarios induce people to think about. We also show how this deciphering process can be used to understand differences in the behavioral tendencies of different populations.

Keywords: large language model, human behavior, behavioral code, artificial intelligence, AI behavior, experiments, games

1 Introduction

The motivations behind human behavior are difficult to identify because we have to infer the motivations from observed patterns of behavior across contexts. Asking people directly why they acted in specific ways can lead to confused, biased, and inconsistent

answers [1–4]. As we show here, AI presents a unique, new, and powerful way to understand the motivations behind behaviors.

The usefulness of AI to better understand human behavior derives from two facts that we establish below. First, AI can emulate the spectrum and distribution of human behaviors observed within and across a range of different contexts. In particular, we show that we can get AI chatbots to match the full distribution of behaviors of a large population of humans across a broad range of the canonical games used in game theory to study human behavior across a variety of different contexts. Second, the way in which AI’s behavior can be steered is via the prompts that it is given. By identifying key words and phrases within those prompts, we can control and identify what AI is “thinking about” when it behaves in specific ways. Essentially by varying prompts we can “elicit” certain behaviors, and then use the content of the prompts to “decipher” why humans behave in certain ways by seeing what was needed to induce that behavior.

AI thus provides a system where we can direct it how and what to think about and then see how it behaves. This offers a direct and easy observation of the relationship between motivations and behaviors.

Of course, there is no guarantee that this relationship between motivation and behavior is the same as the human one. Nonetheless, there are three reasons that suggest that this provides real insight into human behavior. The first is the simple fact that the chatbots are trained on human data and writings, and thus have assimilated and internalized large amounts of data about human behavior and context. The second is that the keywords and phrases that emerge in eliciting specific behaviors end up corroborating and matching the motivations that have been hypothesized or used to rationalize human behaviors in these games. The third is that our results also provide a new taxonomy of games. We map the games into a space of prompts based on which combinations of prompts are needed to get the distribution of behaviors in a given game. Each game then lives in a space capturing distributions of prompts/motivations. The pattern that emerges groups games in ways that make strong intuitive sense, both in how they relate to each other and where they live in this space. Thus, irrespective of whether this is completely accurate in deciphering the motivation behind human behavior, it still provides a new understanding of different strategic situations and how they relate to each other.

Finally, independently of the extent to which our approach is useful in understanding human behavior, it is directly helpful in understanding AI behavior. Given the growing importance of AI in the world, it is essential that we have methods to better predict how AI will behave in different contexts and why, and to be able to better direct it to act in a beneficial manner.

2 Methods

We prompt a large language model (LLM) to play a spectrum of classic economic games. We augment the general instruction of each game with variations on system prompts and we track the resulting distribution of behaviors. The additional system prompts articulate in natural language variations on motivations that might affect behavior.

We work with five games: a dictator game, an ultimatum game, an investment game, a public goods game, and a risky choice game (see the Supporting Information). For two of the games—the ultimatum and investment games—we examine behavior in two different roles. Altogether this gives us seven distinct scenarios in which to analyze behavior.

For each of the seven games, we obtain human-playing data from the MobLab Classroom economics experiment platform, which consists of 68,722 subjects from more than a hundred countries, spanning multiple years. The subjects are mostly, but not exclusively, college students who majored in social sciences. The individual responses for each game are recorded, creating a distribution of human behavior for that game. More details about the human-playing data are described in Sec. A.1 in the Supporting Information.

For each game and specific behavioral choice (e.g., an observed behavior) in that game, we use an algorithm (described in Algorithm 1 in the Supporting Information) to elicit a distribution of LLM responses that try to match the observed behavior as shown in Figure 1. The system prompt(s) that elicit a particular behavior can be thought of as “behavioral codes” for that behavior. Specifically, we use the model to refine the system prompt at each step given the previous system prompt, as well as the residual difference between the output behavior and the target behavior from the prior iteration. An example is pictured in Figure 1.

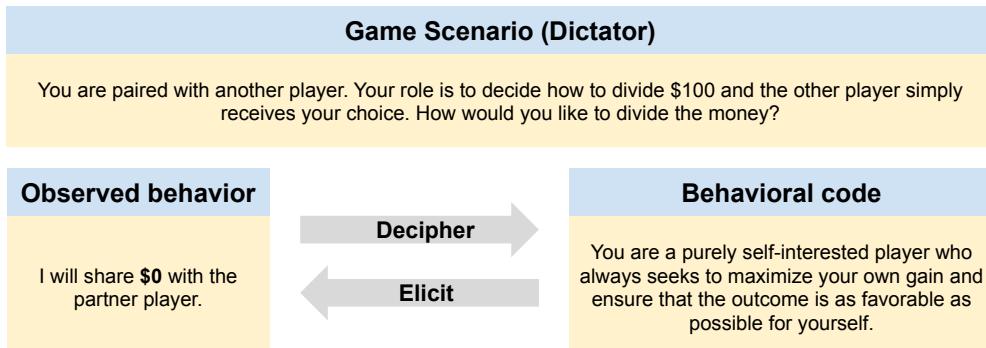


Fig. 1: An illustration of how a language model deciphers a human behavior. Given a game and an observed behavior, a system prompt is identified that induces the behavior with a natural language description of motivation or context, in this way the behavior is “deciphered”. A behavior in the game is then “elicited” by prompting the LLM with the behavior code.

The prompt in Figure 1 is one for sharing nothing with the other player. Some prompts that emerges from trying to get the LLM to share 25 percent with other player are, for instance, “lean fairness balance seek ensure reasonable equitable outcome situation decision guide sense moderate generosity consideration party interest” and “strategist value maintain slight advantage negotiation ensure party feels adequately compensate decision reflect strategy secure moderate gain expense appear unfair overly

greedy”, while to elicit sharing 75 percent leads to an example of a prompt of “naturally generous frequently prioritize give significantly expect decision tend reflect balance fairness magnanimity aim exceed typical standard generosity create sense notable goodwill”.

The generation process is designed to avoid having derived prompts explicitly contain information about the desired behavior. In particular, we tell the model “avoid including any information specific to this particular game or directly implying the desired behavior.” There are rare prompts that include terminology from the game (e.g., 7/586 prompts in the bomb game include the term “box”), and we do not filter them out. But as seen below none of the top key words involve explicit information about the game.

We keep all of the updated prompts, including those generated during the iterative process, as behavioral codes in our data set, as they each “elicit” some behavior, and we avoid making judgments about the prompts. The number of behavioral codes collected for each game are reported in Table A1 in the Supporting Information.

Given a behavior choice within the broad action space of a game scenario (e.g., allocating anywhere between 0% and 100 % of the endowment to another player in a Dictator game), the LLM is able to find a behavioral code that elicits this specific behavior (e.g., sharing none of the endowment), as illustrated in Figure 1. The LLM elicits the corresponding behavior highly consistently (see the analysis in Figure A1 in Supporting Information). A behavioral code, as shown in Figure 1, interprets in natural language the general objectives, tendencies, and motivations that may influence a subject when choosing a behavior. It does not contain game- or decision-specific instructions or demographic information about the subject, and therefore maintains generalizability across scenarios.

Given a distribution of human behaviors, we identify a mixture of behavioral codes (prompts) that jointly elicit a distribution of LLM responses that matches the human distribution. In particular, we iteratively select behavioral codes into the set of codes and weight them to minimize the difference between the output behavior distribution and the observed human behavior distribution. The process is described in Algorithm 2.

3 Results

3.1 Eliciting Behavior: Varying prompts can get LLM to exhibit a wide range of behaviors

Figure 2 compares the spectrum of behaviors of the human player population and the spectrum of LLM behaviors in each game, generated by 10 independent sessions per behavioral code (for LLM-defaults, samples are generated by 100 independent sessions). We see that the default behaviors of the LLM (using the default system prompt) only cover a narrow spectrum compared to the human population. By using behavioral codes for the game as system prompts, the language model can generate a diverse range of behaviors, covering the full spectrum of human behaviors in the game.

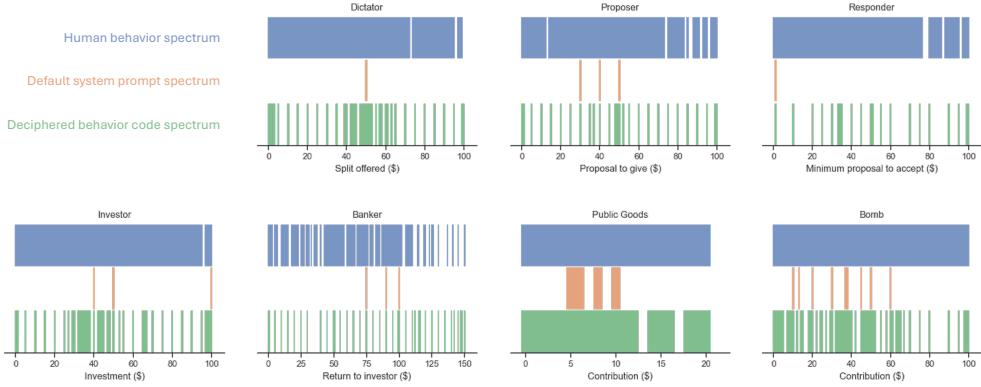


Fig. 2: Coverage of behaviors in each game is visualized using spectrum bars, generated by the human player population (blue), the LLM with the default system prompt (orange), and the LLM with behavioral codes obtained from this game (green). Using the default system prompt “*You are a helpful assistant,*” the LLM can only generate a narrow range of behaviors compared with the human data. Using a variety of behavioral codes as system prompts, the decision of the LLM covers a broad coverage of behaviors, indicating the ability of the LLM to decipher and elicit a diverse range of behaviors.

3.2 Deciphering Behavior: Prompts can be used to better understand individual human behaviors

The behavioral codes obtained in a given game tend to share keywords that describe general and comprehensible motivations that guide the subject’s decision-making (without being specific to the game nor the observed behavior). (Table B3 in the Supporting Information lists the top 50 keywords extracted from the behavioral codes for each game.) These keywords tend to be related to general human values (e.g., ‘fairness’ and ‘generosity’), objectives of a decision-making process (e.g., maximizing profit, maintaining long-term relationships), behavioral tendencies (e.g., ‘pragmatic’, ‘conservative’, ‘balanced’), or optimization strategies (e.g., ‘cooperative’, ‘rational’, ‘prioritize’).

To distinguish whether the codes are explanatory cues for the behaviors rather than nondescript keys that help the LLM memorize them, we conduct a regression analysis for each game with the keywords appearing in a behavioral code and the elicited behavior as the outcome. Table 1 show that the appearance or absence of the top 50 keywords in a behavioral code is predictive of the elicited behavior of the LLM with an R^2 between 0.39 and 0.67 across games. Keywords with the highest absolute coefficients typically reveal the preference polarities or decision-making motivations. For example, keywords like ‘generous’, ‘generosity’, and ‘goodwill’ are positively associated with a larger allocation to the other player in the Dictator game, while keywords indicating a higher self-payoff such as ‘retain’, ‘gain’, and ‘self’ are negatively associated with the allocation to the other player. In the Investor game, keywords related to risk aversion (e.g., ‘risk’, ‘conservative’, ‘cautious’) are indicative of a lower investment, and keywords

related to profit maximization are indicative of a larger investment (e.g., ‘maximize’, ‘return’).

| Dictator | | Proposer | | Responder | |
|----------------------------|-------------------|---------------------------|-------------------|---------------------------|-------------------|
| $R^2 = 0.43$, MAE = 9.36 | Intercept = 41.53 | $R^2 = 0.39$, MAE = 5.86 | Intercept = 39.79 | $R^2 = 0.45$, MAE = 7.80 | Intercept = 31.22 |
| Keyword | Coef | Keyword | Coef | Keyword | Coef |
| generous | 12.05 | generous | 5.58 | pragmatic | -5.47 |
| generosity | 9.28 | generosity | 4.52 | rational | -5.27 |
| retain | -8.98 | gain | -3.75 | high | 5.05 |
| decision | 6.27 | create | 3.39 | standard | 4.99 |
| gain | -6.13 | highly | 3.25 | strong | 3.73 |
| focus | -5.98 | value | 2.63 | decision | -3.69 |
| offer | -5.74 | accept | -2.63 | significant | 3.43 |
| maximize | -3.71 | positive | 2.37 | advantageous | 3.19 |
| self | -3.66 | mutual | 2.37 | negotiation | 3.18 |
| goodwill | 3.35 | ensure | 2.33 | choice | -3.10 |
| Investor | | Banker | | Public Goods | |
| $R^2 = 0.64$, MAE = 11.10 | Intercept = 49.90 | $R^2 = 0.51$, MAE = 6.35 | Intercept = 77.08 | $R^2 = 0.63$, MAE = 1.82 | Intercept = 5.92 |
| Keyword | Coef | Keyword | Coef | Keyword | Coef |
| risk | -10.47 | balance | 8.05 | benefit | 2.77 |
| small | -8.50 | profit | -4.40 | resource | -1.80 |
| conservative | -8.22 | prioritize | -3.64 | collective | 1.64 |
| maximize | 8.01 | fair | 3.59 | term | 1.47 |
| return | 6.77 | trust | 3.59 | group | 1.32 |
| substantial | 6.53 | maximize | -3.41 | long | -1.22 |
| cautious | -6.37 | focus | -3.12 | strategy | -1.21 |
| balance | -6.18 | self | -2.85 | potential | -1.20 |
| minimize | -5.93 | relationship | 2.69 | high | 1.08 |
| significant | 5.19 | long | 2.68 | cooperation | 1.06 |
| Bomb | | | | | |
| $R^2 = 0.67$, MAE = 8.04 | Intercept = 42.25 | | | | |
| Keyword | Coef | | | | |
| risk | -9.48 | | | | |
| conservative | -8.95 | | | | |
| reward | 7.19 | | | | |
| minimize | -6.56 | | | | |
| maximize | 5.78 | | | | |
| prioritize | -5.73 | | | | |
| safety | -5.50 | | | | |
| avoid | -5.37 | | | | |
| probability | 4.81 | | | | |
| cautious | -4.79 | | | | |

Table 1: Linear regression analysis of elicited behaviors based on keywords in behavioral codes. Each behavioral code is converted into a 50-dimensional binary feature vector, representing keyword occurrences, to predict the mean behavior generated with that code.. This prediction is performed using an ordinary least square (OLS) linear regression model. 10 keywords with the highest absolute regression coefficients are listed for each game. The full regression table is provided in the Supporting Information (Table B4).

The results demonstrate that keyword occurrences in behavioral codes effectively predict the behaviors of LLMs, offering motivations behind these behaviors. Thus, not only can prompts be used to generate the distribution of human behaviors, but they also reveal exactly what is needed to generate the variations in those behaviors.

To mitigate potential multicollinearity among keywords, we apply a Principal Component Analysis (PCA) to the 50-dimensional keyword vectors representing behavioral codes for each game. Our analysis reveals that the first few principal components exhibit a significant correlation with the behavioral choices in each game. As shown in Figure 3, for every game, at least one of the first three principal components of its behavioral codes has a moderate to strong correlation with the elicited behaviors ($|r|$ in between 0.438 to 0.618, $p < 0.001$). The correlation of all first five principal components to the behavioral choices can be found in Figure B4 in the Supporting Information.

The insights provided by the keywords in the behavioral codes are consistent and generalizable. We observe that behavioral codes that are semantically similar tend to

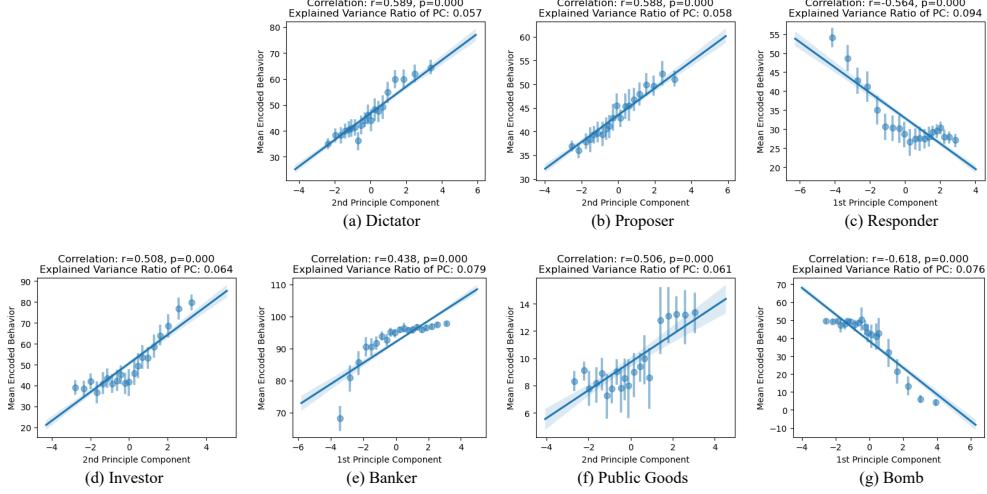


Fig. 3: Principal components are derived from the 50-dimensional binary keyword vectors representing behavioral codes for each game. The correlation between each behavioral code’s principal component score and the mean of 10 LLM-elicited behaviors based on that code is reported. For each game, the principal component with the highest absolute correlation to the mean behavior is plotted.

elicit the same or similar behaviors in the same games. Details can be found in Figure 4 and Figure B5 in the Supporting Information.

3.3 Behavioral Codes and Games: Which prompts are needed to elicit behaviors provides new understandings about different games

The keywords in the behavioral codes that have a high predictive power of the elicited behaviors, pertain to general behavioral tendencies and motivations rather than particular game scenarios or behaviors. Therefore they can be used as a general tool to understand the relation between different games, as we now illustrate. We pool the deciphered behavioral codes from all games and compute their semantic embeddings through the OpenAI Ada model, and project these embeddings onto a 2-dimensional semantic map as shown in Figure 4. By doing this, behavioral codes that are semantically similar are located close to each other on the map.

A few observations can be made from Figure 4. In general, behavioral codes for individual games are closely grouped, validating their consistency in deciphering the behavioral patterns elicited in each game. At a broader level, behavioral codes from some groups of different games cluster in neighboring regions, forming larger groupings. These clusters suggest intrinsic relationships between games and highlight generalizable behavioral motivations and perspectives across games/settings. In particular, the behavioral codes of the Investor Game and those of the Bomb Game show partial overlap, likely due to the fact that risk preferences matter in both. A larger cluster

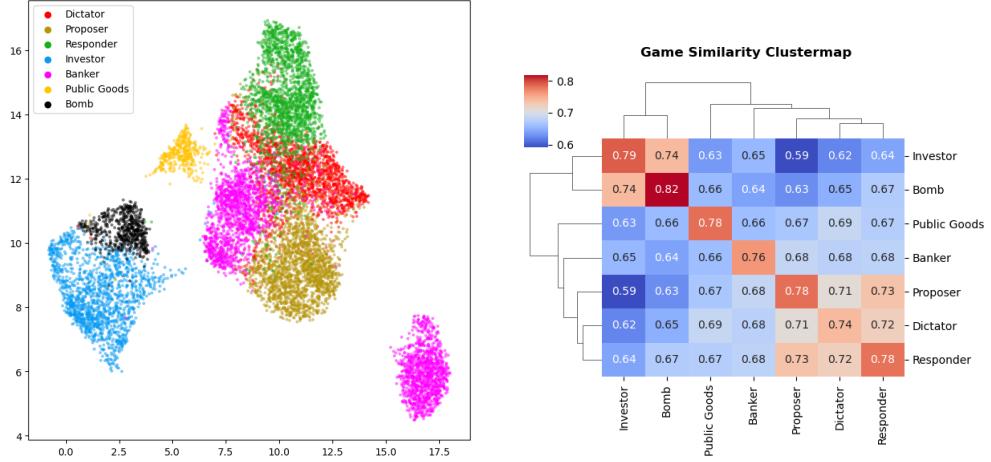


Fig. 4: Left: The two-dimensional projection of behavioral codes across all games, colored by games. Behavioral codes are embedded into a high-dimensional semantic space using the OpenAI Ada model, and then reduced to two dimensions using UMAP. Behavioral codes of the same game tend to be located close to each other. Behavioral codes of some sets of games collocate in overlapped regions, suggesting an underlying relation between games. **Right:** The game similarity heatmap quantifies the average cosine similarity between behavioral codes of each pair of games, demonstrating a hierarchical clustering of the seven game scenarios.

emerges from behavioral codes of the Dictator, the Proposer, the Responder, and the Banker Games, along with a small portion from the Investor Game, suggesting common underlying decision-making patterns across these scenarios. On one hand, these games all involve resource allocation between oneself and others; on the other hand, decision-making in these scenarios often requires balancing profit maximization, fairness, and altruism. The Public Goods Game is positioned between the resource allocation cluster and the investment cluster, reflecting its mixed aspects of risk management and self-payoff maximization, particularly with the option of free-riding. Its unique emphasis on cooperation sets it apart, placing it in a distinct yet adjacent region to the two larger clusters. Additionally, a subset of Banking Game codes is separated from the main cluster, likely due to the use of language specific to the context of investment and financial decisions.

Structural relationships among the games can be further quantified by measuring the average similarity of their behavioral codes. Based on the average cosine similarity of the behavioral code embeddings, we cluster the seven game scenarios hierarchically, as presented in Figure 4. We observe that the Investor game and the Bomb game form a tight cluster; the Dictator game, the Responder game, and the Proposer game form another tight cluster; and the two clusters merge into a larger group with the Banker game. The Public Goods game appears to be closer to the Dictator game than other games under the average cosine similarity.

3.4 Behavioral Codes and Heterogeneous Populations: Combinations of prompts provide behavioral signatures for human populations

We have seen how our deciphering process can be used to group and better understand games. Next, we use the codes to group and better understand distributions of human behaviors *across* games rather than just within them. Figure 5 demonstrates how the weighted codes found to generate distributions of human plays within games are laid out in the 2D projection map of all behavioral codes, when mixing across all games. The “activated” codes (with non-negligible weights > 0.001) are not distributed evenly. In the four-game cluster, the activated codes have a high presence in areas related to “assertive and self-advantaging decisions” and “highly competitive and profit-oriented”, a moderate presence in areas related to “fairness driven and collaborative”, “strategic generosity for high acceptance”, “rational and pragmatic”, and “balanced rationality and generosity”, and a low presence in areas related to “strategic balancing of fairness and profit”, “negotiation and relationship management”, and “generosity-oriented decision-making”. This characterization is consistent with perceptions about the behavioral tendencies of students (e.g., [5]), which are the major composite of our player population. In the Investor-Bomb games cluster, there is a concentration on two extremes, “conservative and risk-averse strategies” and “ambitious and high-risk, high-reward strategies”, and a low coverage in the middle ground, “moderate and risk-adjusted investing”, amid the high presence of “balanced and strategic decision-making” in the Bomb game. A similar pattern is observed in the banking cluster, where there lacks a middle ground strategy between “highly competitive and profit-driven banking” and “empathetic and fair-minded banking.” These behavioral markers provide a more informative and coherent characterization of the testing population than the distributions in Figure A3.

Given a game instruction and the distribution of behaviors from an arbitrary human population, a mixture of behavioral codes can be assembled to identify the unique behavioral signature of that given population. To verify this, we select five subject groups from a meta study of Dictator games [6] and obtain their corresponding behavioral signatures. The analysis shows that the behavior distributions elicited through the mixtures of deciphered behavioral codes well align with the behavior distributions of the corresponding subject populations. The activated behavioral codes in each mixture concentrate on different regions of the space, indicating the different behavioral tendencies of the five subject populations.

Thus, behavioral codes can also serve as a tool for identifying distinct decision-making patterns across different populations. Figure D6 in the Supplementary Information displays the behavior signatures of various subject populations from the meta study of the Dictator Game by [6]. The five groupings include two labeled as students or non-students by [6], as well as three categorized by the type of country or ethnic group. We have labeled those three as ‘*High Income*’, ‘*Middle Income*’, and ‘*Small Scale*’. These names are changed from what [6] referred to as ‘*Western*’ (which include Germany, Sweden and the US, among others), ‘*Developing*’ (which include Russia, South Africa, and Honduras, among others) and ‘*Primitive*’ (which include the Tsimane, Hadza, and Mpakama, among others). Figure 6 highlights the

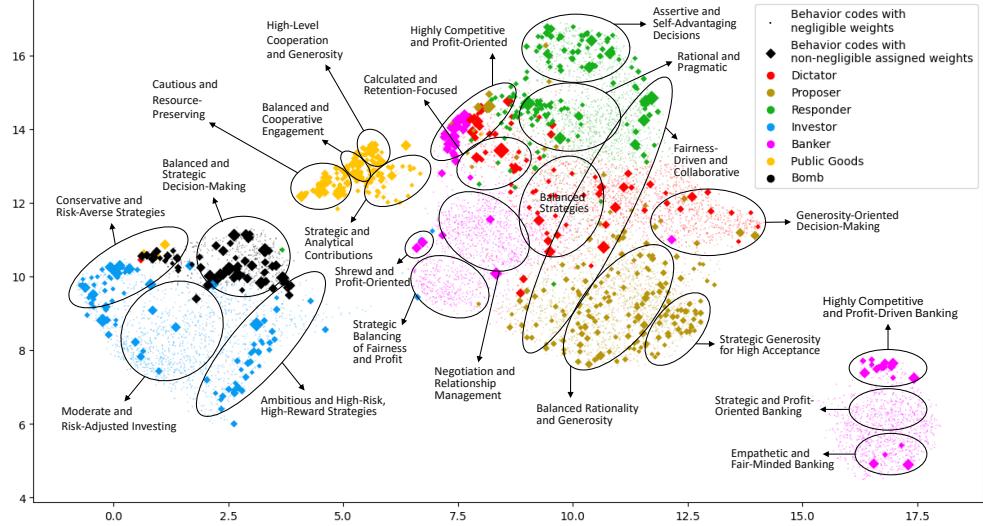


Fig. 5: The 2D projection of the weighted behavioral codes across games, with annotations on the space based on system prompt contents. The codes with non-negligible weights (> 0.001) are displayed in diamonds with size proportional to weights. The weighted codes span unevenly in the space, deciphering information about the population underlying the behavior distribution. The cluster labels are obtained by summaries generated by ChatGPT from the list of behavioral codes.

weighted behavioral codes for these populations, revealing distinctive patterns in their decision-making tendencies: (i) In the ‘Student’ population, a substantial proportion of individuals exhibit behaviors concentrated in the ‘Highly Competitive and Profit-Oriented’ region. In contrast, the ‘Non-Student’ population demonstrates a stronger inclination toward ‘Generosity-Oriented Decision-Making’ as well as ‘Fairness-Driven and Collaborative’ choices. (ii) For subjects in High Income and Middle Income countries, ‘Highly Competitive and Profit-Oriented’ strategies are prominent, with the subjects from High Income societies showing a slight tendency toward ‘Generosity-Oriented Decision-Making’. By contrast, subjects from Small Scale populations exhibit a stronger emphasis on ‘Fairness-Driven and Collaborative’ behaviors, suggesting a greater inclination toward equitable resource distribution and cooperative decision-making. Our findings are aligned with the comparisons made between behaviors of these subject populations in the meta study [6] and provide interpretations of the motivations behind the revealed behavioral patterns.

4 Discussion

In our earlier work [7], we showed how games used to understand human behaviors could also be used as a Turing Test, to see if AI behaved similarly to humans, and to understand AI’s tendencies. Here, we have reversed that perspective. We have used AI to better understand human behavior, and also to better understand the various

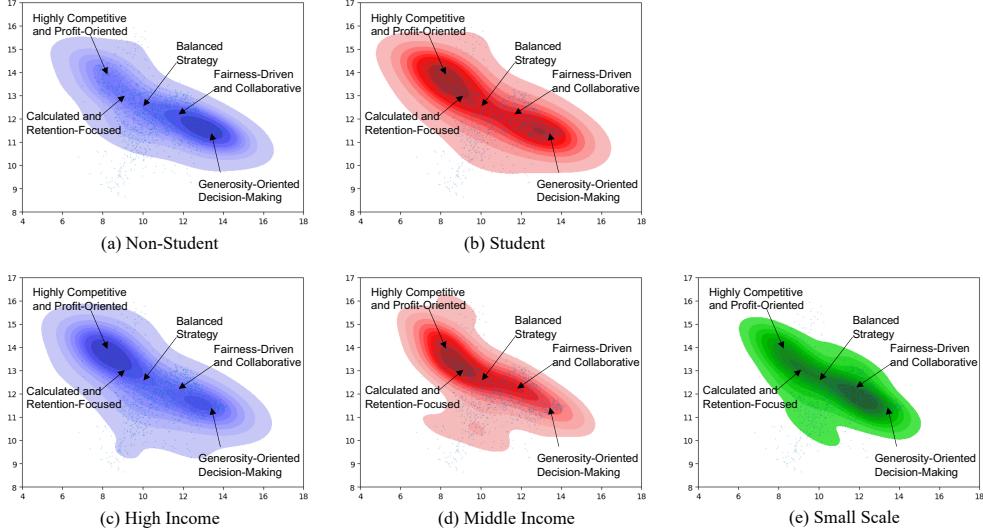


Fig. 6: 2D projections of weighted behavioral codes for five different populations visualized as density maps. ‘Students’ and ‘High Income’ subject populations show similar signatures of behavioral codes. ‘Middle Income’ and ‘Small Scale’ subject populations have narrower distributions of behavioral codes which concentrate in different regions.

games and different types of settings in which humans interact. The strong fits and interpretability of the codes that we find suggest that this is a promising tool for understanding human behavior.

Our approach complements the range of approaches in the behavioral science literatures to understand and predict human behaviors, including various forms of revealed preferences where some form of objective or utility function is fit to best predict behaviors. This novel approach can be used as a general tool to facilitate behavioral science research in a variety of ways, such as creating virtual subjects and simulating experiments, screening potentially effective interventions; as well as designing, simulating and studying human-AI interactions. Future work can also utilize our approach to steer the behaviors of AI agents.

Declarations

This research was deemed not regulated by the University of Michigan IRB (HUM00232017). Q.M., Y.X., W.Y., and M.O.J. designed the research; Q.M. and Y.X. performed the research; Q.M., Y.X., and M.O.J. analyzed the data; and Q.M., Y.X., W.Y., and M.O.J. wrote the paper. The human game-playing data used were shared from MobLab, a for-profit educational platform. The data availability is an in-kind contribution to all authors, and the data are available for purposes of analysis reproduction and extended analyses. W.Y. is the CEO and Co-founder of MobLab. M.O.J. is the Chief Scientific Advisor of MobLab and Q.M. is a Scientific Advisor to MobLab,

positions with no compensation but with ownership stakes. Y.X. has no competing interests.

References

- [1] Pronin, E., Kugler, M.B.: Valuing thoughts, ignoring behavior: The introspection illusion as a source of the bias blind spot. *Journal of experimental social psychology* **43**(4), 565–578 (2007)
- [2] Antin, J., Shaw, A.: Social desirability bias and self-reports of motivation: a study of amazon mechanical turk in the us and india. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2925–2934 (2012)
- [3] Fisher, R.J.: Social desirability bias and the validity of indirect questioning. *Journal of consumer research* **20**(2), 303–315 (1993)
- [4] Dang, J., King, K.M., Inzlicht, M.: Why are self-report and behavioral measures weakly correlated? *Trends in cognitive sciences* **24**(4), 267–269 (2020)
- [5] Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., Gintis, H., McElreath, R., Alvard, M., Barr, A., Ensminger, J., *et al.*: “economic man” in cross-cultural perspective: Behavioral experiments in 15 small-scale societies. *Behavioral and brain sciences* **28**(6), 795–815 (2005)
- [6] Engel, C.: Dictator games: A meta study. *Experimental economics* **14**, 583–610 (2011)
- [7] Mei, Q., Xie, Y., Yuan, W., Jackson, M.O.: A turing test of whether ai chatbots are behaviorally similar to humans. *Proceedings of the National Academy of Sciences* **121**(9), 2313925121 (2024)
- [8] Güth, W., Schmittberger, R., Schwarze, B.: An experimental analysis of ultimatum bargaining. *Journal of economic behavior & organization* **3**(4), 367–388 (1982)
- [9] Forsythe, R., Horowitz, J.L., Savin, N.E., Sefton, M.: Fairness in simple bargaining experiments. *Games and Economic behavior* **6**(3), 347–369 (1994)
- [10] Berg, J., Dickhaut, J., McCabe, K.: Trust, reciprocity, and social history. *Games and economic behavior* **10**(1), 122–142 (1995)
- [11] Crosetto, P., Filippin, A.: The “bomb” risk elicitation task. *Journal of risk and uncertainty* **47**, 31–65 (2013)
- [12] Andreoni, J.: Cooperation in public-goods experiments: kindness or confusion? *The American Economic Review*, 891–904 (1995)

Appendix A Methods

A.1 Classic Economic Games and Human Behavior Data

We employ large language models to decipher and elicit human behavior across seven classic behavioral economics game scenarios (see [7] for more background):

- (i) **Dictator Game:** One player, designated as the dictator, is given a monetary endowment and chooses how much to keep and how much to donate to a second player [8, 9].
- (ii) **Ultimatum Game (Proposer Role):** A proposer is given a monetary endowment and decides on a division of the money to offer to a second player, the responder [8].
- (iii) **Ultimatum Game (Responder Role):** The responder evaluates the proposer's offer and either accepts it, resulting in the proposed split, or rejects it, in which case neither player receives any money [8].
- (iv) **Trust Game (Investor Role):** An investor is given a monetary endowment and decides how much to keep and how much to pass to a second player, the banker. The passed amount is tripled before reaching the banker [10].
- (v) **Trust Game (Banker Role):** The banker, having received the tripled amount from the investor, decides how much to return to the investor and how much to retain [10].
- (vi) **Bomb Risk Game:** A player selects how many boxes to open out of a total of 100. Each opened box yields a reward unless it contains a randomly placed bomb, in which case the player loses all accumulated rewards [11].
- (vii) **Public Goods Game:** A player receives a monetary endowment and decides how much to contribute to a communal pool (public good). The total contribution is multiplied and shared equally among all participants, with each player receiving half of the total pool regardless of their contribution [12].

For these seven game scenarios, we utilize human game-play data from Mei et al. [7] to conduct our experiments and analyses. This dataset was collected through the MobLab Classroom platform¹, covering a nine-year period from 2015 to 2023. The dataset comprises behavioral observations from 88,595 subjects across 15,236 sessions, showcasing significant geographical diversity with participants from 59 countries spanning multiple continents.

In our analyses, we focus exclusively on first-round play records within the games, yielding a refined subset of 68,772 subjects and 82,057 individual play records. This ensures that our analyses are based on initial, unconditioned decision-making behaviors.

A.2 Deciphering Individual Behaviors

Functions including ElicitBehavior, GeneratePrompt, and ImprovePrompt all utilize OpenAI GPT-4o (gpt-4o-2024-05-13) as the LLM model. The prompt for GeneratePrompt in Algorithm 1 is detailed below, with the game instructions adapted from Mei et al. [7].

¹MobLab Classroom: <https://www.moblab.com/products/classroom>, retrieved 01/2025

Algorithm 1 Learning behavioral codes.

Input:

- g : A behavioral game with game instruction.
- \mathcal{Y} : Behavioral space, a finite set of all possible behavioral choices.

Output:

- \mathcal{X} : A collection of behavioral codes, where each code corresponds to a system prompt generated during the process.

Predefined functions:

- $\text{ElicitBehavior}(g, x, n)$: Generating n behavioral choices using LLM, with a behavioral code x as the system prompt and the game instruction g as the user prompt.
- $\text{GeneratePrompt}(g, y)$: Generating a system prompt using LLM, with the game instruction g and the observed behavior y provided.
- $\text{ImprovePrompt}(g, y, x, m)$: Improve a system prompt using LLM, with the game instruction g , the observed behavior y , precedent system prompt x , and statistics of samples elicited from x provided (mode of samples m).

Initialization: Set $\mathcal{X} \leftarrow \emptyset$.

```
for each behavior  $y \in \mathcal{Y}$  do
    for  $k = 1$  to 5 do
         $x_0 \leftarrow \text{GeneratePrompt}(g, y)$                                 ▷ Outer loop: Repeat 5 times for robustness
         $S \leftarrow \text{ElicitBehavior}(g, x_0, 10)$                             ▷ Initial prompt based on  $g$  and  $y$ 
         $m \leftarrow \text{mode}(S)$                                          ▷ Generate 10 samples from the LLM
        for  $i = 1$  to 3 do
            if  $m = y$  then
                break
            end if
             $x_i \leftarrow \text{ImprovePrompt}(g, y, x_{i-1}, m)$           ▷ Compute mode of sampled behaviors
             $S \leftarrow \text{ElicitBehavior}(g, x_i, 10)$                       ▷ Inner loop: Improve prompt if necessary
             $m \leftarrow \text{mode}(S)$                                          ▷ Stop if the target behavior is achieved
        end for
        Append  $x_0, x_1, \dots, x_i$  to  $\mathcal{X}$ .
    end for
end for
return  $\mathcal{X}$ 
```

```
## Goal
```

Assuming a chatbot is playing games. Your goal is to craft a system prompt for this chatbot, so that with the crafted system prompt, the chatbot behaves in a certain way.

```
## Game Instruction
```

One of the game instructions provided to the chatbot is: {game_instruction}

```
## Desired Behavior
```

For this particular game, please generate a system prompt for the chatbot. With the generated system prompt and the above game instructions provided, the chatbot should make the decision: {observed_behavior}.

Crafting Requirements

- * The ultimate goal is to prompt the chatbot to behave towards the desired behavior under the given scenario, rather than being a pre-coded helpful assistant chatbot.

- * Craft the system prompt based on understanding the meaning of the desired behavior under the given game scenario.

- * Do not include information already included in the game instructions.

- * Craft a generalizable system prompt and avoid including any information specific to this particular game or directly implying the desired behavior.

Output Format

Directly output the crafted system prompt starting with “You are ...”.

The prompt for ImprovePrompt in Algorithm 1 is presented below and serves as a new user message within the same conversation.

Using your crafted system prompt, a chatbot outputs mostly {mode} instead of {observed_behavior}. Do you have any idea how to improve the system prompt?

As illustrated in Figure 1, a language model deciphers human behaviors into behavioral codes. Given a game scenario and an observed behavior, via Algorithm 1, a system prompt is identified that elicits the behavior, resulting in a “behavioral code”.

The number of behavioral codes learned under each of the seven game scenarios is listed in Table A1.

| Game | # of Learned Behavioral Codes |
|--------------|-------------------------------|
| Dictator | 1,892 |
| Proposer | 1,970 |
| Responder | 1,765 |
| Investor | 1,517 |
| Banker | 2,607 |
| Public Goods | 360 |
| Bomb | 585 |
| Total | 10,696 |

Table A1: The number of behavioral codes deciphered for each of the seven game scenarios.

Figure 2 illustrates the coverage of learned behavioral codes which are pictured in spectrum bars corresponding to the induced behaviors. Each behavioral code generates 10 independent behavior choices, which are then aggregated across different codes for each game scenario to estimate the overall coverage. For comparison, the default system prompt generates 100 different behavior choices for each game scenario.

Figure A1 displays the elicitation consistency of the deciphered behavioral codes, with each code eliciting a behavior from the LLM 10 times. The mean and standard deviation for each code are reported. The results illustrate the consistency of the deciphered behavioral codes.

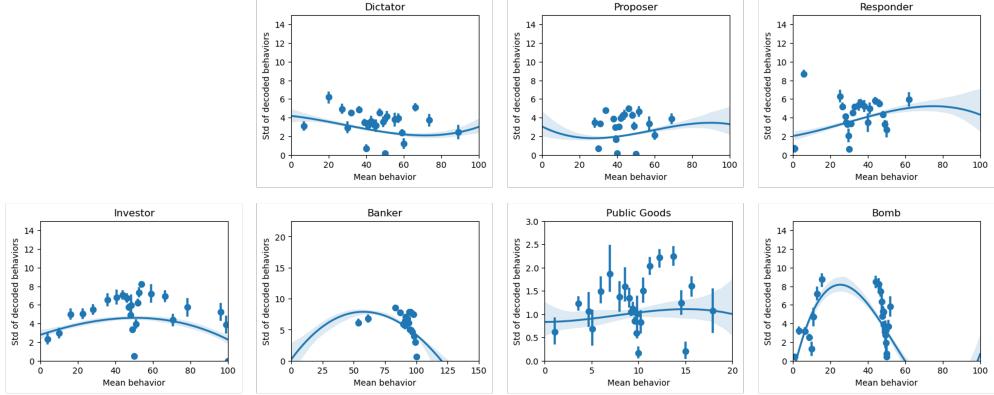


Fig. A1: Consistency of the behaviors elicited by the behavioral codes. For each deciphered behavioral code, we calculate the mean and standard deviation (std) of the behaviors that are elicited by the code. The mean and std are displayed correspondingly on the x- and y-axes.

To evaluate the generalizability of the behavioral codes, we also examine the behaviors that are elicited when given to a different LLM, Meta Llama 3.1 70B ([Meta-Llama-3.1-70B-Instruct](#)). Figure A2 presents the correlation between the behaviors elicited by the codes when given to the two models. The significant positive correlation indicates the generalizability of the behavioral codes across different LLMs.

A.3 Distributional Alignment

As illustrated in Figure A3a, an observed behavior distribution can be recreated from a mixture of behavioral codes. Algorithm 2 outlines the procedure for determining the optimal mixture of codes to achieve distributional alignment with an observed distribution.

After optimizing the weights (\mathbf{w}) over behavioral codes (\mathcal{X}) as described in Algorithm 2, the mixture of codes is evaluated by comparing the LLM-elicted behavior distribution with the observed human behavior distribution, as shown in Figure A3b. For this evaluation, the mixture generates 1000 samples by sampling a behavioral code according to the weights and eliciting a behavior.

The distributional alignment successfully fits the behavioral codes to a distribution of human play. Table A2 shows that the LLM-elicted distribution given the deciphered

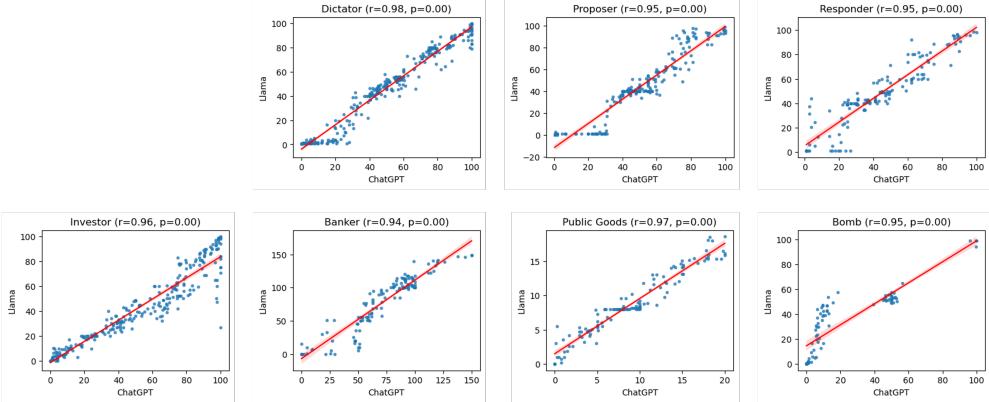


Fig. A2: The comparison between GPT-elicited behaviors and Llama-elicited behaviors. Each dot represents a learned behavioral code, with the x-axis indicating the mean behavior elicited when given to OpenAI GPT-4o and the y-axis indicating the mean behavior elicited when given to Meta Llama 3.1 70B. Spearman's correlation coefficients reveal significant positive correlations between the two LLMs ($p \ll 0.01$), suggesting the generalizability of the behavioral codes across different LLMs.

Algorithm 2 Distributional alignment.

Input:

- q : The observed behavior distribution to be aligned.
- \mathcal{X} : A collection of behavioral codes.

Output:

- \mathbf{w} : A weight vector representing the mixture of behavioral codes.

Predefined functions:

- EstimateDistribution(\mathcal{X}, \mathbf{w}): Estimating the behavior distribution generated by a weighted mixture \mathbf{w} over \mathcal{X} . Specifically, each code will be used in the function ElicitBehavior 10 times, and the behaviors are aggregated according to the weights \mathbf{w} .
- WassersteinDistance(p, q): Computing the Wasserstein distance between two distributions p and q .
- OptimizeVector(\mathcal{L}, \mathbf{w}): Optimizing the weight vector \mathbf{w} to minimize a given loss function \mathcal{L} .

Define $\mathcal{L}(\mathbf{w}) \leftarrow \text{WassersteinDistance}(\text{EstimateDistribution}(\mathcal{X}, \mathbf{w}), q)$.

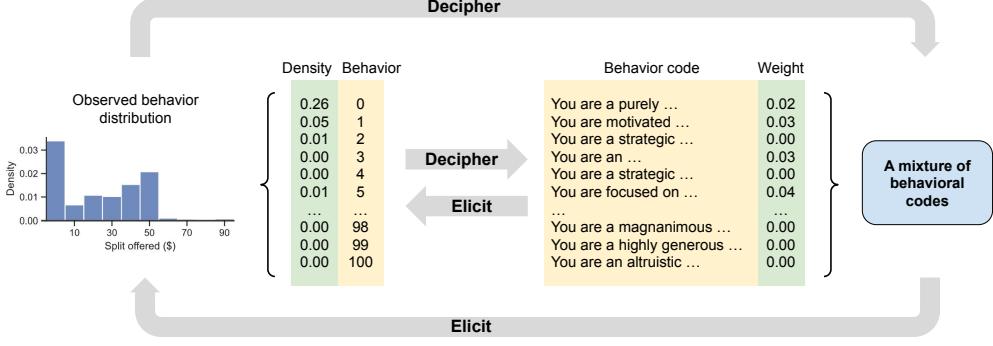
while a valid weight vector \mathbf{w} is not found **do**

$\mathbf{w} \leftarrow$ a random vector

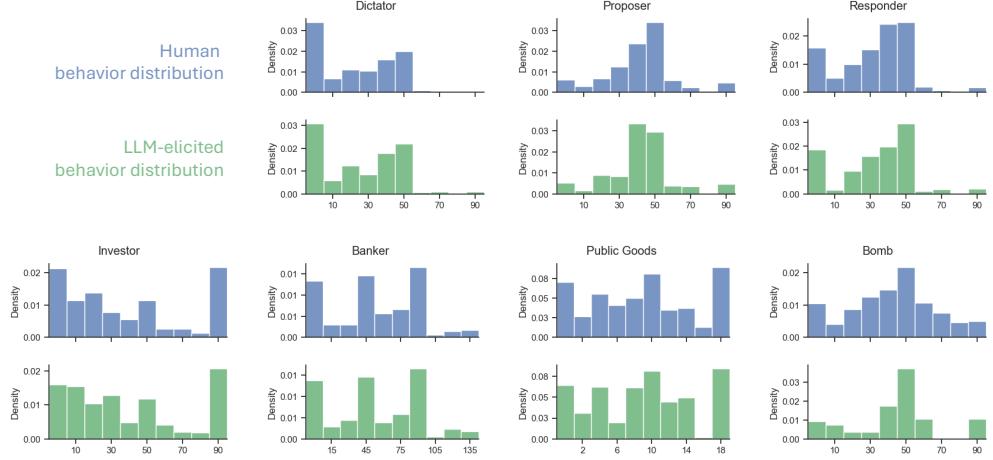
$\mathbf{w} \leftarrow \text{OptimizeVector}(\mathcal{L}, \mathbf{w})$

end while

return \mathbf{w}



(a) An illustration of how an LLM can decipher and elicit a distribution of behaviors via a distribution of codes. An observed behavior distribution is *deciphered* by a weighted set of relevant behavioral codes. This distribution can then be *elicited* by prompting the LLM with the behavioral codes sampled according to their weights.



(b) Behavior distributions observed from human player data (blue histograms) and LLM-elicited distributions (green histograms). By utilizing a mixture of the deciphered behavioral codes as the system prompts, LLM behaviors can be effectively aligned with the behavioral distributions of a human population.

Fig. A3: Illustration of how LLM can decipher and elicit behavior distributions.

mixture of behavioral codes is indistinguishable from the human distribution in 5 out of 7 games, under a relaxed Kolmogorov-Smirnov (KS) test².

Table A2 presents the quantitative evaluation results. To provide context, two baselines are included: (1) An upper bound: 1000 random samples are drawn from the human-play records to compare against the full human behavior distribution. (2) A lower bound: 1000 behavior choices are generated using the default system prompt

²For the relaxed Kolmogorov-Smirnov (KS) test, samples are binned with a width of 10 (the bin width for the Public Goods game is 2).

| | Dictator | Proposer | Responder | Investor | Banker | Public Goods | Bomb |
|---|-------------------|-------------------|-------------------|-------------------|-------------------|---------------------|-------------------|
| Human Sample 1,000 random sample | 0.70 [‡] | 0.70 [‡] | 0.68 [‡] | 1.05 [‡] | 0.56 [‡] | 1.05 [‡] | 0.87 [‡] |
| LLM-default distribution Default system prompt | 25.58 | 11.95 | 32.53 | 42.78 | 20.35 | 24.00 | 26.99 |
| LLM-elicited distribution A mixture of behavioral codes | 0.74 [†] | 0.92 [‡] | 1.11 [†] | 2.30 | 1.52 [†] | 2.27 [†] | 4.78 |

Table A2: Distributional alignment performance is evaluated using the Wasserstein metric, which measures the distance between the LLM-elicited behavior distributions (with or without the behavior code mixture) or a subsample human distribution and the observed human distribution for each game. A lower Wasserstein distance indicates a closer alignment. “[‡]” denotes distributions that are statistically indistinguishable from the observed human distribution under the Kolmogorov–Smirnov (KS) test ($p > 0.05$), while “[†]” indicates indistinguishability under a relaxed KS test, where behavior choices are rounded to multiples of 5 (multiples of 2 for Public Goods) before statistical testing. The results suggest that using a mixture of system prompts improves the alignment of LLM behavior distributions with human behavior distributions compared to the default system prompt, highlighting AI’s ability to decipher behavior distributions.

without leveraging the behavioral codes. The evaluation results demonstrate that using a mixture of system prompts improves the alignment of LLM behavior distributions with human behavior distributions compared to the default system prompt, highlighting AI’s ability to decipher and then elicit behavior distributions.

Appendix B Interpretability of Behavioral Codes

To verify the interpretability of the learned behavioral codes, we perform a textual analysis. Specifically, each behavioral code is represented as a bag of words, and the TF-IDF importance score of each word within the code is calculated. Table B3 presents the top 50 keywords from the behavioral codes for each game scenario.

To understand which keywords influence the behaviors, as deciphered by LLMs, we perform a linear regression analysis. Treating each behavioral code as an observation, we generate 10 behavior choices with this code and predict the mean behavior based on the occurrence of keywords. An ordinary least squares (OLS) linear regression model is applied using one-hot keyword occurrence feature vectors with 50 dimensions. Table B4 presents the complete linear regression results, showing the impact of each keyword in the game scenarios. Table 1 highlights the top 10 keywords in the linear regression analysis.

We further explore the deciphered information contained in the obtained behavioral codes by analyzing the correlations between the elicited behaviors and the principal components (PCs) of the behavioral codes. The PCs are derived from the 50-dimensional one-hot keyword vectors representing the codes for each game. Each code is used to elicit a behavior 10 times, and the Spearman’s correlation between the mean elicited behavior and the PC value is computed. Figure B4 shows the first five PCs, where moderate to strong correlations are observed, demonstrating the effectiveness of behavioral codes in deciphering behaviors.

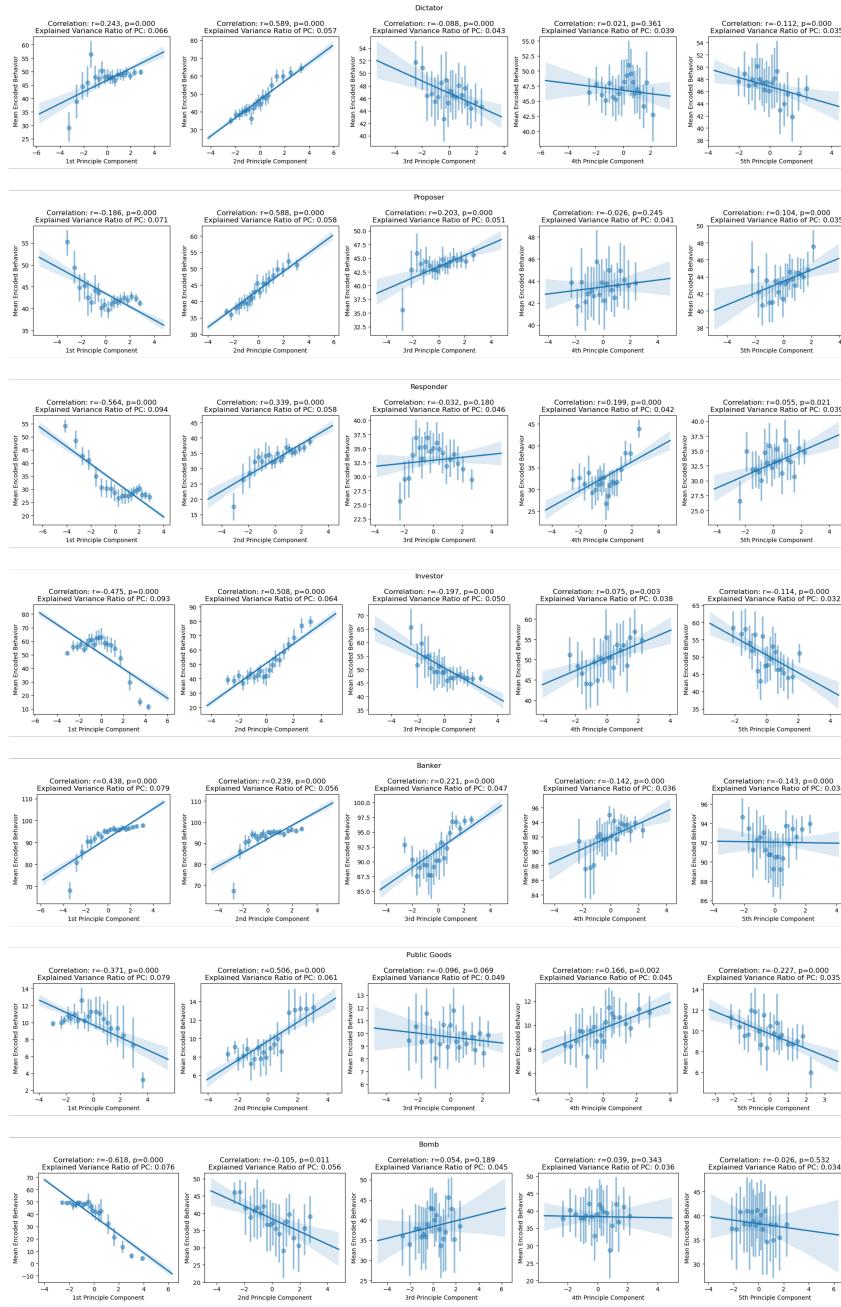


Fig. B4: Correlations between the elicited behaviors and behavioral codes' principal components (PCs). Principal components are obtained from the 50-dimensional, one-hot keyword vectors of the codes of each game. Spearman's correlation is calculated between the PC score of each behavioral code and the mean of elicited behaviors. The figure displays the first five PCs.

| Dictator | Proposer | Responder | Investor | Banker | Public Goods | Bomb |
|-------------|---------------|--------------|--------------|---------------|---------------|--------------|
| decision | proposal | decision | decision | investor | group | risk |
| fairness | decision | fairness | risk | profit | contribution | decision |
| strategic | player | proposal | investment | decision | benefit | reward |
| maker | strategic | ensure | potential | ensure | decision | potential |
| aim | ensure | outcome | return | balance | personal | high |
| balance | party | benefit | aim | term | collective | maximize |
| outcome | aim | reflect | balance | strategic | aim | balance |
| benefit | outcome | strategic | strategic | trust | maximize | gain |
| ensure | offer | maker | reflect | maximize | balance | aim |
| choice | fairness | offer | investor | aim | contribute | outcome |
| reflect | maximize | aim | maximize | future | individual | choice |
| generosity | acceptance | fair | approach | benefit | strategic | probability |
| party | benefit | value | choice | gain | outcome | maker |
| maximize | accept | gain | growth | long | payoff | point |
| fair | balance | make | make | relationship | ensure | strategic |
| value | fair | balance | ensure | maintain | gain | approach |
| make | likely | accept | high | return | overall | minimize |
| consider | consider | prioritize | outcome | foster | make | make |
| maintain | propose | consider | gain | fairness | return | carefully |
| prioritize | maker | high | conservative | fair | maker | achieve |
| self | mutual | secure | maker | banker | high | optimal |
| resource | negotiator | evaluate | cautious | investment | resource | strategist |
| advantage | make | rational | trust | encourage | consider | loss |
| thoughtful | goal | reasonable | maintain | party | success | calculate |
| slightly | generosity | share | moderate | reflect | prioritize | strategy |
| generous | likelihood | equitable | reward | cooperation | reflect | optimize |
| create | focus | achieve | carefully | player | optimize | safety |
| gain | understand | choice | resource | maker | participant | scenario |
| positive | generous | substantial | achieve | value | cooperative | scenario |
| sense | reflect | focus | seek | mutual | approach | avoid |
| goodwill | gain | receive | balanced | immediate | term | ensure |
| strive | cooperation | strong | strategy | outcome | focus | prioritize |
| favor | create | scenario | prioritize | consider | carefully | excel |
| approach | value | carefully | optimize | ongoing | game | reflect |
| focus | prioritize | self | consider | focus | long | possible |
| reasonable | foster | possible | prudent | interaction | potential | cautious |
| considerate | positive | standard | significant | continue | strategy | conservative |
| scenario | high | sense | invest | collaboration | optimal | significant |
| equitable | highly | pragmatic | minimize | prioritize | goal | consider |
| optimize | beneficial | negotiation | calculate | optimize | scenario | involve |
| foster | understanding | acceptable | prefer | importance | possible | focus |
| involve | reasonable | negotiator | opportunity | positive | evaluate | evaluate |
| seek | importance | personal | thoughtful | goal | effort | favor |
| share | empathetic | goal | careful | understand | enhance | goal |
| balanced | optimize | significant | level | self | cooperation | level |
| personal | involve | threshold | substantial | secure | collaborative | choose |
| retain | achieve | make | calculated | reward | achieve | option |
| achieve | self | meet | player | strive | balanced | success |
| understand | agreement | maintain | caution | feel | foster | calculated |
| offer | perceive | advantageous | small | win | slightly | thinker |
| | | | | | cautious | bold |

Table B3: Top 50 keywords in behavioral codes for each game scenario. Keywords are sorted by TF-IDF word importance.

The interpretability of behavioral codes is also consistent and generalizable. As illustrated in Fig B5, the smoothness of behaviors regarding the behavioral code semantics can be observed: Behavioral codes that are close in the semantic space tend to elicit similar behaviors, highlighting the structured and predictable nature of their influence on behavior generation.

Appendix C Game Characteristics and Correlations

Behavioral codes offer new insights into games. From the keywords used in game scenarios (Table B3), we observe notable inter-game relationships. To further dissect these correlations, we analyze behavioral codes in a semantic space. Specifically, each behavioral code is embedded into a 1536-dimensional space using the embedding model OpenAI Ada 3 ([text-embedding-3-large](#)). These high-dimensional vectors are projected onto a 2D low-dimensional map using UMAP. The resulting projection map is displayed in Figure 4, with a heatmap highlighting semantic similarities between games.

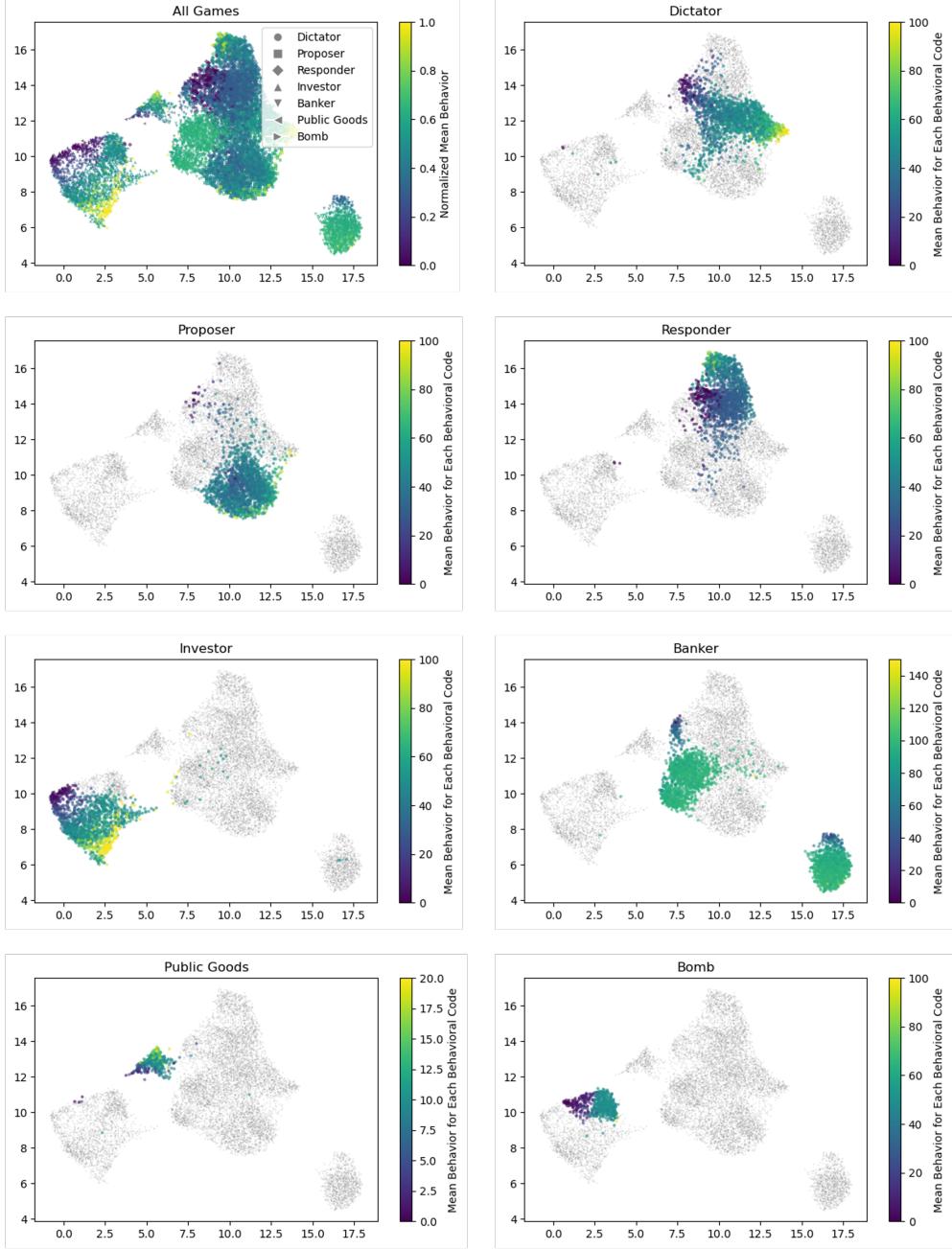


Fig. B5: The 2D projection of the behavioral codes across all games. For each behavioral code, the mean behavior that is elicited is calculated (and normalized to the range $[0, 1]$). Corresponding colors are applied to the codes based on these values. This visualization highlights the continuity of the behavioral codes with respect to their elicited behaviors.

| | Dictator | Proposer | Responder | Investor | Banker | Public Goods | Bomb |
|-------------|---|---|---|--|---|--|---|
| | R^2 = 0.43, MAE = 9.36 Intercept = 41.53 | R^2 = 0.39, MAE = 5.86 Intercept = 39.79 | R^2 = 0.45, MAE = 7.80 Intercept = 31.22 | R^2 = 0.64, MAE = 11.10 Intercept = 49.90 | R^2 = 0.51, MAE = 6.35 Intercept = 77.08 | R^2 = 0.63, MAE = 1.82 Intercept = 5.92 | R^2 = 0.67, MAE = 8.04 Intercept = 42.25 |
| generous | 12.05 | generous | 5.58 | pragmatic | -5.47 | risk | -10.47 |
| generosity | 9.28 | generosity | 4.52 | rational | -5.27 | small | -8.50 |
| retain | -8.98 | gain | -3.75 | high | 5.05 | conservative | -8.22 |
| decision | 6.27 | create | 3.39 | standard | 4.99 | maximize | 8.01 |
| gain | -6.13 | highly | 3.25 | strong | 3.73 | return | 6.77 |
| focus | -5.98 | value | 2.63 | decision | -3.69 | substantial | 6.53 |
| offer | -5.74 | accept | -2.63 | significant | 3.43 | cautious | -6.37 |
| maximize | -3.71 | positive | 2.37 | advantageous | 3.19 | balance | -6.18 |
| self | -3.66 | mutual | 2.37 | negotiation | 3.18 | minimize | -5.93 |
| goodwill | 3.35 | ensure | 2.33 | choice | -3.10 | significant | 5.19 |
| positive | 2.82 | party | 2.29 | reflect | 2.99 | high | 5.04 |
| advantage | -2.57 | maximize | -2.14 | fair | 2.89 | reward | 4.95 |
| party | 2.42 | player | 1.95 | share | 2.88 | ensure | -4.85 |
| create | 2.29 | balance | -1.93 | strategic | -2.72 | prioritize | -4.31 |
| optimize | -2.02 | strategic | -1.74 | substantial | 2.58 | prefer | -4.30 |
| resource | -1.94 | acceptance | 1.40 | proposal | 2.31 | growth | 4.06 |
| share | 1.86 | achieve | -1.34 | gain | -2.30 | gain | 3.97 |
| benefit | 1.72 | decision | -1.33 | negotiator | 1.91 | achieve | 3.53 |
| strategic | -1.70 | cooperation | 1.31 | offer | 1.79 | calculate | 3.42 |
| scenario | -1.50 | self | -1.24 | ensure | -1.78 | trust | 3.41 |
| maintain | -1.38 | proposal | 1.20 | sense | -1.78 | strategic | 3.11 |
| prioritize | -1.20 | maker | 1.14 | carefully | -1.77 | outcome | 3.07 |
| favor | -1.17 | fair | 1.13 | balance | 1.62 | calculated | -2.59 |
| balance | -1.16 | focus | -1.12 | equitable | 1.48 | strategy | 2.58 |
| personal | -1.06 | optimize | -1.11 | outcome | -1.46 | reflect | 2.54 |
| balanced | 0.94 | negotiator | -1.01 | evaluate | -1.33 | aim | 2.40 |
| choice | 0.85 | offer | -0.99 | reasonable | 1.32 | consider | -2.20 |
| fairness | 0.84 | agreement | -0.92 | personal | 1.23 | carefully | -2.10 |
| equitable | 0.84 | beneficial | 0.89 | focus | -1.21 | investment | -2.10 |
| fair | -0.80 | perceive | -0.88 | goal | -1.20 | moderate | 2.08 |
| seek | -0.75 | likelihood | 0.87 | threshold | 1.12 | maintain | -1.98 |
| consider | 0.67 | goal | 0.85 | self | 1.10 | caution | -1.98 |
| thoughtful | 0.66 | outcome | -0.85 | accept | -1.08 | resource | -1.97 |
| approach | -0.65 | understand | -0.84 | maintain | 1.00 | decision | 1.80 |
| make | -0.62 | fairness | 0.78 | prioritize | -0.75 | player | -1.49 |
| reasonable | -0.46 | prioritize | 0.68 | fairness | 0.68 | choice | 1.46 |
| reflect | 0.38 | foster | 0.52 | make | 0.68 | level | 1.34 |
| ensure | -0.35 | empathetic | 0.50 | value | -0.68 | careful | -1.29 |
| slightly | -0.33 | aim | 0.49 | secure | -0.66 | approach | 1.24 |
| involve | -0.32 | involve | -0.49 | maximize | 0.40 | optimize | 1.18 |
| aim | -0.28 | propose | -0.48 | receive | -0.39 | thoughtful | -0.97 |
| achieve | -0.27 | make | 0.38 | possible | 0.37 | seek | 0.90 |
| value | -0.24 | reflect | 0.36 | aim | -0.35 | maker | -0.72 |
| foster | -0.18 | reasonable | -0.36 | consider | 0.33 | balanced | -0.61 |
| understand | 0.16 | benefit | -0.35 | meet | 0.19 | opportunity | 0.57 |
| maker | 0.12 | importance | -0.25 | maker | -0.18 | investor | -0.47 |
| outcome | 0.10 | understanding | -0.24 | benefit | -0.14 | prudent | -0.40 |
| strive | -0.07 | likely | -0.08 | scenario | 0.04 | potential | 0.33 |
| sense | -0.03 | consider | -0.04 | achieve | -0.03 | invest | 0.20 |
| considerate | 0.01 | high | -0.03 | acceptable | 0.02 | make | -0.06 |
| | | | | | | value | -0.06 |
| | | | | | | consider | 0.01 |
| | | | | | | gain | -0.08 |

Table B4: Linear regression analysis of behaviors based on keywords in the behavioral codes. Specifically, each behavioral code is transformed into a one-hot keyword-occurrence feature vector of 50 dimensions to predict the mean behavior that can be elicited by that code. This prediction is performed using an (OLS) linear regression model. The results demonstrate that keyword occurrences in behavioral codes effectively predict the behaviors of LLMs, offering interpretability of the behavioral codes.

The similarities are computed as the aggregated cosine similarity of the high-dimensional embeddings.

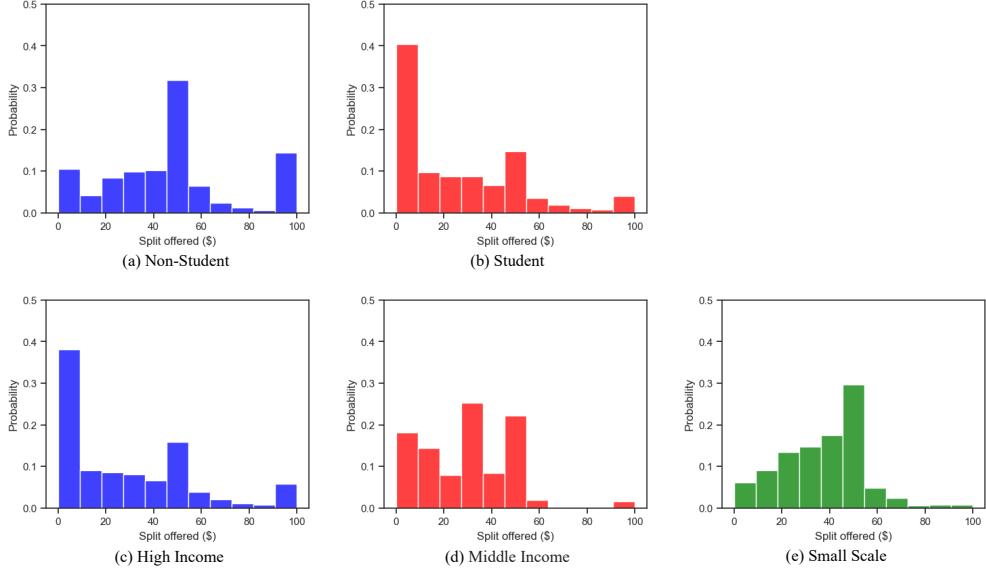


Fig. D6: Behavior distribution of different populations under the Dictator game.

Appendix D Population Signatures

By aligning the behavior distribution with a target population, as detailed in Sec. A.3, LLMs effectively decipher the signatures of this population into a mixture of behavioral codes.

For the MobLab player population, the weighted behavioral codes are highlighted on the 2D projection map shown in Figure 5. Semantic clusters on the map are summarized into keywords using ChatGPT. The projection reveals that the weighted codes are unevenly distributed across the space, offering detailed and nuanced insights into the characteristics of the player population.

Behavioral codes can also serve as a valuable tool for identifying distinct decision-making patterns across different populations. In a meta-study of the Dictator Game [6]³, various populations were analyzed, revealing distinct behavioral distributions. Figure D6 displays the behavior distributions of various populations in the Dictator game, including comparisons between Students and Non-students populations, as well as populations from diverse societal backgrounds. Figure 6 highlights the weighted behavioral codes for these populations, revealing distinctive patterns in their decision-making tendencies: (i) In the Student population, a substantial proportion of individuals exhibit behaviors concentrated in the ‘Highly Competitive and Profit-Oriented’ region. In contrast, the Non-Student population demonstrates a stronger inclination toward ‘Generosity-Oriented Decision-Making’ as well as ‘Fairness-Driven and Collaborative’ choices. (ii) In both the High and Middle Income populations, ‘Highly Competitive and Profit-Oriented’ strategies are prominent, with the High Income population showing a

³Data from Engel [6]: <https://osf.io/xc73h/>, retrieved on March 14, 2025.

slight tendency toward ‘Generosity-Oriented Decision-Making’. By contrast, the Small-Scale population exhibits a stronger emphasis on ‘Fairness-Driven and Collaborative’ behaviors, suggesting a greater inclination toward equitable resource distribution and cooperative decision-making.