Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

MODEL VALIDATION AND LLM-BASED MODEL ENHANCEMENT FOR ANALYZING NETWORKED ANAGRAM EXPERIMENTS

Hao He[1], Xueying Liu[1], and Xinwei Deng[1]

[1]Department of Statistics, Virginia Tech, Blacksburg, VA, USA

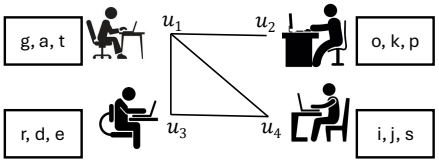
ABSTRACT

Agent-based simulations for networked anagram games, often taking advantage of experimental data, are useful tools to investigate collaborative behaviors. To confidently incorporate statistical analysis of experimental data into agent-based simulations, it is crucial to conduct sufficient validation for the underlying statistical models. In this work, we propose a systematic approach to evaluate the validity of statistical methods of players' action sequence modeling for networked anagram experiments. The proposed method can appropriately quantify the effect and validity of expert-defined covariates for modeling the players' action sequence data. We further develop a Large Language Model (LLM)-guided method to augment the covariate set, employing iterative text summarization to overcome token limits. The performance of the proposed methods is evaluated under different metrics tailored for imbalanced data in networked anagram experiments. The results highlight the potential of LLM-driven feature discovery to refine the underlying statistical models used in agent-based simulations.

1 INTRODUCTION

1.1 Background

Anagram Game. Online anagram games/experiments were conducted with team members playing via screens in their web browsers. Each team's goal was to form as many words as possible in the five-minute game. Team players split the earnings evenly, irrespective of their individual performance (earnings were proportional to the number of words formed). One possible game configuration is shown in Figure 1a, with four players and player degrees (i.e., numbers of neighbors) ranging from 1 to 3. In a more general setting, the number of players in a game is denoted as n and the player degrees are denoted as d. Each player was given three letters initially; for this setup, the letters are shown in the boxes beside the players. Each player could choose any of three actions at any time t: requesting a letter from a neighboring player, replying to a neighbor's letter request, or forming a word. Actions could be repeated in any order and any number of times. (Analysis of the data shows that player actions can be discretized into integer seconds because there are very few occurrences, over all experimental data, of two consecutive actions within a one-second interval, and that the majority of time a player is idle (i.e., thinking) and not taking one of the three actions.) A possible (but fictitious) sequence of player actions is given in Figure 1b. Most of these actions are between players u_3 and u_1, where u_3 requests letters (a, t, and g) from u_1, and then u_1 replies at later times with the requested letters. These letters enable u_3 to form words *trader* and *grader*. In this game, based on the edges, u_3 can interact with u_1 and u_4, but not u_2. When a player receives a requested letter, the player providing the letter does not lose it. The provider maintains the letter (the multiplicity of the letter increases by one). Also, when a player forms a word, she does not lose the letters of the word and can reuse them. This is why u_3 can form *trader* and then *grader*: the letters in $\{r,a,d,e\}$ are not lost. In fact, both words indicate that a letter possessed by a player, in this case r, can be used any number of times in a word. These rules were used to simplify the game in enabling players to form more words and thus increase their earnings.



(a) A four-player game configuration

Time (s)	Initiating Player	Receiving Player	Action
10	u_3	u_1	Request t
14	u_3	u_1	Request a
15	u_1	u_3	Reply t
21	u_1	u_3	Reply a
25	u_3	--	Form $trader$
31	u_3	u_1	Request g
42	u_1	u_3	Reply g
46	u_3	--	Form $grader$
51	u_3	u_4	Request i

(b) Illustrative game actions

Figure 1: Online anagram, i.e., word formation, game. In (a), a four-player game configuration is shown, with communication channels for sharing letters denoted by lines (edges). In this configuration, u_1 and u_4 can communicate and share letters, but u_2 and u_4 cannot. In modeling this game, a player can take any of four actions any number of times: idle [i.e., thinking] (a_1), replying to a neighbor's request (a_2), requesting a letter from a neighboring player (a_3), or forming a word (a_4). In (b), illustrative player actions are focused on u_3.

The game actions for each player are recorded as a time sequence depicted in Figure 1b. In this work, we use the processed data that only contains the following information: players' action sequence, players' pending requests at each time point, and the number of letters available to players. The data analyzed in this work contains the 300-second time sequence of 210 players, under the game setting of number n of players $n = 10$, and number d of neighbors of players $d = 2$.

Multinomial Logistic Regression for Modeling Player's Action. In previous work, player actions in an anagram game are modeled as a discrete-time stochastic process (Ren et al. 2018). At each time-step, players choose one of the four actions a_k, $k \in \{1,2,3,4\}$: staying idle (a_1), replying to a neighbor's request (a_2), requesting a letter from a neighboring player (a_3), or forming a word (a_4). This decision is assumed to be influenced by four key factors (variables): (i) the number of pending letter requests that the player has not yet answered $Z_B(t)$, (ii) the number of letters currently available for forming words $Z_L(t)$, (iii) the total words already formed by that player $Z_W(t)$, and (iv) the number of consecutive time-steps the player has taken the same action $Z_C(t)$.

The rationale for using these expert-selected covariates is as follows. As the number $Z_B(t)$ increases, the more likely a player will respond to these letter requests. As the number $Z_L(t)$ of letters a player has increases, the more likely she is to form words than to request more letters. As the number $Z_W(t)$ of words formed increases, the more skill a player has and is therefore more likely to form more words. Particularly for the idle state, as the time count $Z_C(t)$ increases for consecutive idle states, the more thinking a player has done and therefore the more likely a player is to act (i.e., request, reply, or form word).

Conditioned on the player's most recent action (denoted as Now-State) $a_i(t)$, we apply a multinomial logistic regression (MLR) to estimate the probability of next action (denoted as Next-State), i.e., taking action a_j at time $(t+1)$. Formally,

$$\pi_{ij}(t+1) \;=\; \frac{\exp\left(\mathbf{z}^T \beta_j^{(i)}\right)}{\sum_{m=1}^{4} \exp\left(\mathbf{z}^T \beta_m^{(i)}\right)}, \quad j \in \{1,2,3,4\}, \tag{1}$$

where $\mathbf{z} = \left(1, Z_B(t), Z_L(t), Z_W(t), Z_C(t)\right)^T$ comprises an intercept plus the four covariates, and $\beta_j^{(i)}$ is the corresponding parameter vector for transitioning from action i to action j. (In many formulations, we use

the indices i and j instead of the actions a_i and a_j for clarity.) The model is *conditioned* on each of the four possible most recent actions (Now-State), so that this setup captures how a player's behavior depends not only on her internal state $\mathbf{z}$ but also on which action was most recently taken. In this work, we denote this model as the **original** model.

1.2 Motivation

The anagram game can be used to investigate how individual decisions and interactions can lead to emergent collective behaviors, particularly when studied via agent-based models (ABMs). An action sequence model (ASM) via multinomial logistic regression (MLR) has been used extensively in previous work to simulate each player's decision-making process within such ABMs. In previous work (Cedeno-Mieles et al. 2020), validation of the model has been done by comparing the simulation output and experimental data. However, the covariates used in the *statistical model* have not yet undergone systematic validation, leaving open questions about its validity and the possibility that it overlooks key behavioral dynamics that may lead to improved model performance. Moreover, many existing statistical modeling workflows rely on domain experts handpicking covariates, an approach that may be inadequate for large, complex datasets. As collaborative experiments grow in scale and nuance, there is a pressing need to augment expert knowledge with more robust, data-driven methods that can rigorously evaluate and refine behavioral models.

1.3 Contributions

The contributions of our work are threefold. First, we conduct a systematic *validation* of the ASM used in prior studies. By systematically examining its predictive capability under various simplifications, we identify which expert-selected covariates contribute marginally to modeling player decisions, and we pinpoint the aspects that need improvement.

Second, we introduce a novel *LLM-based Model Validation and Enhancement* framework, as shown in Figure 2, which includes a *covariate augmentation* approach that shifts the focus from parameter tuning to expanding the covariates set in ABM validation. Unlike traditional model validation—often based solely on domain-expert judgment and processed numerical data—*pretrained language models*, such as BERT (Devlin et al. 2019) and GPT (Brown et al. 2020), have shown strong capabilities in extracting meaningful patterns from massive text corpora, in large part due to their transformer architectures. We therefore employ the *large language model (LLM)* in an iterative summarization workflow to explore additional covariates within textual descriptions of experiment dynamics. This enables us to identify nuanced behavioral factors that might otherwise go unnoticed, thus improving the predictive performance of the ASM, which is the underlying model used in agent-based simulations (ABS). The validation and LLM-based models are explained in Section 3 below.

Third, we propose a novel metric called *mean relative improvement for rare observations* (MRI-RO, specified in Section 3.2) that is tailored to our collaborative anagram game, yet we believe is broadly applicable to scenarios with high class imbalance. By emphasizing pairwise model comparisons and focusing on the accurate prediction of rare but crucial player transitions, this metric complements standard measures such as classification accuracy and area under curve (AUC). Altogether, these contributions form a systematic framework for refining, validating, and improving ASM in complex collaborative settings.

2 RELATED WORK

2.1 Model Validation of ABM

Three anagram ABMs using data-driven ASMs were reported in (Cedeno-Mieles et al. 2020), where the model validations were performed with respected to ABS outputs. In that work, distributions of numbers of players for each of (i) numbers of letter requests sent, (ii) numbers of letter requests received, (iii) numbers of letter replies sent (i.e., player sending letter to requestor), (iv) numbers of replies received (i.e., letter

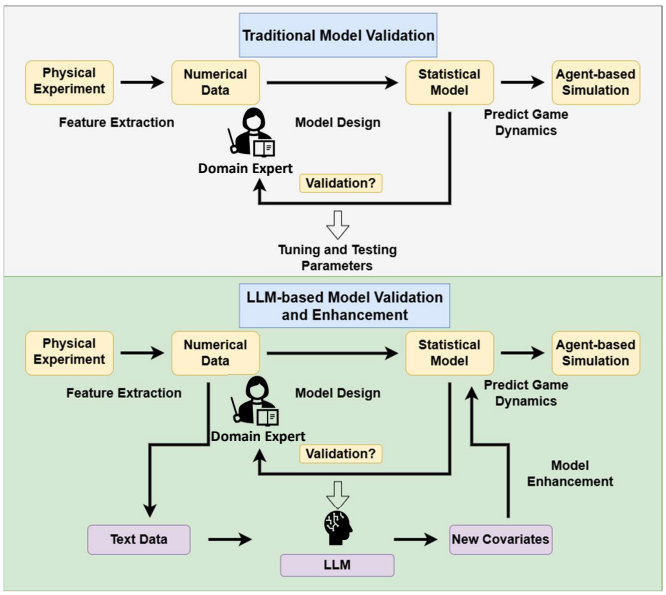


Figure 2: The traditional model validation pipeline vs. proposed LLM-based enhancement pipeline. The traditional pipeline is displayed in the grey box at top while the proposed pipeline is in the green box at the bottom, which is the focus of this work.

received), and (v) numbers of words formed, were generated from the experimental data. Then, these distributions were generated from model predictions of the experimental games, for each of three logistic regression models: $M0$ using all experimental data; $M1$ using only the game data where each player had $d = 2$ neighbors; and $M2$ using all data where the model was parameterized to be polynomial in player degree d. Experimental and model prediction distributions were compared using the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951). The models are listed in increasing levels of sophistication. Model $M2$ exhibited the least values of KL divergence, typically in the range 0.05 to 0.3, where zero indicates perfect agreement between two distributions. Note that KL divergence values can grow much greater than one. In this work, by comparison, we are focusing on the details of the ABM structure.

2.2 Use of LLMs in Agent-based Modeling

Recent study has explored integrating LLMs into ABMs to create LLM-driven agents that simulate human-like decision-making, behavior and communication. This approach, often named "generative agents" or "generative ABMs" allows behaviors to emerge from natural-language prompts instead of hand-coded rules (Xi et al. 2023). A representative example is the work by Park et al. (2023), which simulated a virtual town that resembles a sandbox social world. A recent critical review, however, notes that most such models still rely on informal face-validity checks rather than rigorous empirical tests (Larooij and Törnberg 2025). To address this issue, previous validation-oriented efforts considered LLMs as evaluation assistants. The work of Kleiman et al. (2025) uses an LLM to interpret simulation outputs, while Rios et al. (2024) use ChatGPT to co-design surrogate models and benchmark their predictive accuracy against high-fidelity simulations. Distinct from the previous perspectives, our work utilizes an LLM to validate and augment the underlying ASM for ABM in the collaborative anagram game, which is a novel model validation framework that leads to better model predictive accuracy.

3 METHODOLOGY

This section contains the methods used in the model validation and LLM-based model enhancement described in the bottom half of Figure 2. Section 3.1 introduces alternative models (either with reduced complexity or increased complexity) for comparison with the original MLR model. Section 3.2 provides a detailed explanation of the metrics used to evaluate the models. Section 3.3 proposes a novel pipeline for using LLMs to enhance the existing model.

3.1 Model Validation

As described in Section 1.1, the ASM using multinominal logisitic regression—our agent model of predicting player's next action based on her current action—is originally built by conditioning on the possible now-state and includes four expert-selected covariates. For a general setting with m states and p covariates, the number of parameters to be estimated is $m^2 \times (p+1)$. Each transition from state i to state j is characterized by a parameter vector $\beta_j^{(i)} = (\beta_{j1}^{(i)}, \beta_{j2}^{(i)}, \ldots, \beta_{j(p+1)}^{(i)})^T$ for $i, j = 1, \ldots, m$. In our specific case, $m = 4$ and $p = 4$, resulting in $4 \times 4 \times 5 = 80$ parameters to be estimated. We aim to validate this multinomial logistic regression model by addressing three key questions.

(i) Do the original, expert-defined covariates indeed contribute to modeling player behavior?
(ii) Is the current model complexity *necessary*? In other words, how much of the game's dynamics are lost if we reduce the model's complexity?
(iii) Can we *further improve* predictive performance by introducing additional covariates?

To address the first two questions, we consider several alternative and simpler models (SM) in two general groups. The first group focuses on *models with reduced numbers of covariates*. We use likelihood-based variable selection (Burnham and Anderson 2002) to reduce the covariates from p to some $p_{\text{reduced}} < p$. Consequently, the number of parameters to estimate becomes $m^2 \times (p_{\text{reduced}} + 1)$. Still conditioning on the now-state, we explore the impact of removing 1, 2, and 3 covariates, yielding:

- **3-covariate model (SM-3):** $4 \times 4 \times 4 = 64$ parameters,
- **2-covariate model (SM-2):** $4 \times 4 \times 3 = 48$ parameters,
- **1-covariate model (SM-1):** $4 \times 4 \times 2 = 32$ parameters.

The second group considers *models with simplified structures*. We eliminate the need to condition on the now-state in two scenarios:

- **General model (SM-G):** A single MLR across all data, regardless of the now-state, which requires $m \times (p+1)$ parameters.
- **Two-stage model (SM-T):** First apply a binary logistic regression to distinguish between "idle" and "non-idle" states (yielding $p+1$ parameters). Next, for non-idle predictions, use another MLR over the $m-1$ remaining states (an additional $(m-1) \times (p+1)$ parameters). Summing up gives $m \times (p+1)$ parameters in total.

Note that $m = 4$ and $p = 4$ in our situation. Thus both the general model and the two-stage model yield $4 \times 5 = 20$ parameters.

To address the third question (i.e., (iii) above), we develop an LLM-Guided Covariate Augmentation (LGCA) approach detailed in Section 3.3. After introducing additional covariates, we consider:

- **Full model (AM-F):** MLR using the full set of available covariates to estimate $m^2 \times (\tilde{p}+1)$ parameters, where $\tilde{p} > p$ represents the number of parameters after covariate augmentation.
- **Selected model (AM-S):** MLR using p covariates chosen from the full set of $\tilde{p}$ covariates via variable-selection methods (Burnham and Anderson 2002). Then the model complexity remains the same as the original model.

By systematically comparing these alternative models in terms of several performance metrics (see Section 3.2), we examine whether the *original* model complexity with expert-selected covariates is necessary, or whether it can benefit from different covariates using LGCA.

3.2 Evaluation Metrics

In this subsection, we introduce two metrics to evaluate and compare the performance of different MLR models. The first is a standard one-vs-all Receiver Operating Characteristic (ROC) analysis with a weighted AUC (Sokolova and Lapalme 2009). The second is a novel *mean relative improvement* measure tailored to highlight non-idle prediction in our highly imbalanced data (about 93% of our anagram data are transitions from idle to idle).

Weighted AUC. In a binary classification setting, an ROC curve depicts the trade-off between the true positive rate (TPR) and the false positive rate (FPR) as the decision threshold τ varies. We define

$$\mathrm{TPR}(\tau) \;=\; \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau) \;+\; \mathrm{FN}(\tau)}, \quad \mathrm{FPR}(\tau) \;=\; \frac{\mathrm{FP}(\tau)}{\mathrm{FP}(\tau) \;+\; \mathrm{TN}(\tau)}, \tag{2}$$

where $\mathrm{TP}(\tau)$, $\mathrm{FP}(\tau)$, $\mathrm{TN}(\tau)$, and $\mathrm{FN}(\tau)$ are the numbers of true positives, false positives, true negatives, and false negatives under threshold τ. AUC follows by integrating TPR with respect to FPR from 0 to 1. For a multi-class problem with classes indexed by $c = 1,\ldots,C$, we apply a one-vs-all scheme by treating each class c as positive (and the rest as negative) to obtain an AUC value AUC_c. Let n_c be the number of samples of class c out of the total $n = \sum_{c=1}^{C} n_c$. We define a weight $w_c = n_c/n$ and compute the Weighted AUC as:

$$AUC_{\text{weighted}} \;=\; \sum_{c=1}^{C} w_c \;\cdot\; \mathrm{AUC}_c. \tag{3}$$

In our case, more weights are given to class 1 (idle) versus all other classes (i.e., states), which exactly fits our intention to evaluate a model's ability to capture non-idle events.

Mean Relative Improvement. We propose a customized metric called *mean relative improvement for rare observations* (MRI-RO) to focus on the prediction of *non-idle* Next-States (which are considered rare-observations). Denote a non-idle transition instance (i.e. observation) as ℓ, and suppose the transition is from $a_i(t)$ to $a_j(t+1)$ (i.e., transition from any state i to a non-idle state $j \in \{2,3,4\}$) at time t. Let $\hat{p}_j^{(\ell)}$ be the predicted probability of the observed non-idle transition, and let e_{j_ℓ} (the notation j_l describes the transition at instance ℓ) denote an empirical baseline probability for the same event (for example, the global frequency of the specific non-idle transition to a_j). Define the relative improvement for each observed non-idle instance ℓ as

$$\Delta_\ell = \frac{\hat{p}_j^{(\ell)} - e_{j_\ell}}{e_{j_\ell}}. \tag{4}$$

Suppose there are M such non-idle transition instances in total, then the mean relative improvement is

$$\mathrm{MRI} \;=\; \frac{1}{M} \sum_{\ell=1}^{M} \Delta_\ell. \tag{5}$$

A positive value indicates that, on average, the model assigns a higher likelihood to the actual non-idle transition than the empirical baseline does; a negative value suggests it underperforms the baseline. This parameter measures the model's improvement from a baseline model specified in Section 4 as *null* model, so a larger positive value of MRI indicates a better model performance. To compare two models, say Model 1 and Model 2, we compute

$$\delta_{\mathrm{MRI}} \;=\; (\mathrm{MRI}_1 - \mathrm{MRI}_2)/\mathrm{MRI}_2, \tag{6}$$

which measures the performance advantage of Model 1 over Model 2 in terms of relative likelihood improvement for non-idle transitions. In this work, we adopt 5-fold cross-validation specified in Section 4, and the MRI for each model is calculated by calculating the mean of test data.

The proposed metric highlights the ability of a model to capture rare (non-idle) events more effectively than standard metrics. For example, suppose for a non-idle transition from idle (a_1) to requesting a letter (a_3), the empirical probability is $e_3 = 0.04$. Consider two models M_1 and M_2. Model M_1 predicts $\pi_{1j} = (0.85, 0.05, 0.08, 0.02)$, while Model M_2 predicts $\pi_{1j} = (0.85, 0.07, 0.05, 0.03)$. In this case, standard metrics are unable to capture the nuanced change in M_1's increased prediction for non-idle events. While using MRI, we calculate $\text{MRI}_{M_1} = \frac{0.08-0.04}{0.04} = 1$ and $\text{MRI}_{M_2} = \frac{0.05-0.04}{0.04} = 0.25$. Then $\delta_{\text{MRI}} = \frac{1-0.25}{0.25} = 3$ effectively captures Model 1's improvement from Model 2. The proposed metric is tailored to our highly imbalanced data and it yields a more intuitive pairwise comparison between models by sensitively evaluating the improvement in likelihood.

3.3 LLM-Guided Covariate Augmentation (LGCA)

As shown in the proposed model validation framework (the green box in Figure 2), we aim to explore the possibility of improving the original model's performance by augmenting the covariate space. Our proposed LGCA approach utilizes the semantic and pattern-recognition strengths of generative AI to capture more nuanced dynamics of the experiment. The flowchart of our proposed LGCA approached is presented in Figure 3. In this work, the version of the LLM used is OpenAI's ChatGPT (o1 model). Since this is a closed-source model, no hyperparameters (e.g., temperature, max tokens) were user-configurable beyond the API interface. All the outputs were generated using the platform's default settings.

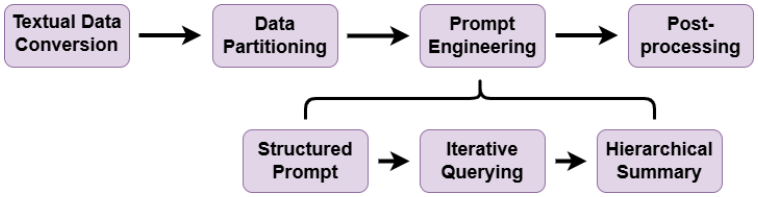


Figure 3: The flowchart for the proposed LLM-Guided Covariate Augmentation (LGCA). The content in each purple box is elaborated in Section 3.3 after the bold texts. This figure corresponds to the purple boxes in Figure 2.

Player ID	Time	Action	Words Formed	...
1	1	Idle	0	
1	...	...	0	
1	20	Form word	1	
...	...	...	1	
1	26	Reply	1	
...	...	...	...	
2	1	Idle	0	
...	...	...	...	

```
Player1: stay idle from 1s to 19s, form a
word at 20s, stay idle from 21s to 25s, reply
letter at 26s, stay idle from 27s to 32s,
reply letter at 33s, stay idle from 34s to
43s, form a word at 44s, stay idle from 45s
to 66s, form a word at 67s, stay idle from
68s to 73s, form a word at 74s, stay idle
from 75s to 77s, form a word at 78s, stay
idle from 79s to 133s, reply letter at 134s,
stay idle from 135s to 181s, request letter
at 182s, stay idle from 183s to 300s
```

Figure 4: An example of data conversion from numerical data to descriptive data in text.

Textual Data Conversion. Figure 4 provides a simple illustration of the data conversion mentioned in the upper left box of Figure 3. In more generalized settings, more detailed descriptions can be added during this conversion. By performing this row-wise conversion and preserving relevant contextual information, we enable the LLM to extract or infer potential latent factors that are not obvious in purely tabular form.

Data Partitioning. After data conversion, challenges arise as the amount of textual data is huge. In our case, there are 210 samples of player action sequences in total. To manage token limits and ensure representative coverage, we split the textual dataset into multiple partitions. Each partition contains a subset of the text descriptions (in our case, each partition contains 1/4 of total observations), aiming to capture a balanced variety of participant behaviors and outcomes.

Prompt Engineering. Within each partition, we design a structured prompt for the LLM to elicit new covariate ideas. Our methodology includes three key steps.

Step 1. **Structured Prompt.** We provide the LLM with clear instructions and a fixed template to ensure consistent responses. Below is an illustrative example of such a prompt:

```
"Context: You are given descriptions of an anagram game, where players will
choose one of the following four actions: "idle", "request letter", "reply letter",
and "form a word" during a 300 second game. Each player has three initial letters
and has neighbors to request letters from.

Multinomial logistic regression is used to model players' behaviors. The following
covariates are used in the model: size of the buffer of letter requests that the
player has yet to reply at time t; number of available letters to form a word at time
t; number of words already formed by the player at time t, number of consecutive time
steps that the player has taken the same action.

Task: Please summarize the potentially uncovered pattern from the data, and propose
    additional variables that might predict a player's decision.

Constraint: The additional variables need to be numerical, tractable, and derivable
from the existing data.

Data: There are 210 players' action sequence in total. You will be given 55 at a time:
     <Player1>, ..., <Player55>."
```

Step 2. **Iterative Querying.** Instead of a single prompt, we use a consistent prompt for each partition of the data. By consistent, we mean that the structure of the prompts is kept the same, and the only difference across partitions is the "Data" section near the end of the prompt.

Step 3. **Hierarchical Summaries.** We employ a multi-tier summarization process. First, we aggregate a first-level summary for each partition by asking the LLM in the "Task" section to produce a concise summary of relevant behavioral patterns along with possible features. Then we merge the summaries from multiple partitions into a single aggregated text. Finally, we present the aggregated summary to the LLM using a structured prompt, requesting an integrated set of potential new covariate suggestions.

Post-processing of LLM Output. Following the prompt-engineering phase, we examine the output from the LLM, which includes the suggested covariates along with their corresponding reasoning and explanations. In the post-processing stage, expert intervention is applied to mitigate potential hallucinations from LLM-generated outputs (Vosoughi 2023). After filtering infeasible ideas that rely on external data not recorded in the experiment, we then map each feasible suggestion into a well-defined numeric variable and proceed to subsequent model evaluation and validation as defined in Sections 3.1 and 3.2.

4 RESULTS

In this section, we compare different models for model validation. In addition, we also consider a *null* model, which simply predicts the next step action using the empirical probability (the global frequency of transitioning to a state), as the baseline model. Specifically, Section 4.1 reports the variable selection results for the models listed in Section 3.1 and the variable augmentation results using the LGCA framework illustrated in Section 3.3. The model evaluation results using the metrics defined in Section 3.2 are presented in Section 4.2.

4.1 Variable Reduction and Expansion

Variable Reduction. Table 1 displays the best-performing models when the number of covariates is successively reduced by one, two, or three under each Now-State (action $a_i(t)$, $i \in \{1,2,3,4\}$), and our goal is to predict the next action (Next-State) $a_j(t+1)$. The four actions are listed in the caption of Figure 1.

For example, in the best three-covariate model with Now-State 1, i.e., $a_1(t)$, meaning the most recent action is idle, the variable Z_W (number of word already formed) is the first to be removed. As we continue reducing covariates, Z_B (size of the request buffer) is the second variable dropped, and Z_C (number of consecutive steps) is the third to be dropped for the idle Now-State. On the other hand, for Now-States 2, 3 and 4, Z_C is the first variable eliminated, indicating that Z_C adds relatively little predictive value in these cases. Then Z_W and Z_L are the next two variables dropped for Now-State 2, while Z_L and Z_B are the next two dropped for Now-States 3 and 4. This sequential removal process identifies which variables contribute least to model prediction before proceeding to evaluate the models' performance.

Table 1: Best reduced-covariate models under each Now-State. "+" indicates the variable is retained in that scenario. The sequence of backward selection process can be read from the table. For example, for the model with player Now-State of 1, the order of variable dropped is: Z_W, Z_B, and Z_C.

Now-State	a_1 (idle)			a_2 (reply)			a_3 (request)			a_4 (form a word)		
Covariates \ Model	SM-3	SM-2	SM-1	SM-3	SM-2	SM-1	SM-3	SM-2	SM-1	SM-3	SM-2	SM-1
Z_B	+			+	+	+	+	+		+	+	
Z_L	+	+	+	+	+		+			+		
Z_W				+			+	+	+	+	+	+
Z_C	+	+										

Covariate Expansion. Next, we augmented the covariate space based on suggestions from our LLM-guided approach. After post-processing model output, four new variables are selected as candidates to be included in the full model:

- X_F: **Time fraction of idle behavior** over the preceding T_{frac} window; here, T_{frac} is chosen to be 60 seconds.
- X_T: **Time elapsed** since the last non-idle transition. If a player has not yet performed a non-idle action, X_T equals the time elapsed from the start of the game.
- X_R: **Reciprocity**, defined as $\frac{\text{cumulative replies}}{\text{cumulative requests}+\delta}$, where δ is a small constant that avoids division by zero.
- X_P: **Time pressure**, measured as the fraction of remaining time over the total game time.

Table 2 summarizes the sequential order in which variables are removed as we reduce the covariate set from eight (the four Z-variables and the four X-variables) back down to four. Notably, Z_C (the original "consecutive time steps" variable) is the first to be dropped under every Now-State, suggesting it contributes relatively little compared with the newly added or remaining original variables. The newly introduced variables X_T and X_R are also dropped relatively early in certain states, whereas X_F constantly remains in the model, indicating its potential importance. Overall, the removal sequence of the original variables aligns closely with the patterns observed in Table 1, reinforcing that Z_C is consistently the least informative.

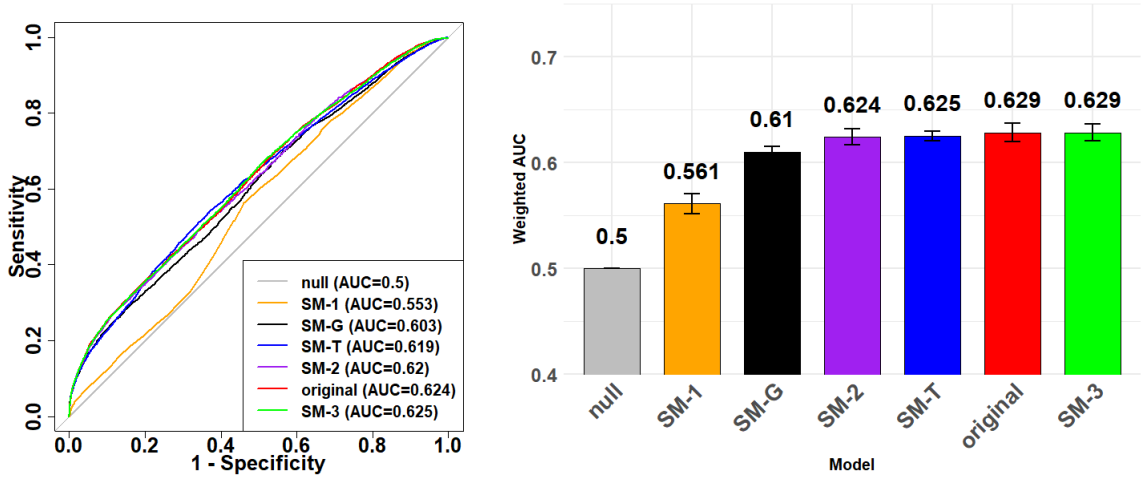
The rationale behind the results in Table 2 is that although Z_C, X_F, and X_T each characterize players' idle behavior, X_F emerges as superior because first, it is a continuous measure, which enables it to capture subtle changes compared to a discrete measure that resets after each non-idle transition; second, it's resilient to fluctuations and noises as looking at a specific time window makes it less sensitive to brief deviations. Additionally, we observe that X_P survives the variable selection process in most scenarios (except when the Now-State is a_1 (i.e., idle) with X_p being removed in the third step). As the game approaches the end, players tend to take more frequent non-idle actions.

Table 2: Order of variable removal when transitioning from 8 down to 4 covariates after LLM-based augmentation. There is almost an equal number of Z and X variables removed: 9 versus 7.

Now-State \ Drop Order	1*st*	2*nd*	3*rd*	4*th*
a_1	Z_C	X_R	X_P	Z_W
a_2	Z_C	X_T	X_R	Z_W
a_3	Z_C	X_T	Z_L	X_R
a_4	Z_C	X_T	Z_L	Z_B

4.2 Performance Evaluation

We employ a 5-fold cross-validation procedure using our proposed evaluation metrics to estimate how well each candidate model generalizes to new data. Each cross validated result is repeated 50 iterations under different sampling to measure the variance. Partitioning the data by player preserves the imbalance structure of the data, reflecting the situation in practice. This process reduces the risk of overfitting and provides more reliable performance estimates.



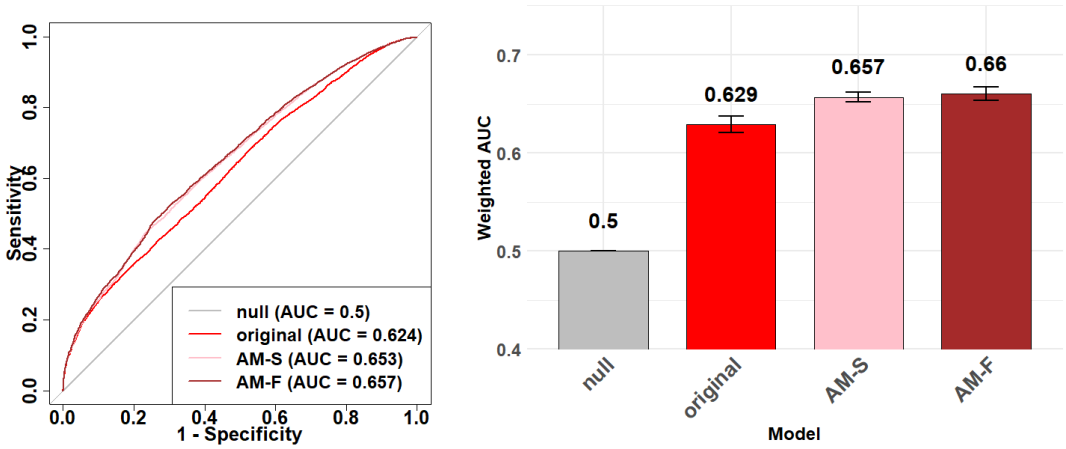
(a) ROC curve of idle (a_1) vs. others.

(b) Weighted AUC.

Figure 5: Cross-validated performance for reduced-complexity models, which correspond to the first five bullets in Section 3.1. Note that the error bars in Figure 5b have been **scaled** by multiplying the standard deviation of weighted AUC values by 50 to ensure visibility. The actual variances are very small.

Figure 5 summarizes the cross-validated performance of the reduced-complexity models using one-versus-all ROC curves and weighted AUC. Both metrics improve substantially from the null model (empirical probability) to the 1-covariate model ($\frac{0.561-0.5}{0.5} = 12.2\%$ increase in weighted AUC), and again from the 1-covariate to the 2-covariate model ($\frac{0.611-0.5}{0.5} = 25.2\%$ increase). The general model (22.2% increase) and two-stage model (25.2% increase) perform comparably to the 2-covariate model. Increasing the covariate number to 3 (26.2% increase) or using all 4 (the original model, also 26.2% increase) yields only marginal additional gains (two-sample t-test yields a p-value of 0.846, which means there is no significant difference between the means of the weighted AUC value over 50 iterations).

Figure 6 illustrates the cross-validated performance of the augmented models derived from the LGCA approach of Section 3.3. The full model with 8 covariates (right-most bar in Figure 6b) achieves a 31.4% improvement over the null model, while the selected model (second-from-right bar in Figure 6b)—which



(a) ROC curve of idle (a_1) vs. others. (b) Weighted AUC.

Figure 6: Cross-validated performance for augmented models via LGCA, which correspond to the last two bullets in Section 3.1. The error bars in Figure 6b have also been **scaled** by multiplying the standard deviation of weighted AUC values by 50 as in Figure 5b. The actual variances are very small.

omits less informative variables—still attains a 30.6% improvement. Notably, at the same level of complexity as the original model, this augmented approach yields an additional $\frac{0.306-0.262}{0.262} = 16.8\%$ performance gain (two-sample t-test yields a p-value of approximately 0, which means there is significant difference between the means of the weighted AUC value over 50 iterations), suggesting that our LGCA-based covariate augmentation effectively enhances predictive accuracy for the anagram game.

Table 3 reports the mean relative improvement (MRI) for each model, along with the pairwise comparison δ_{MRI} relative to the original model. Since MRI emphasizes correct classification of non-idle events, models that condition on Now-States consistently outperform the general and two-stage models by a large margin. For the reduced-covariate models, the results parallel those seen in the weighted AUC: the 3-covariates model is within about 4% of the original model, while the *selected model* under *Enhanced models* outperforms it by 12%. Overall, both the weighted AUC and MRI results indicate that the original MLR design is somewhat redundant and that our LGCA approach effectively improves predictive performance without increasing model complexity.

Table 3: Results of model performance evaluation using MRI and δ_{MRI}. MRI measures the target model's improvement from null model while δ_{MRI} measures the target model's improvement from the original model. δ_{MRI} is defined as $\frac{\text{MRI}_{\text{target}}-\text{MRI}_{\text{original}}}{\text{MRI}_{\text{original}}}$.

Model	Number of Parameters	Conditioning on Now-States?	MRI	δ_{MRI}
Reduced models				
SM-G	20	No	0.573	-0.563
SM-T	20	No	0.596	-0.546
original	80	Yes	1.313	0
SM-3	64	Yes	1.273	-0.039
SM-2	48	Yes	1.228	-0.064
SM-1	32	Yes	1.077	-0.179
Enhanced models				
AM-F	144	Yes	1.524	0.161
AM-S	80	Yes	1.469	0.120

5 SUMMARY

In this paper, we investigated the validation and refinement of a multinomial logistic regression model that predicts player behavior in a collaborative anagram game. Our results indicate that the original, expert-defined covariates are partially redundant, as simpler models can achieve similar predictive accuracy. By contrast, augmenting the model with additional features derived via the proposed LLM-Guided Covariate Augmentation (LGCA) approach leads to improved performance. It demonstrates that large language models can assist in uncovering latent factors that enrich behavioral modeling at comparable levels of complexity. We evaluated various model configurations using both Weighted AUC and the newly proposed Mean Relative Improvement (MRI) metric. The MRI was particularly informative in highlighting each model's ability to capture rare yet significant *non-idle* transitions in the anagrame game. Overall, these analyses confirm that incorporating novel covariates identified by an LLM can substantially enhance model fit without increasing the model complexity with unnecessary parameters. For future research, one can consider fine-tuning the LLM with more detailed anagrams game data, including information such as the exact transactions of players' letters and the exact words players form. Moreover, LLMs hold potential for assisting the design, validation, and enhancement of agent-based models in broader contexts.

REFERENCES

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, *et al*. 2020. "Language Models Are Few-Shot Learners". In *Advances in Neural Information Processing Systems*. December 6th–12th, Virtual Conference, 1877–1901.

Burnham, K. P., and D. R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. 2nd ed. New York: Springer.

Cedeno-Mieles, V., Z. Hu, Y. Ren, X. Deng, A. Adiga, C. L. Barrett, *et al*. 2020. "Networked Experiments and Modeling for Producing Collective Identity in a Group of Human Subjects Using an Iterative Abduction Framework". *Social Network Analysis and Mining* 10(1):1–43.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In *Proceedings of NAACL-HLT 2019*. June 2–7, Minneapolis, Minnesota, USA, 4171–4186.

Kleiman, J., K. Frank, and S. Campagna. 2025. "Simulation Agent: A Framework for Integrating Simulation and Large Language Models for Enhanced Decision-Making". *arXiv preprint arXiv:2505.13761*.

Kullback, S., and R. A. Leibler. 1951. "On information and sufficiency". *The Annals of Mathematical Statistics* 22(1):79–86.

Larooij, M., and P. Törnberg. 2025. "Do Large Language Models Solve the Problems of Agent-Based Modeling? A Critical Review of Generative Social Simulations". *arXiv preprint arXiv:2504.03274*.

Park, J. S., J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. 2023. "Generative Agents: Interactive Simulacra of Human Behavior". In *Proceedings of the 36th ACM Symposium on User Interface Software and Technology*. October 29–November 1, San Francisco, CA, USA.

Ren, Y., V. Cedeno-Mieles, Z. Hu, X. Deng, A. Adiga, C. L. Barrett *et al*. 2018. "Generative Modeling of Human Behavior and Social Interactions Using Abductive Analysis". In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. August 28–31, Barcelona, Spain, 413–420.

Rios, T., F. Lanfermann, and S. Menzel. 2024. ""Large Language Model-Assisted Surrogate Modelling for Engineering Optimization"". In *2024 IEEE Conference on Artificial Intelligence*. June 25–27, Singapore, 796–803.

Sokolova, M., and G. Lapalme. 2009. "A Systematic Analysis of Performance Measures for Classification Tasks". *Information Processing & Management* 45(4):427–437.

Vosoughi, S. 2023. "LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples". *arXiv preprint arXiv:2310.01469*.

Xi, Z., W. Chen, X. Guo, W. He, Y. Ding, B. Hong *et al*. 2023. "The Rise and Potential of Large Language Model Based Agents: A Survey". *arXiv preprint arXiv:2309.07864*.

AUTHOR BIOGRAPHIES

HAO HE is a Ph.D. student in the Department of Statistics at Virginia Tech. His email address is haoh@vt.edu.

XUEYING LIU is a Ph.D. student in the Department of Statistics at Virginia Tech. Her email address is xliu96@vt.edu.

XINWEI DENG is a Professor in the Department of Statistics at Virginia Tech. His e-mail address is xdeng@vt.edu.