Automated Human Strategic Behavior Modeling via Large Language Models

Xiaohan Xie[1,2], **Haoran Yu**[*3], **Biying Shou**[4], **Jianwei Huang**[1,2,5]
[1]School of Science & Engineering, CSIJRI Joint Research Centre on Smart Energy Storage, Shenzhen Key Laboratory of Crowd Intelligence Empowered Low-Carbon Energy Network, The Chinese University of Hong Kong, Shenzhen,
[2]Shenzhen Loop Area Institute,
[3]School of Computer Science & Technology, Beijing Institute of Technology,
[4]School of Management & Economics, The Chinese University of Hong Kong, Shenzhen,
[5]Shenzhen Institute of Artificial Intelligence and Robotics for Society
xiaohanxie@link.cuhk.edu.cn, yhrhawk@gmail.com, biyingshou@cuhk.edu.cn, jianweihuang@cuhk.edu.cn

Abstract

What if machines could discover human behavioral patterns better than experts? Traditional behavioral modeling in economics depends on costly manual refinement by domain experts, severely limiting scalability and discovery potential. We introduce AutoBM, an automated behavioral modeling framework leveraging large language models (LLMs) to systematically generate, evaluate, and refine interpretable behavioral models directly from human behavior data. AutoBM represents candidate models as structured natural language specifications, explicitly defining symbolic terms along with their tunable parameters, interpretations, and design rationales. AutoBM leverages LLMs to automatically translate each language specification into executable code, optimize tunable parameters, and evaluate model performance. Utilizing LLM-guided search strategies, AutoBM iteratively recombines and improves models at the term level, closely mirroring human expert practices. Experiments conducted across three distinct strategic environments (the ultimatum game, repeated rock-paper-scissors, and continuous double auctions) demonstrate that AutoBM-generated models consistently outperform leading manually crafted models, achieving significant improvements in prediction accuracy while maintaining clear interpretability. Our results demonstrate that automated frameworks can not only match but systematically exceed human expertise in behavioral modeling, fundamentally changing how we understand strategic human behavior.

Code — https://github.com/Crescenx/autobm

1 Introduction

Modeling human strategic behavior is a foundational challenge in behavioral economics and decision sciences, critical for optimizing economic policies and market operations (Thaler 2016). Historically, behavioral models have relied extensively on domain experts to manually hypothesize, construct, and iteratively refine candidate models through trial-and-error methods using empirical data (Fehr and Schurtenberger 2018; Burton-Chellew and West 2021). While insightful, this process is labor-intensive, slow, and heavily dependent on subjective expert judgment, thus limiting scalability and general applicability.

Recent advances in deep learning have shown promise in automatically modeling complex behavioral patterns from data (Hartford, Wright, and Leyton-Brown 2016; Kolumbus and Noti 2019). These methods typically generate black-box models lacking interpretability, hindering their adoption in contexts where transparency and understanding of underlying decision-making processes are essential.

Recent advances in large language models (LLMs) enable them to serve as generalpurpose modeling tools beyond traditional natural language tasks (Bommasani et al. 2021; Singhal et al. 2023; Katz et al. 2024). Their emergent reasoning abilities and extensive pretrained knowledge suggest significant potential for behavioral modeling. We therefore pose the following research questions:

- **Framework Design:** How to design a framework that harnesses LLMs to systematically automate the generation and refinement of interpretable behavioral models?
- **Performance Assessment:** How well do these automatically generated interpretable models compare to traditionally crafted models in capturing complex human decision-making?

To address these questions, we introduce **AutoBM**, a framework leveraging the generative and reasoning capabilities of LLMs (Romera-Paredes et al. 2024; Cheng et al. 2024) to automate the discovery and optimization of interpretable behavioral models. Our framework builds on two key ideas to enhance the effectiveness of model design.

The first idea concerns *model representation*, where *we propose a new structure for representing behavioral models to facilitate the automated design process*. When using LLMs to automatically search for solutions requiring computational evaluation (e.g., optimization heuristics (Romera-Paredes et al. 2024; van Stein, Vermetten, and Back 2025), scientific equations (Ma et al. 2024; Shojaee et al. 2025), and statistical models (Li, Fox, and Goodman 2024)), many studies represented the generated solutions directly as code to facilitate evaluation. In contrast, we represent each behavioral model through a structured natural language specification. It explicitly defines the individual symbolic terms that constitute the model, and describes the interpretation and design rationale of each term. Although the specification is later converted into code for model evaluation, our **AutoBM** framework searches for new models directly within

*Corresponding author.

the space of the structured specifications rather than code. This design enables LLMs to conduct more effective exploration and ensures explainability, as the representation itself contains the model's interpretation, removing the need for any expert intervention during the modeling process.

The second idea concerns *model search*, where *we introduce a hybrid approach that integrates two paradigms: evolutionary search and LLM-driven self-refinement*. On the one hand, we enhance the evolutionary process by incorporating LLM-based judgment. Before generating new models, we prompt an LLM to critically evaluate each existing model, providing a detailed rationale that enables more informed and principled recombination and perturbation of existing models. On the other hand, the overarching evolutionary framework ensures robustness, mitigating risks inherent in simple self-refinement cycles. By maintaining a diverse population of models, our framework avoids the accumulation of hallucinations and is less likely to become trapped in a local optimum.

We evaluate **AutoBM** using real behavior data from three strategic environments (the ultimatum game (Lin et al. 2020), repeated rock-paper-scissors (Komai, Kurokawa, and Kim 2022), and continuous double auctions (Ikica et al. 2023)). The contributions of this paper are as follows:

- **Automated Behavioral Modeling**: We introduce **AutoBM**, a framework that operates without human intervention to automatically discover high-performance, interpretable behavioral models, given only behavioral data and a description of the strategic environment.
- **Structured Model Representation**: We propose a structured natural-language-based representation for behavioral models. It explicitly specifies individual terms along with their design rationales, mirroring the reasoning-driven design process of human experts and improving the efficiency of model evolution.

Overall, this research demonstrates the effectiveness of LLM-powered automation in enhancing the efficiency and interpretability of behavioral modeling, providing valuable tools for behavioral scientists and economists seeking to understand and predict complex human strategic interactions.

2 Related Work

Strategic behavior modeling with neural networks There is increasing interest in using neural networks to learn human behavior patterns from data. Most studies framed it as a supervised learning task, and learned behavior in strategic environments, such as one-shot normal-form games (Hartford, Wright, and Leyton-Brown 2016), repeated normal-form games (Kolumbus and Noti 2019), repeated ad auctions (Shen et al. 2020), Stackelberg security games (Perrault et al. 2020), and Chess (McIlroy-Young et al. 2022). The resulting models are black boxes with limited interpretability.

Few studies in this direction have explored model interpretability. For example, (Peterson et al. 2021) studied learning human choices under uncertainty. By constraining the neural network architecture to align with classical decision theories, the study improved the interpretability of the trained model. Trained on extensive human-generated text, LLMs have the potential to reason about human decision processes. Some recent studies prompted LLMs to make decisions in strategic environments (e.g., dictator games (Filippas, Horton, and Manning 2024), persuasion games (Shapira et al. 2024), and ring-network games (Fan et al. 2024)), and compared their consistencies with human decisions. However, none of the existing studies have leveraged LLMs to analyze human behavioral data and propose interpretable behavioral models, which motivates our work.

Automated algorithm and model design with LLMs Using LLMs to automatically generate heuristics for tackling combinatorial optimization problems has gained significant attention (Romera-Paredes et al. 2024; van Stein, Vermetten, and Back 2025; Liu et al. 2024; Ye et al. 2014; Yao et al. 2025; Wu et al. 2024). The idea is to prompt an LLM to iteratively evolve a set of heuristics, where each heuristic is evaluated based on its performance in solving the target optimization problem. The evaluation feedback guides the LLM to refine the heuristics. Our behavioral model design problem differs from automated heuristic algorithm design in two aspects. First, we aim to discover models with tunable parameters that can fit observed data, which introduces an inner loop of parameter optimization nested within the outer loop of model search. Second, we require our models to be interpretable, whereas heuristics in automated algorithm design only need to be provided as executable code.

Recently, (Ma et al. 2024; Shojaee et al. 2025; Li, Fox, and Goodman 2024) used LLMs to automatically discover statistical models and physics equations that explain observed data. Compared to the methods in these studies, the innovations of our approach lie in introducing a more effective structure for model representation and integrating model judgment into the model evolution process.

3 Problem Formulation

Consider a general strategic environment with multiple interacting players whose decisions mutually affect each other. Let $\mathcal{R}$ denote the set of player roles in the environment. For each role $r \in \mathcal{R}$, we have an empirical behavioral dataset $\mathcal{D}_r$, which records real user behavior in strategic interactions. Each data point in $\mathcal{D}_r$ is represented by a tuple $(\boldsymbol{o}_r, a_r)$, where $\boldsymbol{o}_r$ denotes the contextual information available to a player when making a decision, and a_r denotes the player's observed action. Next, we illustrate the tuple $(\boldsymbol{o}_r, a_r)$ using two classical games.

Example 1 (Ultimatum Game) *This game involves two playersa proposer and a responder. The player with $r =$ "proposer" first suggests a division of a fixed resource, and then the player with $r =$ "responder" chooses to accept or reject. If rejected, both players receive nothing. For the proposer, the information $\boldsymbol{o}_r$ is the total resource available, and the action a_r is the proportion of the resource offered to the responder. For the responder, the information $\boldsymbol{o}_r$ includes the offered proportion, and the action a_r is the binary accept/reject decision.*

Example 2 (Repeated Rock-Paper-Scissors Game) *The game involves two players with the same role. They*

repeatedly play a standard rock-paper-scissors game over multiple rounds. For each player, the information $\boldsymbol{o}_r$ comprises historical actions taken by both players in previous rounds, and a_r is the player's current action choice (rock, paper, or scissors).

Given a strategic environment and an empirical behavioral dataset $\mathcal{D}_r$ capturing observed human behavior for a specific role $r \in \mathcal{R}$, our goal is to automatically derive an interpretable mathematical model $\pi_{\boldsymbol{\theta}}$ that effectively describes and predicts player behavior. Let i index each data point, and $\mathcal{I}$ denote the dataset's index set. Then, the dataset $\mathcal{D}_r$ can be written as $\left\{(\boldsymbol{o}_r^{(i)}, a_r^{(i)})\right\}_{i\in\mathcal{I}}$.

We seek a $\boldsymbol{\theta}$-parameterized behavioral model $\pi_{\boldsymbol{\theta}}$ that maps contextual information $\boldsymbol{o}_r^{(i)}$ to a probability distribution over possible actions. Let $\pi_{\boldsymbol{\theta}}(a_r^{(i)}|\boldsymbol{o}_r^{(i)})$ denote the predicted probability assigned to the true action $a_r^{(i)}$. We aim to minimize the following negative log-likelihood (NLL) loss:

$$\min_{\pi_{\boldsymbol{\theta}}} \; -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \log \pi_{\boldsymbol{\theta}}(a_r^{(i)}|\boldsymbol{o}_r^{(i)}). \tag{1}$$

Unlike neural networks, we focus on generating white-box models. These models explicitly represent mathematical relationships between elements of the contextual information $\boldsymbol{o}_r$ and observed actions a_r, using a limited number of parameters $\boldsymbol{\theta}$ that are interpretable and practically meaningful.

4 Automated Behavior Modeling Framework

4.1 Representation, Evaluation, and Judgment

We develop LLM-aided methods for *representing*, *evaluating*, and *judging* solutions to problem (1).

LLM-generated structural representation $\mathcal{M}$ We denote the structured representation of an LLM-generated solution as $\mathcal{M}$. Specifically, $\mathcal{M}$ includes the following four components:

- **Parameter dictionary**: It is a dictionary of the form $\{\theta_j : (\texttt{para\_name}_j, \texttt{para\_desc}_j, \texttt{para\_domain}_j)\}$. It explicitly specifies each tunable parameter θ_j, including its natural-language name $\texttt{para\_name}_j$, description $\texttt{para\_desc}_j$, and valid numerical range $\texttt{para\_domain}_j$.
- **Constant dictionary**: The dictionary has the form $\{c_m : (\texttt{cons\_name}_m, \texttt{cons\_desc}_m, \texttt{cons\_val}_m)\}$. It details each constant c_m used within the model, specifying its natural-language name $\texttt{cons\_name}_m$, description $\texttt{cons\_desc}_m$, and numerical value $\texttt{cons\_val}_m$.
- **Term dictionary**: It is a dictionary of the form $\{\phi_n : (\texttt{term\_name}_n, \texttt{term\_desc}_n, \texttt{term\_expr}_n)\}$. It defines each mathematical term ϕ_n explicitly, specifying its name $\texttt{term\_name}_n$, description $\texttt{term\_desc}_n$, and mathematical expression $\texttt{term\_expr}_n$. Each term uses parameters θ_j and constants c_m to characterize how contextual information $\boldsymbol{o}_r$ influences action probabilities.
- **Action probability function** $\pi_{\boldsymbol{\theta}}(a_r|\boldsymbol{o}_r)$: It combines the defined terms ϕ_n to explicitly calculate the action probabilities given the contextual information $\boldsymbol{o}_r$.

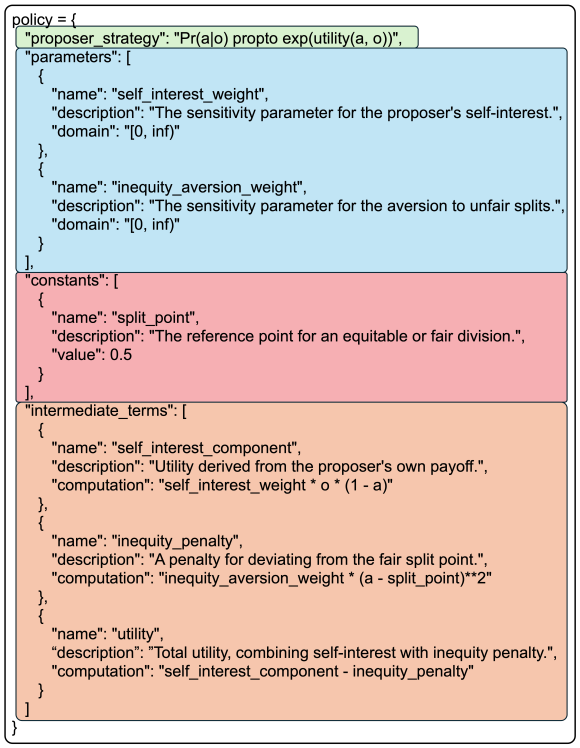

Figure 1: Structural representation of a behavioral model for the Ultimatum game.

Figure 1 illustrates an example of $\mathcal{M}$. Our model representation is a key contribution, diverging from prior work that represents models directly as code (Romera-Paredes et al. 2024; Ma et al. 2024; Li, Fox, and Goodman 2024). We instead prompt the LLM to generate a structured natural language representation, which offers three distinct advantages: (i) **Term-level characterization**: The representation explicitly defines the individual symbolic terms, ϕ_n, that constitute the model's action probability function. This term-level granularity enables targeted modifications, such as adding or altering specific components, making the search for improvements more fine-grained and efficient. (ii) **Language-based rationale**: The representation also embeds a natural language description for each component, ensuring the design rationale is identifiable and traceable throughout the search process. Consequently, the LLM is better equipped with the necessary context to reason about and propose principled improvements to the model. (iii) **Human interpretability**: The combination of this explicit structure and embedded rationale makes the entire model readily understandable to human users. One can therefore trace the model's hierarchical components and design logic to gain insights into both the model and the strategic behavior, without requiring subsequent interpretation by domain experts.

Automated code generation and evaluation $\mathcal{V}$ For each model $\mathcal{M}$, we partition the behavioral dataset $\mathcal{D}_r$ into training ($\mathcal{D}_r^{\text{train}}$) and validation ($\mathcal{D}_r^{\text{val}}$) sets. Our generation and evaluation pipeline consists of:

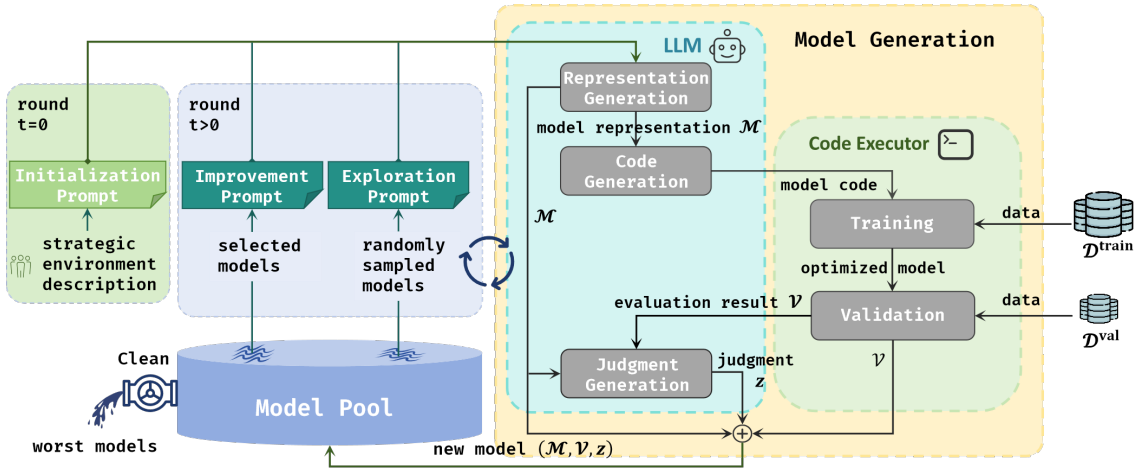

Figure 2: Illustration of Our **AutoBM** Framework.

1. **Code synthesis:** An LLM translates $\mathcal{M}$ into executable PyTorch code, followed by automated verification for consistency.
2. **Parameter optimization:** We minimize the NLL loss on $\mathcal{D}_r^{\text{train}}$ using gradient-based optimization to obtain the optimal parameters.
3. **Performance evaluation:** The optimized model is evaluated on $\mathcal{D}_r^{\text{val}}$, yielding evaluation results $\mathcal{V} = \{(\boldsymbol{o}_r^{(i)}, a_r^{(i)}, \ell^{(i)})\}_{i \in \mathcal{I}_{\text{val}}}$, where $\mathcal{I}_{\text{val}}$ is the index set for the validation data and $\ell^{(i)}$ represents the prediction loss for data point i.

LLM-generated judgment z Utilizing evaluation results $\mathcal{V}$, we prompt the LLM to systematically analyze model performance. To reduce the input length, we may randomly sample a subset $\mathcal{V}' \subseteq \mathcal{V}$ and feed it to the LLM. We prompt the LLM to perform the following tasks:

- **Categorizing patterns of predictions**: This task categorizes patterns across $\mathcal{V}'$, identifying the types of $(\boldsymbol{o}_r^{(i)}, a_r^{(i)})$ for which the model $\mathcal{M}$ incurs high or low prediction loss $\ell^{(i)}$.
- **Evaluating contributions of model terms**: This task evaluates the contributions of individual terms ϕ_n to predictive accuracy, identifying effective and redundant components.
- **Generating concise natural-language judgments**: This task generates a concise natural-language judgment z summarizing predictive strengths and weaknesses.

4.2 Automated Model Design Pipeline

AutoBM employs a hybrid search strategy that maintains a diverse population of models while leveraging intelligent guidance for systematic improvement. As illustrated in Figure 2, our pipeline operates through three phases:

- **Initialization Phase**: We begin by providing the LLM with a comprehensive description of the strategic environment, including game rules, player roles, and contextual information structure. The LLM generates an initial diverse population $\mathcal{P}_0 = \{(\mathcal{M}_k, \mathcal{V}_k, z_k)\}_{k=1}^K$ of K candidate models, where each entry contains a model representation $\mathcal{M}_k$, its evaluation results $\mathcal{V}_k$, and performance judgment z_k.
- **Expansion Phase**: In each iteration $t \in \{1, \ldots, T\}$, we expand the model pool $\mathcal{P}_{t-1}$ via two complementary strategies:
  - **Exploitation via Improvement**: We select top-performing models from $\mathcal{P}_{t-1}$ based on evaluation results $\mathcal{V}_k$. Leveraging explicit insights from LLM-generated judgments z_k, we prompt the LLM to construct new models by iteratively recombining beneficial terms and removing ineffective terms, thereby improving predictive accuracy.
  - **Exploration via Diversification**: To maintain model diversity and thoroughly explore the solution space, we randomly sample models from $\mathcal{P}_{t-1}$ and prompt the LLM to generate new models by investigating novel or previously unconsidered behavioral assumptions and terms.
- **Selection and Cleaning Phase**: Newly generated models from the expansion stage undergo evaluation and judgment. Models that perform poorly (those with high average NLLs) are eliminated to maintain a fixed pool size, resulting in a new model pool $\mathcal{P}_t$.

The process terminates upon convergence or after a maximum of T iterations. The power of this hybrid architecture lies in its synergy: the evolutionary process provides a robust backbone that prevents search stagnation and hallucination accumulation, while the LLM injects semantic intelligence into the process, enabling a more targeted and rationale-driven exploration of the model space.

5 Experiments

We conduct experiments across three environments: the ultimatum game, the repeated rock-paper-scissors (RPS) game, and single-unit continuous double auctions. They span vary-

Method	NLL
AutoBM	**5.306**
BoxLM	5.476
FunSearch	5.608
J&I	5.724
QRE	6.060
IA	5.954
ERC	5.977

Table 1: Model Performance in the Ultimatum Game.

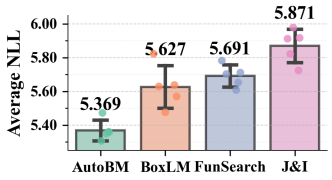

Figure 3: Performance of LLM-Based Frameworks Over 5 Runs in the Ultimatum Game.

ing complexity levels in both the game rules and the information $\boldsymbol{o}_r$ that a player considers when making decisions.

We compare the models generated by our **AutoBM** against four baselines: (1) **BoxLM models**, generated through an LLM-based self-critique iterative refinement process (Li, Fox, and Goodman 2024); (2) **FunSearch models**, generated using an evolutionary algorithm to guide LLM-based exploration (Romera-Paredes et al. 2024); (3) **Human-crafted models**, established behavioral models from existing literature; (4) **Judgment and Improvement models (J&I)**, models derived by performing a single round of evaluation and refinement on human-crafted models, leveraging LLM judgments adapted from our framework.

To ensure fair comparisons, we implement several controls. All LLM-based frameworks receive the same input information and the same computational budget, allowing up to 200 candidate models post-initialization. Furthermore, all models with tunable parameters are trained on the same dataset until convergence.

Although the dataset may contain outliers, they constitute a small proportion. Since models are trained using the average NLL loss, the influence of these outliers is limited, preserving the approach's overall robustness. For our **AutoBM**, we initialize the model pool with 5 models and update it over 20 rounds, with each round generating 5 improvement-based and 5 exploration-based models.

In all LLM-based frameworks, we use Qwen2.5-Max (Team 2024), chosen for its mathematical and reasoning capabilities, to handle language-intensive tasks such as model generation. Code implementation is automated using Claude 3.7 Sonnet, selected for its code synthesis performance.

5.1 Ultimatum Game

Experimental setup We focus on the proposer's behavior, denoting the total resource as o and the proposed share allocated to the responder as $a \in [0, 1]$. We use a dataset comprising 38,287 trials of ultimatum game (Lin et al. 2020), dividing it into 70% for training, 15% for validation, and 15% for testing.

We consider three human-crafted models: (1) Quantal Response Equilibrium (**QRE**) (Haptonstahl 2008), which assumes the players randomly select actions based on expected utilities; (2) Equity, Reciprocity, and Competition (**ERC**) (Bolton and Ockenfels 2000), which assumes the proposer balances self-interest with fairness concerns; (3) Inequity Aversion (**IA**) (Fehr and Schmidt 1999), which is similar to **ERC** but models fairness considerations differently.

Comparison results Table 1 summarizes the predictive performance of various models on the test dataset using negative log-likelihood (NLL) as the evaluation metric. For each LLM-based framework, we conduct the experiment 5 times, selecting the model with the best validation performance from these runs for comparison. Figure 3 further illustrates the performance variability across the 5 runs. All LLM-based frameworks outperform the human-crafted models, with our **AutoBM** consistently delivering the lowest NLL across all runs. The computational cost of our **AutoBM** remains comparable to those of the other two iterative frameworks. The average token consumption per model generation for **AutoBM** is 7,030, compared to 5,969 for **BoxLM** and 3,922 for **FunSearch**. Remarkably, the **J&I** framework, employing a single round of LLM-based refinement on human-crafted models, significantly improves upon these human-crafted models.

Discovered behavioral model Equations (2)-(3) present the best-performing model discovered by our **AutoBM**, defined below (the parameters $\theta_1, \ldots, \theta_6$ are non-negative):

$$\pi_{\boldsymbol{\theta}}(a|o) \propto \exp(u_{\boldsymbol{\theta}}(a, o)), \tag{2}$$

$$u_{\boldsymbol{\theta}}(a, o) = \underbrace{\theta_1(1-a)o}_{\text{self-interest term}} + \underbrace{\frac{\theta_2}{1+e^{-\theta_3(a-0.5)}}}_{\text{responder pressure term}} + \underbrace{\theta_4\left(1-|a-0.5|^{\theta_5}\right)(\ln(1+o))^{\theta_6}}_{\text{context-adjusted fairness term}}. \tag{3}$$

This model assumes that the probability density of selecting a share a is proportional to the exponential of the utility $u_{\boldsymbol{\theta}}(a, o)$ for that choice. This is analogous to the logit quantal response adopted in conventional behavior theories. The $u_{\boldsymbol{\theta}}(a, o)$ explicitly captures three insights: *(1) Self-interest term*: It represents the proposer's incentive to maximize personal gain; *(2) Responder pressure term*: It reflects the proposer's anticipation of the responder's acceptance likelihood as the offered share a increases; *(3) Context-adjusted fairness term*: It captures fairness considerations, becoming prominent when the offer a approaches an equitable split and is intensified by the total resource o.

5.2 Repeated Rock-Paper-Scissors Game

Experimental setup As described in Example 2, in round $s \in \{0, \ldots, S-1\}$, each player chooses an action a_s from $\{\text{Rock}, \text{Paper}, \text{Scissors}\}$. The information $\boldsymbol{o}_s$ includes historical actions of both the modeled player and its opponent. We employ a dataset consisting of 500 independent

Method	NLL	Acc (%)
AutoBM	**1.074**	**44.29**
BoxLM	1.091	37.94
Funsearch	1.094	38.63
J&I	1.092	38.22
SPNE	1.099	33.33
RM	1.098	35.49
FP	1.099	33.18
WLSL	1.095	35.92

Table 2: Model Performance in the Repeated RPS Game.

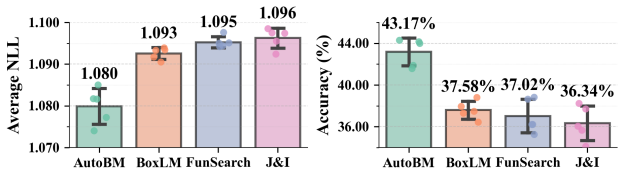

Figure 4: Performance of LLM-Based Frameworks Over 5 Runs in the Repeated RPS Game.

episodes, each containing 50 rounds of repeated rock-paper-scissors interactions (Komai, Kurokawa, and Kim 2022).

We consider four human-crafted baselines: (1) Subgame Perfect Nash Equilibrium (**SPNE**) (Osborne et al. 2004), where players randomly select actions uniformly, reflecting equilibrium behavior; (2) Regret Matching (**RM**) (Hart and Mas-Colell 2000), where players adapt action probabilities based on accumulated regrets; (3) Fictitious Play (**FP**) (Fudenberg and Levine 1998), where players form beliefs about opponents' future actions based on historical frequencies and optimize actions accordingly; (4) Win-Stay-Lose-Shift (**WSLS**) (Nowak and Sigmund 1993), where players repeat successful actions and switch after unsuccessful ones.

Comparison results Table 2 compares model performance based on NLL and the accuracy for predicting a_s. Figure 4 further illustrates performance variations across five experimental runs. Our **AutoBM** significantly outperforms all baseline models.

Discovered behavioral model Equations (4)-(5) present the optimal model identified by our **AutoBM**, where $\theta_2 \in [0, 1]$ is a discounting factor, $\theta_1, \theta_3, \theta_4$ are positive real numbers, and c is a positive integer constant. Here, v_τ indicates the game outcome in round τ ($v_\tau = 1$ if the modeled player wins, $v_\tau = -1$ otherwise).

$$\pi_{\boldsymbol{\theta}}(a_s|\boldsymbol{o}_s) \propto \exp\left(u_{\boldsymbol{\theta}}(a_s, \boldsymbol{o}_s)\right), \tag{4}$$

$$u_{\boldsymbol{\theta}}(a_s, \boldsymbol{o}_s) = \underbrace{\theta_1 \sum_{\tau=0}^{s-1} \theta_2^{s-1-\tau} \mathbf{1}_{\{a_\tau = a_s\}}}_{\text{recency-weighted action bias}} - \underbrace{\frac{\theta_3}{c} \sum_{\tau=s-c}^{s-1} \mathbf{1}_{\{a_\tau = a_s\}}}_{\text{action-specific penalty}}$$

$$+ \underbrace{\frac{\theta_4}{c} \sum_{\tau=s-c}^{s-1} \mathbf{1}_{\{a_\tau = a_s\}} v_\tau}_{\text{outcome-based emotional bias}} . \tag{5}$$

Method	NLL	MSE
AutoBM	**4.788**	**0.0450**
FunSearch	4.986	0.0503
BoxLM	5.056	0.0520
J&I	5.652	0.0588
BB	6.037	0.1663
TDB	6.607	0.0749

Table 3: Performance in Continuous Double Auctions.

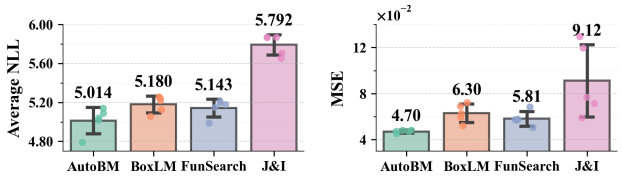

Figure 5: Performance of LLM-Based Frameworks Over 5 Runs in Continuous Double Auctions.

This utility function $u_{\boldsymbol{\theta}}(a_s, \boldsymbol{o}_s)$ explicitly captures three behavioral components: *(1) Recency-weighted action bias*: It represents the tendency to repeat recently chosen actions; *(2) Action-specific penalty*: It discourages frequent repetitions of the same action; *(3) Outcome-based emotional bias*: It reflects preferences for actions associated with recent wins.

5.3 Continuous Double Auctions

Experimental setup We focus on buyers' bidding behavior in single-unit continuous double auction experiments conducted by (Ikica et al. 2023). Each experiment includes multiple buyers and sellers trading indivisible units over 150 seconds. Each buyer demands a single unit, and each seller supplies one. Buyers may submit bids, sellers may submit asks, and all active quotes are displayed in order books. A transaction occurs when a bid meets an ask. We aim to derive the action probability function $\pi_{\boldsymbol{\theta}}(a_\tau|\boldsymbol{o}_\tau)$, where the action $a_\tau \in [0, 1]$ represents a buyer's bid at time τ, normalized by its reservation price. The contextual information $\boldsymbol{o}_\tau = (\tau, \boldsymbol{h}_\tau, \boldsymbol{p}_\tau, \mathcal{E}_\tau, \mathcal{B}_\tau)$ comprises the current timestep τ, the buyer's historical bids $\boldsymbol{h}_\tau$, historical transaction prices $\boldsymbol{p}_\tau$, and the current order books $(\mathcal{E}_\tau, \mathcal{B}_\tau)$ displaying active asks and bids. All prices in $\boldsymbol{o}_\tau$ are normalized by the buyers reservation price. The dataset consists of 17,237 bid records.

We chose this scenario for its novelty and realistic complexity. To our best knowledge, human behavior models within this specific context are poorly explored, and its complexity provides a testbed for evaluating **AutoBM**'s performance on novel, real-world tasks.

We consider two human-crafted baselines: (1) Belief-based Bidding (**BB**) (Gjerstad and Dickhaut 1998), where buyers bid to maximize expected utilities estimated using beliefs about bid acceptance probabilities derived from order books; (2) Time-Dependent Bidding (**TDB**), where buyers increase bids as deadlines approach.

Comparison results Table 3 shows that the model proposed by our **AutoBM** achieves the lowest NLL and MSE

among all models. Figure 5 illustrates that over 5 experimental runs, our method achieves the lowest mean NLL and MSE among all LLM-based frameworks.

Discovered behavioral model The best model discovered by **AutoBM** first defines several intermediate terms based on the market information $\boldsymbol{o}_\tau$:

$$V_\tau = \frac{\text{std}(\boldsymbol{p}_\tau[-3:])}{\text{mean}(\boldsymbol{p}_\tau[-3:])},\ L_\tau = \min(1, |\mathcal{E}_\tau|/10), \tag{6}$$

$$S_\tau = |\min \mathcal{E}_\tau - \max \mathcal{B}_\tau|, \tag{7}$$

$$P_\tau^{\text{ref}} = 0.7\frac{\boldsymbol{p}_\tau[-3] + 1.5\boldsymbol{p}_\tau[-2] + 2\boldsymbol{p}_\tau[-1]}{1+1.5+2} + 0.3\min \mathcal{E}_\tau. \tag{8}$$

In (6), V_τ measures market volatility based on the variation of the last 3 transaction prices (denoted by $\boldsymbol{p}_\tau[-3:]$), L_τ measures market liquidity based on the number of active ask orders in $\mathcal{E}_\tau$ (capped at 1). In (7), S_τ represents the effective bid-ask spread, calculated as the absolute difference between $\min \mathcal{E}_\tau$ (the lowest ask) and $\max \mathcal{B}_\tau$ (the highest bid). In (8), P_τ^{ref} serves as a market reference price, computed as a weighted average of the last three transaction prices and $\min \mathcal{E}_\tau$ (the lowest ask).

Equations (9)-(10) present the behavioral model $(\theta_1, \ldots, \theta_6 \geq 0)$:

$$\pi_{\boldsymbol{\theta}}(a_\tau|\boldsymbol{o}_\tau) \propto \exp(u_{\boldsymbol{\theta}}(a_\tau, \boldsymbol{o}_\tau)), \tag{9}$$

$$\begin{aligned} u_{\boldsymbol{\theta}}(a_\tau, \boldsymbol{o}_\tau) = {} & \underbrace{\theta_1(1-a_\tau)}_{\text{profit motive}} + \underbrace{\theta_2 \exp(-|a_\tau - P_\tau^{\text{ref}}|)}_{\text{dynamic price anchoring}} \\ & + \underbrace{\theta_3 \sum_{h \in \boldsymbol{h}_\tau[-3:]} \exp\left(-\frac{|a_\tau - h|}{0.1 + 0.1V_\tau}\right)}_{\text{history alignment}} + \underbrace{\theta_4 a_\tau \left(\frac{\tau}{150}\right)^{1.3-0.3L_\tau}}_{\text{non-linear urgency}} \\ & + \underbrace{\theta_5 \left(\frac{\sum_{e \in \mathcal{E}_\tau} \mathbf{1}_{\{e \leq a_\tau\}}}{|\mathcal{E}_\tau|} - \frac{\sum_{b \in \mathcal{B}_\tau} \mathbf{1}_{\{b > a_\tau\}}}{|\mathcal{B}_\tau|}\right) \exp(-\theta_6 S_\tau)(1+L_\tau)}_{\text{liquidity-adjusted matching}}. \end{aligned} \tag{10}$$

The utility function $u_{\boldsymbol{\theta}}(a_\tau, \boldsymbol{o}_\tau)$ in (10) consists of five components: (1) *Profit motive*: This encourages lower bids to seek higher profit gains; (2) *Dynamic price anchoring*: This incentivizes bidding near the market reference price P_τ^{ref}; (3) *History alignment*: This promotes consistency with the buyer's recent bid history; (4) *Non-linear urgency*: This encourages adjusting bids more aggressively as time progresses, with the rate of this increased urgency being influenced non-linearly by market liquidity; (5) *Liquidity-adjusted matching*: This aims to promote successful trades by favoring bids that match existing asks, and its impact is modulated by the bid-ask spread and market liquidity.

5.4 Convergence Performance Comparison

Figure 6 analyzes the convergence dynamics of the LLM-based frameworks in continuous double auctions, plotting the average NLL of the top-5 models against the number of generations. The results reveal a three-phase pattern. (1) **Initial descent**: All frameworks reduce NLL to around 5.20, with **AutoBM** the fastest. (2) **Stagnation**: All frameworks

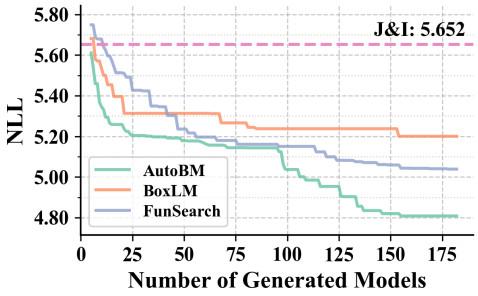

Figure 6: Comparative NLL Evolution Across LLM-based Frameworks (Single Run).

then plateau in a common suboptimal region. This stagnation traps **BoxLM**, highlighting its basic propose-criticize paradigm's limitation in escaping local optima. (3) **Breakthrough**: In contrast, **AutoBM** and **FunSearch** leverage exploration mechanisms to escape. Crucially, **AutoBM** not only achieves convergence empirically with sufficient iterations, but also reaches a superior final performance, demonstrating that its structured representation with integrated validation and judgment enables more effective model space exploration than evolutionary search alone.

6 Discussion and Conclusion

We presented **AutoBM**, a framework that automates the discovery of accurate and interpretable behavioral models without requiring domain expertise. Its innovation is built upon two key ideas: a structured natural language representation for behavioral models, and a hybrid framework that combines evolutionary search with LLM judgments. Empirical evaluations across diverse strategic environments demonstrate that **AutoBM** consistently produces accurate and interpretable models. Notably, **AutoBM** efficiently identifies impactful behavioral insights, providing interpretations aligned with economic and strategic intuitions. Moreover, the modularity of the LLM-based design allows researchers to incorporate knowledge or constraints and extend **AutoBM**'s capabilities with new modules.

A limitation of this work is that experiments were conducted on laboratory data. **AutoBM** can be readily applied to real-world behavior, e.g., sequential bargaining on eBay (Backus et al. 2020; Fong and Waisman 2025) and crowdsourcing (Luo and Jennings 2020, 2021). Another limitation is that **AutoBM** places no explicit bound on the complexity of generated models. This can be mitigated by incorporating a complexity constraint into the prompt or by evaluating each candidate model using a combined accuracycomplexity criterion and performing model search via multi-objective optimization methods.

Our **AutoBM** can be further enhanced by integrating an explicit semantic verifier designed to prevent hallucinations. We can also extend **AutoBM** to personalized behavioral modeling, leveraging LLMs to account for individual variability in decision-making preferences and further enhancing predictive accuracy in diverse practical contexts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62572049 and No. 62202050), the Beijing Institute of Technology Research Fund Program for Young Scholars, Shenzhen Peacock and Guangdong talent program, the National Natural Science Foundation of China (No. 62271434), the Shenzhen Stability Science Program 2023, the Shenzhen Institute of Artificial Intelligence and Robotics for Society, and Longgang District Shenzhen's "Ten Action Plan" for Supporting Innovation Projects (No. LGKCSDPT2024002).

We express our gratitude to the researchers (Lin et al. 2020; Komai, Kurokawa, and Kim 2022; Ikica et al. 2023) who generously shared the datasets publicly online.

References

Backus, M.; Blake, T.; Larsen, B.; and Tadelis, S. 2020. Sequential bargaining in the field: Evidence from millions of online bargaining interactions. *The Quarterly Journal of Economics*, 135(3): 1319–1361.

Bolton, G. E.; and Ockenfels, A. 2000. ERC: A theory of equity, reciprocity, and competition. *American economic review*, 91(1): 166–193.

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Burton-Chellew, M.; and West, S. 2021. Payoff-based learning best explains the rate of decline in cooperation across 237 public-goods games. *Nature human behaviour*, 5(10): 1330–1338.

Cheng, K.; Yang, J.; Jiang, H.; Wang, Z.; Huang, B.; Li, R.; Li, S.; Li, Z.; Gao, Y.; Li, X.; Yin, B.; and Yizhou, S. 2024. Inductive or deductive? Rethinking the fundamental reasoning abilities of LLMs. *arXiv preprint arXiv:2408.00114*.

Fan, C.; Chen, J.; Jin, Y.; and He, H. 2024. Can large language models serve as rational players in game theory? A systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 17960–17967.

Fehr, E.; and Schmidt, K. M. 1999. A theory of fairness, competition, and cooperation. *The quarterly journal of economics*, 114(3): 817–868.

Fehr, E.; and Schurtenberger, I. 2018. Normative foundations of human cooperation. *Nature human behaviour*, 2(7): 458–468.

Filippas, A.; Horton, J. J.; and Manning, B. S. 2024. Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus? In *Proceedings of the 25th ACM Conference on Economics and Computation*, 614–615.

Fong, J.; and Waisman, C. 2025. The effects of delay in bargaining: Evidence from eBay. *Management Science*.

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

Gjerstad, S.; and Dickhaut, J. 1998. Price formation in double auctions. *Games and economic behavior*, 22(1): 1–29.

Haptonstahl, S. R. 2008. Bargaining Under Uncertainty: A Strategic Random Utility Model of the Ultimatum Game. Technical report, Citeseer.

Hart, S.; and Mas-Colell, A. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5): 1127–1150.

Hartford, J.; Wright, J.; and Leyton-Brown, K. 2016. Deep learning for predicting human strategic behavior. *Advances in neural information processing systems*, 29.

Ikica, B.; Jantschgi, S.; Nax, H. H.; Duran, D. G. N.; and Pradelski, B. S. 2023. Competitive market behavior: Convergence and asymmetry in the experimental double auction. *International Economic Review*, 64(3): 1087–1126.

Katz, D. M.; Bommarito, M. J.; Gao, S.; and Arredondo, P. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270): 20230254.

Kolumbus, Y.; and Noti, G. 2019. Neural networks for predicting human interactions in repeated games. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 392–399.

Komai, T.; Kurokawa, H.; and Kim, S.-J. 2022. Human randomness in the rock-paper-scissors game. *Applied Sciences*, 12(23): 12192.

Li, M.; Fox, E.; and Goodman, N. 2024. Automated statistical model discovery with language models. In *Proceedings of the 41st International Conference on Machine Learning*, 27791–27807.

Lin, P.-H.; Brown, A. L.; Imai, T.; Wang, J. T.-y.; Wang, S. W.; and Camerer, C. F. 2020. Evidence of general economic principles of bargaining and trade from 2,000 classroom experiments. *Nature Human Behaviour*, 4(9): 917–927.

Liu, F.; Tong, X.; Yuan, M.; Lin, X.; Luo, F.; Wang, Z.; Lu, Z.; and Zhang, Q. 2024. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning*, 32201–32223.

Luo, Y.; and Jennings, N. R. 2020. A differential privacy mechanism that accounts for network effects for crowdsourcing systems. *Journal of Artificial Intelligence Research*, 69: 1127–1164.

Luo, Y.; and Jennings, N. R. 2021. A budget-limited mechanism for category-aware crowdsourcing of multiple-choice tasks. *Artificial Intelligence*, 299: 103538.

Ma, P.; Wang, T.-H.; Guo, M.; Sun, Z.; Tenenbaum, J.; Rus, D.; Gan, C.; and Matusik, W. 2024. LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *Proceedings of the 41st International Conference on Machine Learning*, 33940–33962.

McIlroy-Young, R.; Wang, R.; Sen, S.; Kleinberg, J.; and Anderson, A. 2022. Learning models of individual behavior in chess. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1253–1263.

Nowak, M.; and Sigmund, K. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432): 56–58.

Osborne, M. J.; et al. 2004. *An introduction to game theory*, volume 3. Springer.

Perrault, A.; Wilder, B.; Ewing, E.; Mate, A.; Dilkina, B.; and Tambe, M. 2020. End-to-end game-focused learning of adversary behavior in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1378–1386.

Peterson, J.; Bourgin, D.; Agrawal, M.; Reichman, D.; and Griffiths, T. 2021. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547): 1209–1214.

Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, P.; Dupont, E.; Ruiz, F. J.; Ellenberg, J.; Wang, P.; Fawzi, O.; Kohli, P.; and Fawzi, A. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995): 468–475.

Shapira, E.; Madmon, O.; Reichart, R.; and Tennenholtz, M. 2024. Can LLMs replace economic choice prediction labs? The case of language-based persuasion games. *arXiv preprint arXiv:2401.17435*.

Shen, W.; Peng, B.; Liu, H.; Zhang, M.; Qian, R.; Hong, Y.; Guo, Z.; Ding, Z.; Lu, P.; and Tang, P. 2020. Reinforcement mechanism design: With applications to dynamic pricing in sponsored search auctions. In *Proceedings of the AAAI conference on artificial intelligence*, 2236–2243.

Shojaee, P.; Meidani, K.; Gupta, S.; Farimani, A. B.; and Reddy, C. 2025. LLM-SR: Scientific equation discovery via programming with large language models. In *Proceedings of the International Conference on Learning Representations*.

Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S. S.; Wei, J.; Chung, H. W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972): 172–180.

Team, Q. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Thaler, R. 2016. Behavioral economics: Past, present, and future. *American economic review*, 106(7): 1577–1600.

van Stein, N.; Vermetten, D.; and Back, T. 2025. In-the-loop hyper-parameter optimization for LLM-based automated design of heuristics. *ACM Transactions on Evolutionary Learning*.

Wu, X.; Wu, S.-h.; Wu, J.; Feng, L.; and Tan, K. C. 2024. Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*, 29(2): 534–554.

Yao, S.; Liu, F.; Lin, X.; Lu, Z.; Wang, Z.; and Zhang, Q. 2025. Multi-objective evolution of heuristic using large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 27144–27152.

Ye, H.; Wang, J.; Cao, Z.; Berto, F.; Hua, C.; Kim, H.; Park, J.; and Song, G. 2014. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. *Advances in neural information processing systems*, 37: 43571–43608.