The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Can LLMs Reliably Simulate Human Learner Actions? A Simulation Authoring Framework for Open-Ended Learning Environments

Amogh Mannekote[1], **Adam Davies**[2], **Jina Kang**[2], **Kristy Elizabeth Boyer**[1]

[1]University of Florida
[2]University of Illinois Urbana-Champaign
amogh.mannekote@ufl.edu, adavies4@illinois.edu, jinakang@illinois.edu, keboyer@ufl.edu

Abstract

Simulating learner actions helps stress-test open-ended interactive learning environments and prototype new adaptations before deployment. While recent studies show the promise of using large language models (LLMs) for simulating human behavior, such approaches have not gone beyond rudimentary proof-of-concept stages due to key limitations. First, LLMs are highly sensitive to minor prompt variations, raising doubts about their ability to generalize to new scenarios without extensive prompt engineering. Moreover, apparently successful outcomes can often be unreliable, either because domain experts unintentionally guide LLMs to produce expected results, leading to self-fulfilling prophecies; or because the LLM has encountered highly similar scenarios in its training data, meaning that models may not be *simulating* behavior so much as *regurgitating* memorized content. To address these challenges, we propose Hyp-Mix, a simulation authoring framework that allows experts to develop and evaluate simulations by combining testable hypotheses about learner behavior. Testing this framework in a physics learning environment, we found that GPT-4 Turbo maintains calibrated behavior even as the underlying learner model changes, providing the first evidence that LLMs can be used to simulate realistic behaviors in open-ended interactive learning environments, a necessary prerequisite for useful LLM behavioral simulation.

Introduction

Open-ended interactive learning environments offer unique educational value by providing tailored and dynamic spaces where learners can explore, experiment, and construct knowledge-capabilities (Renkl and Atkinson 2007; Hannafin, Land, and Oliver 2013; Land and Jonassen 2012). However, developing these environments is challenging. It requires not only the creation of pedagogical content but also mechanisms to adapt learning experiences for learners with varying knowledge levels and psychological characteristics for very large state spaces due to the relatively open-ended nature of the environments (Kim 2012; Hannafin et al. 2014; Akpanoko et al. 2024). This complexity necessitates an iterative process in which theoretical best practices are continuously balanced with practical demands (Sandoval 2014).

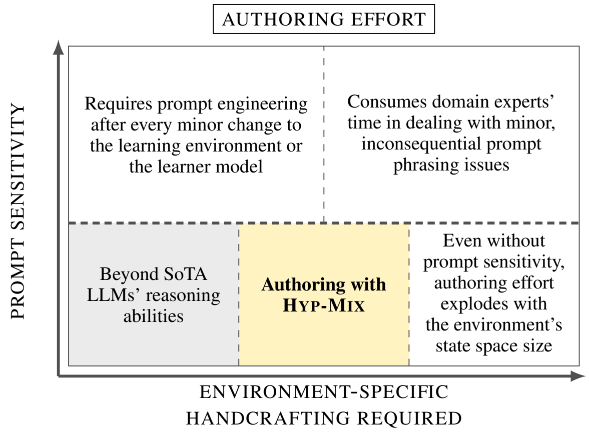
AUTHORING EFFORT
PROMPT SENSITIVITY
ENVIRONMENT-SPECIFIC HANDCRAFTING REQUIRED
Requires prompt engineering after every minor change to the learning environment or the learner model
Consumes domain experts' time in dealing with minor, inconsequential prompt phrasing issues
Beyond SoTA LLMs' reasoning abilities
Authoring with Hyp-Mix
Even without prompt sensitivity, authoring effort explodes with the environment's state space size

Figure 1: We characterize the effort involved in authoring LLM-based simulations of learner behavior as a function of two key attributes of the simulation authoring process: 1) prompt sensitivity and 2) the extent of environment-specific handcrafting required during development. High prompt sensitivity necessitates excessive editing for minor phrasing changes, thus consuming valuable expert time. On the other hand, the need for environment-specific handcrafting arises when an LLM struggles to generalize across learning environments, impeding rapid iteration. The proposed approach of mixing-and-matching expert-hypotheses to define simulation behavior offers a promising balance, enabling authors to impose necessary constraints while leveraging the advantages of state-of-the-art knowledge and reasoning capabilities of LLMs for "filling in the gaps."

Simulations of learner behavior have been instrumental in streamlining the process of developing intelligent systems for education (Koedinger et al. 2015; Matsuda, Cohen, and Koedinger 2015). By allowing developers to rigorously test features before full deployment, these simulations reduce reliance on resource-intensive pilot testing in real-world classrooms (Käser and Alexandron 2023). They enable developers to identify software issues and evaluate design choices early, later fine-tuning the environment to meet learner needs. However, developing simulations during the cold-start phase is challenging due to the lack of real-learner data in new environments. This scarcity prevents purely data-driven approaches, requiring reliance on log data from similar studies, predictions from learning science theories, instructor experience, and expert intuition (Holstein, McLaren, and Aleven 2019). Without action logs from the target demographic, these sources provide the best alternative for accurate simulations.

Combining these alternative information sources to craft realistic simulations of learner actions requires a balanced integration of expert knowledge and automated reasoning. Fully handcrafted, rule-based simulations offer fine-grained control but become impractical as complexity increases, while purely automatic systems may miss critical nuances (Wang et al. 2024). LLM prompting may potentially strike an ideal balance, using rich natural language to specify behavior while leveraging the LLM's reasoning capabilities. This approach holds the potential for flexible, fine-tuned simulations that effectively bridge the gap between manual control and automation.

Promisingly, there has been a recent surge in studies that suggest that LLMs, with their extensive world knowledge and reasoning capabilities, can accurately predict human responses to both natural language descriptions of hypothetical situations and actual experimental setups taken from academic disciplines like psychology and behavioral economics (Aher, Arriaga, and Kalai 2023). However, such claims must be approached with caution. We identify three reasons to be skeptical of simulations based on large language model (LLM) prompting reliably generalizing to new situations.

1. LLMs are known to be highly sensitive to small, inconsequential changes to the prompt wording (prompt sensitivity) (Sclar et al. 2023; Loya, Sinha, and Futrell 2023b). As a result, a simulation that works in one context might fail with changes to either the description of the learning environment (corresponding to, say, a new feature that the developers are planning to add to the environment) or the learner model (corresponding to refinements in the expert's understanding of how learners behave).
2. LLMs, trained on vast web data, may rely on memorization rather than genuine reasoning, limiting their ability to generalize (Sainz et al. 2023).
3. There is no disciplined method to prevent prompt engineers from unconsciously shaping prompts to elicit expected answers, raising concerns of a Clever Hans[1]-like setup, where human cues influence the outcome (Kambhampati 2024).

For the reasons stated above, the usefulness of LLMs for simulating learner actions beyond single proof-of-concept experiments has not yet been established. To address this gap, we introduce a simulation authoring framework that serves the dual purposes of: 1) systematically evaluating whether an LLM-based simulator can usefully generalize to new contexts (e.g., modifications of the original learning environment or the original learner model) without re-engineering the LLM prompt; and 2) establishing a clear prompting workflow to avoid Clever-Hans-style biases, preventing overestimation of the LLM's capabilities.

A robust simulator must dynamically adapt to changes in the simulation context (learning environment or learner model) without extensive prompt recalibration. Once a prompt template is calibrated to specific learner behaviors, this calibration should generalize to new simulation contexts, maintaining consistent simulation behavior. This generalization is important for two reasons: (1) the exponential increase in experiment runs needed as state variables grow, and (2) the limited utility of LLM simulations that only predict behaviors when specifically calibrated, which fails to generate new insights and merely reproduces existing findings (Clever Hans effect).

Our main contribution with HYP-MIX is a systematic simulation authoring framework[2] for incorporating expert knowledge into LLM-based simulations of learner actions. Our hypothesis-based framework presents a well-defined, statistical notion of what it means for the simulation to be robust and generalizable to new simulation scenarios. Using our framework, we find that GPT-4 Turbo is capable of maintaining prompt calibration under changes to the learner model, indicating that it may already be feasible to simulate realistic learner behaviors in learning environments using frontier LLMs.

Related Work

Simulated Learner Behavior for Authoring Educational Technologies. Simulated learners streamline the authoring of intelligent tutoring systems (ITSs), which often require over 100 hours of work per instructional hour (Blessing and Gilbert 2008). Tools like SimStudent (Matsuda, Cohen, and Koedinger 2015) simulate learner behavior to aid in ITS development via interactive tutoring. However, compared to ITSs, open-ended interactive learning environments typically involve more states due to their open-ended and exploratory nature and a greater emphasis on scaffolding the affective aspects of learning (Rieber 1996). While Christensen et al. (2011) simulate psychological aspects of learners, their method is handcrafted, highly context-specific, and therefore, would not scale well to complex interactive environments. To our knowledge, our work is the first to apply learner behavior simulations to these environments. Additionally, Käser and Alexandron (2023) identify a widespread lack of validation in simulated learner research, which we address in the HYP-MIX framework by centering on falsifiable hypotheses for both authoring and evaluation. Our approach is also in line with Ainsworth and Grimshaw (2004), who focus on group-level behavior specification, similar to our use of distributional hypotheses.

Simulating Human Behavior with LLMs Several recent works explore the ability of LLMs to simulate human behaviors across various contexts, including social platform design (Park et al. 2022), market research (Brand, Israeli, and

[1] The term originates from Hans, a horse in early 20th century Germany, who seemed to perform arithmetic by tapping his hoof. It was later found he was responding to subtle cues from his trainer or the audience.

[2] https://github.com/msamogh/hypmix

Ngwe 2023), and experimental economics (Gui and Toubia 2023). LLMs have also been shown to reflect human-like cognitive biases in reasoning tasks (Dasgupta et al. 2022; Ozeki et al. 2024). Most related to our work are studies that analyze LLMs agents' consistency with provided personality traits (Frisch and Giulianelli 2024; Jiang et al. 2024) or character profiles (Xiao et al. 2023). However, in contrast to these works, we evaluate agent consistency using simple hypotheses specifying the statistical relationship between values of agent (learner) characteristics and behaviors, alleviating the need for fine-grained annotation of individual responses; and further consider how these simulated behaviors change in response to changes in the simulation context.

Prompt Sensitivity and Prompt Calibration. Experiments using LLMs rely heavily on natural language prompts to define personas, situations, and tasks, but LLMs are highly sensitive to slight variations in prompt text, making this a critical issue for research (Mohammadi 2024). Loya, Sinha, and Futrell (2023a) find that ChatGPT exhibits sensitivity to prompt phrasing for decision-making tasks such as ours. In response, various prompt calibration approaches have emerged, particularly focusing on reducing the LLMs' sensitivity to the order of in-context examples (Lu et al. 2022; Zhao et al. 2021). In contrast to this family of work that focuses on reducing variance between different templates, in this work we our goal is to test the consistency of LLM behaviors across different simulation contexts.

The HYP-MIX Framework

The HYP-MIX framework is designed to create and evaluate realistic and scalable simulations of learner behavior by translating theoretical constructs into concrete, testable predictions. The unit of authoring and evaluation in this framework is a Marginalized Distributional Hypotheses (MDHyp). These are called "marginal" because they focus on one learner characteristic at a time, while "marginalizing" over all other variables. This is essential because, while it is straightforward to reason about a single characteristic, jointly considering multiple characteristics can quickly become difficult. For instance, an MDHyp might predict that low persistence leads to a higher probability of task-abandonment, focusing specifically on persistence while accounting for other variables in the background. The rest of this section details the motivation and implementation of MDHyps, along with its integration with LLM prompting.

MDHyps for Simulation Evaluation

A common method for validating simulated agents involves presenting the generated behaviors to human crowdworkers, who then rate the realism of these behaviors either over a quantitative scale or according to a qualitative rubric (Park et al. 2023; Jiang et al. 2024). While this approach has been widely adopted in recent studies, particularly with the proliferation of crowdsourcing platforms, it is fundamentally limited and ill-suited for evaluating simulated learner actions in complex, iterative experiments, for several reasons:

1. **Cost Constraints:** Crowdsourcing becomes prohibitively expensive in iterative studies, particularly those requiring extensive experimentation.
2. **State Space Explosion:** As the complexity of the environment and the number of learner characteristics increase, the task of collecting annotations for every possible combination becomes infeasible.
3. **Demographic Mismatch:** The typical crowdworker populace does not include individuals deeply involved in education, such as researchers or educators (Huff and Tingley 2015). As a result, they are generally not equipped to accurately assess the realism of behaviors exhibited by young learners with specific characteristics.
4. **Inherent Noise in Learners' Actions:** The stochastic nature of interactions within learning environments introduces significant noise into the evaluation process. Even with a sophisticated model of a learner, it is nearly impossible to predict with certainty how they will behave in a given situation, making deterministic point estimates unreliable.

We propose using MDHyps to evaluate learner behavior simulations at a distributional level, drawing from prior studies or instructor experience. An MDHyp is a natural language statement that describes a relationship between a learner's characteristic and their probability of taking certain actions (e.g., "*a more persistent learner is less likely to abandon the task as more time passes*"). This relationship can be tested by analyzing the distribution of outcomes from multiple simulation runs across different environment states.

MDHyps for Simulation Authoring

The central thesis of the HYP-MIX framework is that MDHyps serve not only as useful tools for *evaluating* an existing simulation, but also as powerful building blocks for *expert-authoring* LLM-based simulations of learner behavior.

Achieving Mix-and-Match Simulation Authoring with MDHyps For MDHyps to be effective in prompt-based simulation authoring, the LLM must demonstrate compositional generalization (Mannekote 2024). We need MDHyps to function as modular elements that can be easily added, edited, removed, swapped, and combined to shape the LLM's outputs. Similar to SKILL-MIX (Yu et al. 2023), which tests LLMs' ability to combine literary and logical devices to generate free-form text, HYP-MIX tests LLMs' ability to combine calibrated expert-hypotheses to simulate learner actions in a "calibrate once, use forever" fashion. Achieving this, of course, is challenging due to LLMs' sensitivity to prompt phrasing and requires empirical validation.

Existing Notions of "Calibration" The term "calibration" carries different definitions across disciplines. In statistics and machine learning, calibration refers to the alignment between a model's predicted probabilities and the actual observed frequencies of outcomes, ensuring that predictions accurately reflect real-world occurrences over time (Bella et al. 2010). In the context of physical measurement devices, calibration aims to ensure that a measurement device's accuracy is consistent. This process involves aligning the device with a known standard to maintain reliable accuracy across future measurements (Castrup et al. 1994). The

HYP-MIX notion of calibration combines the two: we want the predicted action probabilities from the LLM to align with the MDHyp (analogous to the statistical notion) and also to hold this calibration across different hypotheses and changes in the underlying learner model (analogous to the metrological notion).

Holding Calibration Building on this integrated definition, the ability of an LLM to hold calibration of a prompt template across simulation contexts is critical for minimizing the labor-intensive re-engineering of prompt templates after each modification to the simulation model. By grouping hypotheses into *hypothesis classes* based on similar functional relationships and linking them to specific statistical tests, we aim to ensure robust calibration, even as the simulation model undergoes modifications.

Hypothesis Classes A hypothesis class defines a specific functional relationship that its member-hypotheses posit between independent variables (e.g., learner persona values, environment state variables) and a dependent variable (e.g., probability mass of specific learner actions). Formally, a hypothesis H_i belongs to the hypothesis class $\mathcal{H}_{\text{class}}$ (denoted as $c(H_i) = \mathcal{H}_{\text{class}}$). Each hypothesis class is associated with a prompt template, $\hat{I}_{\text{class}}$, that its member-hypotheses instantiate by specifying slot values, and is linked to a specific success criterion T_{class}, typically expected to be the result of a statistical test (e.g., Chi-squared) designed to assess how well the LLM maintains consistency and accuracy when different instances of that class's characteristic relationship are tested (e.g., any relationship that can be expressed in natural language, such as linear, logarithmic, or piecewise continuous relationships).

Template Calibration Calibration in HYP-MIX applies at the level of hypothesis classes, where a calibrated prompt template serves as the operationalization of the class's statistical relationships. The process involves iteratively refining the template using well-established prompt engineering strategies, such as rephrasing instructions to emphasize critical behavior patterns identified from prior iterations and clarifying ambiguities. Decisions regarding prompt modifications were grounded in observed discrepancies between desired and simulated outputs. Successful calibration aims to ensure that the template—and by extension, the hypothesis class—remains robust across changes to the learner model, swapped variables, or new member-hypotheses, minimizing the need for re-engineering.

Experiments

We test the robustness and generalization capabilities of GPT-4 Turbo, a state-of-the-art LLM, in the HoloOrbits environment by assessing how well it maintains calibration when the learner model is modified.

Learning Environment

We situate our experiments within HoloOrbits (Rajarathinam, Palaguachi, and Kang 2024), an open-ended interactive learning environment designed for teaching Kepler's Laws that we use for our experiments (see Figure 2). We selected HoloOrbits due to its small, well-defined state and action spaces, which make it ideal for preliminary experiments. Since our experiments involve a text-based LLM, we only need a textual description of the learning environment to feed into the model as a natural language prompt. An unintended advantage of this approach is that it allows us to describe learning environments not yet implemented in software.

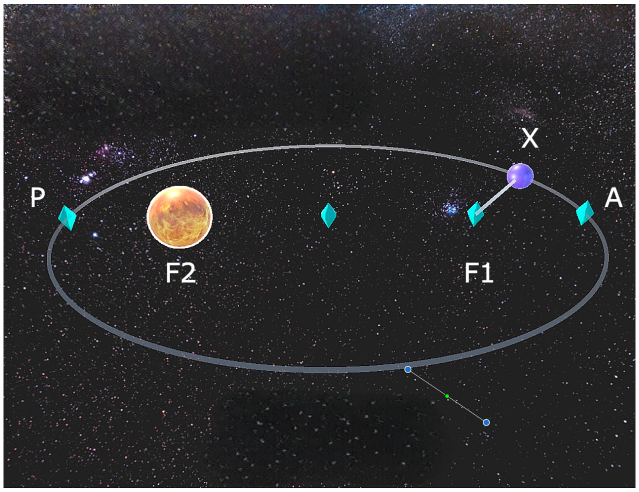

Figure 2: A screenshot of the original HoloOrbits environment (Rajarathinam, Palaguachi, and Kang 2024) with the keypoints annotated.

Learning Task We particularly focus on the learner's task to verify if a given planetary system adheres to Kepler's First Law by submitting three equal arithmetic expressions. These expressions can use any combination of distance measurements between the following points: aphelion (A), perihelion (P), focus 1 (F1), focus 2 (F2), and a fixed point on the orbit (X). The correct solution involves submitting the following measurements: **(F1-A + F2-A)**, **(F1-P + F2-P)**, and **(F1-X + F2-X)**.

State Representation For our experiments, we define a minimal state representation with ten boolean variables indicating whether the learner has measured the distances between each pair of points. Additionally, we include two integer variables to track the number of submission attempts and the time elapsed since the session began, respectively.

Action Space The learner can perform measurements between any pairs of key points in the planetary system, with specific actions such as **MEASURE-F1-X** to measure the distance between Focus 1 (F1) and a fixed point on the orbit (X), **MEASURE-A-F1** to measure the distance between Aphelion (A) and Focus 1 (F1), or **MEASURE-A-P** to measure the distance between Aphelion (A) and Perihelion (P). In addition to these measurement actions, the learner can submit solutions using **SUBMIT(X, Y, Z)**, where X, Y, and Z represent arithmetic expressions involving the measured distances. The goal is for all three expressions to be equal. The learner also has the option to **EXIT** at any time.

Hypothesis $(H_i \in H_{c(H_i)})$	**Initial, Uncalibrated Prompt Template** $\hat{I}_{c(H_i)}(H_i)$
$H_{G1} \in \mathcal{H}_{\text{mono}}$	A learner with a higher **geometry proficiency** is more likely to **make productive measurements** (i.e., **those that measure distances between pairs of points in the planetary system that are potentially useful to verify if the orbit is elliptical**). To **make productive measurements** is to make one of the following actions: **MEASURE-F1-X, MEASURE-F2-X, MEASURE-F1-P, MEASURE-F2-P, MEASURE-A-F1, MEASURE-A-F2**.
$H_{P1} \in \mathcal{H}_{\text{mono}}$	A learner with a higher **persistence** is **less** likely to **abandon the task as the number of measurements increases** (i.e., **to prematurely exit the session before submitting the right solution**). To **abandon the task as the number of measurements increases** is to make one of the following actions: **EXIT**.
$H_{P2} \in \mathcal{H}_{\text{mono}}$	A learner with a higher **persistence** is **less** likely to **abandon the task as the time elapsed increases** (i.e., **to prematurely exit the session before submitting the right solution**). To **abandon the task as the time elapsed increases** is to make one of the following actions: **EXIT**.
$H_{G2} \in \mathcal{H}_{\text{uniform}}$	As learners get closer and closer to the **lower** end of the **geometry proficiency** spectrum (value of 1), they are equally likely to perform the following actions. In other words, such a learner exhibits a uniform distribution over these actions: **<ALL MEASUREMENT ACTIONS>**.

Table 1: Hypotheses used in learner modeling experiments. Regular text shows the template for each hypothesis class, with **blue** indicating specific slot values for each hypothesis. Updated calibrated prompt templates are in the Appendix.

Learner Model

We represent each learner through a learner model $\mathcal{L} = (\mathcal{C}, \mathcal{V}, \mathcal{M})$. C is the set of **learner characteristics** (e.g., geometry proficiency, persistence) being modeled. $\mathcal{V}$ is the mapping between each learner characteristic, $C_i \in \mathcal{C}$, to its corresponding **persona level**, V_i, of the current learner. Each $V_i \in \mathcal{V}$ is quantified on a numerical scale ($V_i \in [1, 10]$). Finally, each learner characteristic C_i is associated with a **learner characteristic model** $M_i \in \mathcal{M}$, which, in turn, comprises one or more MDHyps.

Learner Characteristics For the HoloOrbits learning environment, we model learners using persistence (a psychological factor) and geometry proficiency (which reflects the learner's knowledge of the subject matter) with the following operating theoretical definitions:

- **Persistence:** "maintaining a sustained effort toward completion of a goal-directed task despite challenges or difficulties" (Anderson 2002; Hilton and Pellegrino 2012)
- **Geometry Proficiency:** "the ability to apply the knowledge of the properties of common shapes to solve problems" (Jablonski and Ludwig 2023)

Design of the LLM Prompt for the Simulation The simulation prompt ($\hat{I}_{\text{sim}}$) consists of introductory instructions, a description of the learning environment, current state, and the learner model (a graphic of the prompt template is shown in the Appendix). Furthermore, the prompt template for each learner characteristic model is a concatenation of prompt templates of the MDHyps that make up the learner characteristic model. We also instruct the LLM to perform Chain-of-Thought reasoning (Wei et al. 2022) before outputting the simulated action to strengthen the reasoning and provide the practitioners with a semblance of the intermediary steps used to arrive at the output, which can then be used to refine the simulation.

Approximate Marginalization Testing an MDHyp by running the simulation over all value-assignments of state variables $\mathcal{S}$ requires an intractable number of LLM calls that grows exponentially with $|\mathcal{S}|$. To address this, we statistically approximate the state space by subsampling it. This approach allows for manageable marginalization while controlling computational costs.

Learner Model Edit Graph: A Case Study

This section details a case study of how we developed a simulation of learner actions for the HoloOrbits environment leveraging the HYP-MIX framework. The goal is to demonstrate how HYP-MIX can evaluate the compositional generalization capabilities of an LLM (we used GPT-4 Turbo (Achiam et al. 2023) in our experiments). We focus on five representative types of edits to the learner model—Ex-Situ Transfer, Combine Hypotheses, Variable Swap, LC Swap, and Calibration Regression (defined in detailed in Figure 3)—which reflect the iterative process a developer might follow when constructing a learner model. Throughout development, we use the four MDHyps listed in Table 1. These modifications are represented via a Learner Model Edit Graph (Figure 3).

1. **Initial Hypotheses and Operationalization:** We initialize the learner model with two hypotheses: H_{G1} and H_{P1}, both obtained by operationalizing the theoretical definitions of geometry proficiency and persistence respectively into MDHyps (see Table 1 for all hypotheses used). Both H_{G1} and H_{P1} posit *monotonic* relationships between variables. We grouped them under the hypothesis class $\mathcal{H}_{\text{mono}}$. We calibrated $\hat{I}_{\text{mono}}$ using H_{G1} as the calibration reference hypothesis and tested for generalization on H_{P1}. We define the success criteria function for monotonic hypotheses, T_{mono} using the Spearman correlation coefficient ρ and its corresponding p-value P_ρ as follows:

$$T_{\text{mono}}(\rho, P_\rho) = \begin{cases} \text{TRUE}, & \text{if } \rho > 0 \text{ and } P_\rho \leq 0.05 \\ & \text{for a monotonically} \\ & \text{increasing hypothesis} \\ \text{TRUE}, & \text{if } \rho < 0 \text{ and } P_\rho \leq 0.05 \\ & \text{for a monotonically} \\ & \text{decreasing hypothesis} \\ \text{FALSE}, & \text{otherwise} \end{cases} \tag{1}$$

For H_{G1}, Spearman correlation is computed between the persona value for geometry proficiency and empirical probability of making a productive measurement.

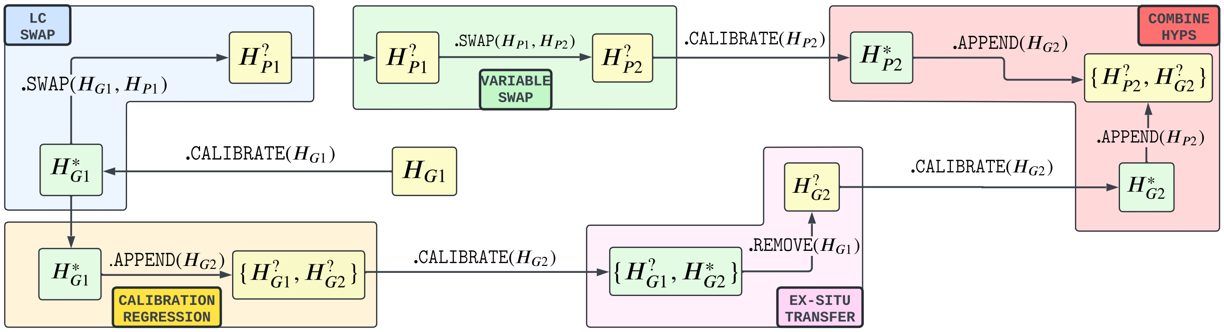
LC SWAP
$.\texttt{SWAP}(H_{G1}, H_{P1})$
$H^?_{P1}$
$H^?_{P1}$
$.\texttt{SWAP}(H_{P1}, H_{P2})$
$H^?_{P2}$
VARIABLE SWAP
$.\texttt{CALIBRATE}(H_{P2})$
H^*_{P2}
$.\texttt{APPEND}(H_{G2})$
COMBINE HYPS
$\{H^?_{P2}, H^?_{G2}\}$
$.\texttt{APPEND}(H_{P2})$
H^*_{G1}
$.\texttt{CALIBRATE}(H_{G1})$
H_{G1}
$.\texttt{CALIBRATE}(H_{G2})$
$H^?_{G2}$
H^*_{G2}
$.\texttt{REMOVE}(H_{G1})$
H^*_{G1}
$.\texttt{APPEND}(H_{G2})$
$\{H^?_{G1}, H^?_{G2}\}$
CALIBRATION REGRESSION
$.\texttt{CALIBRATE}(H_{G2})$
$\{H^?_{G1}, H^*_{G2}\}$
EX-SITU TRANSFER

Figure 3: Learner Model Edit Graph used to evaluate LLM robustness across five distinct edit operations. Nodes represent learner model "snapshots" after developer edits, listing the MDHyps in each snapshot. Green nodes indicate calibrated snapshots; yellow nodes are untested for calibration. Each MDHyp in the learner model is annotated: '?' for untested calibration status and '*' for confirmed calibration. **(1) Ex-Situ Transfer:** Tests if a calibrated MDHyp alongside other MDHyps remains calibrated when tested alone. **(2) Combine Hypotheses:** Assesses stability of two separately calibrated hypotheses when combined. **(3) Variable Swap:** Swaps a single variable within a hypothesis. **(4) LC Swap:** Evaluates if a prompt template calibrated for one learner characteristic works for another in the same class. **(5) Calibration Regression:** Tests stability of a calibrated hypothesis when a new one is added.

2. **Variable Swap:** After consulting with learning science experts, we determined that the "Number of Submissions" was a more suitable measure of "challenge" than "Number of Minutes Elapsed." This led to a modification of the original MDHyp for persistence, resulting in a new hypothesis, H_{P2}.
3. **Append:** During testing, we observed that learners with minimal Geometry Proficiency (1/10) were unexpectedly producing a high percentage ($\sim$80%) of productive measurements, contrary to our expectation of a uniform distribution[3]. To address this, we introduced H_{G2} and a new hypothesis class, $\mathcal{H}_{\text{uniform}}$, to explicitly model this behavior and refine the Geometry Proficiency model, better align it with our theoretical expectations. The success criteria function T_{uniform} for the uniform distribution hypothesis is defined using the p-value of a Chi-squared test P_{χ^2} as follows:

$$T_{\text{uniform}}(P_{\chi^2}) = \text{TRUE if } P_{\chi^2} > 0.05 \text{ else FALSE} \quad (2)$$

4. **Combine Hypotheses:** After calibrating the prompt templates for H_{G2} and H_{P2}, we combined these MDHyps into a unified learner model, completing the development process.

Results and Discussion

To evaluate whether an LLM supports flexible simulation authoring in the HYP-MIX framework, the calibration state of all hypothesis classes must remain intact after a learner model edit. Specifically, the LLM outputs must continue satisfying the success criterion function $T_{c(H_i)}$ for each hypothesis H_i without altering the associated prompt template, $\hat{I}_{c(H_i)}$. This is assessed by comparing $T_{c(H_i)}$ outputs before (Pre-Op) and after (Post-Op) the edit operation, across three distinct action space labelings (e.g., changing EXIT to QUIT) to account for variability. Each operation in Table 2 is evaluated using three rows (details in the Appendix).

For example, consider H_{G1}, which predicts that the learner's probability of making productive measurements increases monotonically with geometry proficiency. Since $H_{G1} \in \mathcal{H}_{\text{monotonic}}$, we calibrate the template $\hat{I}_{\text{monotonic}}$ using H_{G1} until the monotonicity test $T_{\text{monotonic}}$ is satisfied. After calibrating H_{G1}, we modify the learner model by adding a new hypothesis, H_{P1}, and reapply $T_{\text{monotonic}}$ to H_{G1} with H_{P1} included in the LLM prompt. We then report whether $\hat{I}_{\text{monotonic}}$ retains its calibration (see Table 2). This procedure is repeated for all directed edges in the graph.

Except for the COMBINE operation, where calibration failed in two of three action spaces, GPT-4 Turbo successfully maintained calibration across the other four learner model edit operations (Table 2). This demonstrates its ability to generalize to new learner models without requiring re-calibration in most cases (16 of 18), minimizing manual effort and enabling novel insights. The COMBINE failures likely stem from increased prompt complexity or interference between overlapping hypotheses, which challenges the model's compositional generalization. Further investigation is needed to isolate these factors. Nevertheless, GPT-4 Turbo shows strong stability across other multi-hypothesis operations, underscoring its overall reliability.

While further experimental evidence is needed to generalize these claims across learning environments, learner characteristics, and LLMs, our results show promise for using MDHyps as a unit of simulation-authoring with current LLM technology. Balancing explicit and implicit authoring of agent simulations involves deciding which specific agent behaviors must be defined manually and which can be left for the LLM to handle automatically. In sensitive domains

[3] We hypothesize this result to be a result of a more general phenomenon that Aher, Arriaga, and Kalai (2023) refer to as "hyper-accuracy distortion," where LLMs struggle to feign ignorance about a topic to simulate human behavior.

Operation	Learner Model Transformation (Test MDHyp in Bold)	Success Criterion Function	Pre-Op Inputs for Test Results	Post-Op Inputs for Test Results	Is Calibration Held Post-Op?
EX-SITU TRANSFER	$\{H^?_{G1}, \mathbf{H}^*_{G2}\} \rightarrow \{\mathbf{H}^?_{G2}\}$	$T_{\text{uniform}}(P_{\chi^2})$ (Eq. 2)	$P_{\chi^2} = 0.98$	$P_{\chi^2} = 0.98$	Held
			$P_{\chi^2} = 0.46$	$P_{\chi^2} = 0.46$	Held
			$P_{\chi^2} = 0.22$	$P_{\chi^2} = 0.22$	Held
COMBINE HYPOTHESES	$\{\mathbf{H}^*_{P2}\} \rightarrow \{H^?_{P2},\mathbf{H}^?_{G2}\}$	$T_{\text{mono}}(\rho, P_\rho)$ (Eq. 1)	ρ = -0.7,P_ρ=.02	ρ = -0.3,P_ρ=.41	Lost
			ρ = -0.7,P_ρ=.02	Constant series	Lost
			ρ = -0.7,P_ρ=.02	ρ = -0.6,P_ρ=.02	Held
	$\{H^*_{G2}\} \rightarrow \{H^?_{P2}, H^?_{G2}\}$	$T_{\text{uniform}}(P_{\chi^2})$ (Eq. 2)	$P_{\chi^2} = 0.98$	$P_{\chi^2} = 0.98$	Held
			$P_{\chi^2} = 0.46$	$P_{\chi^2} = 0.46$	Held
			$P_{\chi^2} = 0.22$	$P_{\chi^2} = 0.22$	Held
VARIABLE SWAP	$\{H^*_{P1}\} \rightarrow \{\mathbf{H}^?_{P2}\}$	$T_{\text{mono}}(\rho, P_\rho)$ (Eq. 1)	ρ = -0.8,P_ρ<.01	ρ = -0.8,P_ρ<.01	Held
			ρ = -0.8,P_ρ<.01	ρ = -0.8,P_ρ<.01	Held
			ρ = -0.8,P_ρ<.01	ρ = -0.8,P_ρ<.01	Held
LC SWAP	$\{\mathbf{H}^*_{G1}\} \rightarrow \{\mathbf{H}^?_{P1}\}$	$T_{\text{mono}}(\rho, P_\rho)$ (Eq. 1)	ρ = 0.6,P_ρ = .04	ρ = -0.8,P_ρ<.01	Held
			ρ = 0.6,P_ρ = .04	ρ = -0.8,P_ρ<.01	Held
			ρ = 0.7,P_ρ = .01	ρ = -0.8,P_ρ<.01	Held
CALIBRATION REGRESSION	$\{\mathbf{H}^*_{G1}\} \rightarrow \{\mathbf{H}^?_{G1}, H^?_{G2}\}$	$T_{\text{mono}}(\rho, P_\rho)$ (Eq. 1)	ρ = 0.6,P_ρ = .04	ρ = 0.9,P_ρ<7e-6	Held
			ρ = 0.6,P_ρ = .04	ρ = 0.8,P_ρ<.01	Held
			ρ = 0.7,P_ρ = .01	ρ = 0.7,P_ρ = .01	Held

Table 2: Results of statistical tests evaluating the impact of different operations on the calibration state of hypotheses within the learner model. The table compares pre-operation (Pre-Op) and post-operation (Post-Op) results using Chi-squared and Spearman correlation tests, conducted across three different labelings of the action space for improved reliability. The operations include Ex-Situ Transfer, Combine Hypotheses, Variable Swap, Learner Characteristic (LC) Swap, and Calibration Regression. For each operation, the table provides the specific hypotheses tested, the applied statistical test, and the resulting p-values. Bolded hypotheses indicate those tested in both the pre- and post-op phases. Green shading denotes stable test results (holding calibration), red shading shows a total loss of calibration, and yellow shading indicates that the MDHyp is satisfied post-operation, though with some degradation in statistical significance.

like education, a bias toward explicit authoring is prudent (Tian et al. 2024), as LLMs struggle with certain reasoning tasks (Huang and Chang 2022; Kambhampati 2024; Kambhampati et al. 2024). The MDHyps and Learner Model Edit Graph abstractions offer a foundation for building benchmark datasets that evaluate LLM performance across different learner characteristics and multiple learning environments.

Limitations and Future Work

Our study examines two learner characteristics—geometry proficiency and persistence—in a single learning environment. While this limited scope enables targeted insights and method refinement, it may miss complex, non-linear relationships that require real learner data for accurate calibration (Klein-Latucha and Hershkovitz 2024). LLMs-simulated behaviors help address data scarcity but may lack the stochasticity and nuanced diversity of real learners. Expert validation of simulated outputs and benchmarking against real learner logs, where available, are crucial for mitigation. Future work should empirically test the reliability of HYP-MIX predictions by comparing them with real learner behaviors. Additionally, addressing the ethical and practical challenges of data representativeness is critical for ensuring broader applicability and trustworthiness of the framework. Further research could explore alternative LLMs, including open-source models, and evaluate the HYP-MIX framework across a wider range of diverse learning environments.

Conclusion

We present the HYP-MIX framework, which uses Marginal Distributional Hypotheses (MDHyps) to simulate learner actions in open-ended interactive learning environments, reducing the cost and time of real-world testing We show that GPT-4 Turbo maintains calibration across various types of modifications to the learner model, reducing the need for frequent recalibration and highlighting the potential of LLMs for behavioral simulation. Our key contribution is a scalable method for leveraging LLMs to enhance the adaptability of open-ended interactive learning environments. By addressing prompt sensitivity through modular hypothesis calibration and compositional generalization, HYP-MIX enables robust, generalizable simulations with minimal re-engineering effort.

Acknowledgements

We thank ChengXiang Zhai for his valuable feedback in developing the ideas behind this work. This material is based upon work supported by the National Science Foundation and the Institute of Education Sciences under Grant #2229612 (National AI Institute for Inclusive Intelligent Technologies for Education). Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation or the U.S. Department of Education.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aher, G.; Arriaga, R. I.; and Kalai, A. T. 2023. Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies. ArXiv:2208.10264 [cs].

Ainsworth, S.; and Grimshaw, S. 2004. Evaluating the REDEEM authoring tool: can teachers create effective learning environments? *International Journal of Artificial Intelligence in Education*, 14(3-4): 279–312.

Akpanoko, C. E.; S., A. T.; Cordell, G.; and Biswas, G. 2024. Investigating the Relations between Studentsâ€™ Affective States and the Coherence in their Activities in Open-Ended Learning Environments. In PaaÃŸen, B.; and Epp, C. D., eds., *Proceedings of the 17th International Conference on Educational Data Mining*, 511–517. Atlanta, Georgia, USA: International Educational Data Mining Society. ISBN 978-1-7336736-5-5.

Anderson, P. 2002. Assessment and development of executive function (EF) during childhood. *Child neuropsychology*, 8(2): 71–82.

Bella, A.; Ferri, C.; Hernández-Orallo, J.; and Ramírez-Quintana, M. J. 2010. Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 128–146. IGI Global.

Blessing, S. B.; and Gilbert, S. 2008. Evaluating an authoring tool for model-tracing intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*, 204–215. Springer.

Brand, J.; Israeli, A.; and Ngwe, D. 2023. Using GPT for market research. *Harvard Business School Marketing Unit Working Paper*, (23-062).

Castrup, H. T.; Eicke, W. G.; Hayes, J. L.; Mark, A.; Martin, R. E.; and Taylor, J. L. 1994. Metrology: Calibration and measurement processes guidelines. *NASA STI/Recon Technical Report N*, 95: 18745.

Christensen, R.; Knezek, G.; Tyler-Wood, T.; and Gibson, D. 2011. SimSchool: An online dynamic simulator for enhancing teacher preparation. *International Journal of Learning Technology*, 6(2): 201–220.

Dasgupta, I.; Lampinen, A. K.; Chan, S. C.; Creswell, A.; Kumaran, D.; McClelland, J. L.; and Hill, F. 2022. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*.

Frisch, I.; and Giulianelli, M. 2024. LLM Agents in Interaction: Measuring Personality Consistency and Linguistic Alignment in Interacting Populations of Large Language Models. ArXiv:2402.02896 [cs] version: 1.

Gui, G.; and Toubia, O. 2023. The challenge of using llms to simulate human behavior: a causal inference perspective. *ArXiv*, abs/2312.15524.

Hannafin, M.; Land, S.; and Oliver, K. 2013. Open learning environments: Foundations, methods, and models. In *Instructional-design theories and models*, 115–140. Routledge.

Hannafin, M. J.; Hill, J. R.; Land, S. M.; and Lee, E. 2014. Student-centered, open learning environments: Research, theory, and practice. *Handbook of research on educational communications and technology*, 641–651.

Hilton, M. L.; and Pellegrino, J. W. 2012. *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press.

Holstein, K.; McLaren, B. M.; and Aleven, V. 2019. Co-designing a real-time classroom orchestration tool to support teacher-AI complementarity. *Grantee Submission*.

Huang, J.; and Chang, K. C.-C. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Huff, C.; and Tingley, D. 2015. “Who are these people?” Evaluating the demographic characteristics and political preferences of MTurk survey respondents. *Research & Politics*, 2(3): 2053168015604648.

Jablonski, S.; and Ludwig, M. 2023. Teaching and Learning of Geometry—A Literature Review on Current Developments in Theory and Practice. *Education Sciences*, 13(7).

Jiang, H.; Zhang, X.; Cao, X.; Breazeal, C.; Roy, D.; and Kabbara, J. 2024. PersonaLLM: Investigating the Ability of Large Language Models to Express Personality Traits. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, 3605–3627. Mexico City, Mexico: Association for Computational Linguistics.

Kambhampati, S. 2024. Can Large Language Models Reason and Plan? *Annals of the New York Academy of Sciences*, 1534(1): 15–18. ArXiv:2403.04121 [cs].

Kambhampati, S.; Valmeekam, K.; Guan, L.; Stechly, K.; Verma, M.; Bhambri, S.; Saldyt, L.; and Murthy, A. 2024. LLMs Can’t Plan, But Can Help Planning in LLM-Modulo Frameworks. ArXiv:2402.01817 [cs].

Kim, C. 2012. The role of affective and motivational factors in designing personalized learning environments. *Educational Technology Research and Development*, 60: 563–584.

Klein-Latucha, O.; and Hershkovitz, A. 2024. When Leaving is Persisting: Studying Patterns of Persistence in an Online Game-Based Learning Environment for Mathematics. *Journal of Learning Analytics*, 1–10.

Koedinger, K. R.; Matsuda, N.; MacLellan, C. J.; and McLaughlin, E. A. 2015. Methods for Evaluating Simulated Learners: Examples from SimStudent. In *AIED Workshops*.

Käser, T.; and Alexandron, G. 2023. Simulated Learners in Educational Technology: A Systematic Literature Review and a Turing-like Test. *International Journal of Artificial Intelligence in Education*.

Land, S.; and Jonassen, D. 2012. *Theoretical foundations of learning environments*. Routledge.

Loya, M.; Sinha, D.; and Futrell, R. 2023a. Exploring the sensitivity of LLMs’ decision-making capabilities: Insights from prompt variations and hyperparameters. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the association* *for computational linguistics: EMNLP 2023*, 3711–3716. Singapore: Association for Computational Linguistics.

Loya, M.; Sinha, D. A.; and Futrell, R. 2023b. Exploring the Sensitivity of LLMs' Decision-Making Capabilities: Insights from Prompt Variation and Hyperparameters. *arXiv preprint arXiv:2312.17476*.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 8086–8098. Dublin, Ireland: Association for Computational Linguistics.

Mannekote, A. 2024. Towards Compositionally Generalizable Semantic Parsing in Large Language Models: A Survey. *arXiv preprint arXiv:2404.13074*.

Matsuda, N.; Cohen, W. W.; and Koedinger, K. R. 2015. Teaching the teacher: tutoring SimStudent leads to more effective cognitive tutor authoring. *International Journal of Artificial Intelligence in Education*, 25: 1–34. Publisher: Springer.

Mohammadi, B. 2024. Wait, It's All Token Noise? Always Has Been: Interpreting LLM Behavior Using Shapley Value. ArXiv:2404.01332 [cs].

Ozeki, K.; Ando, R.; Morishita, T.; Abe, H.; Mineshima, K.; and Okada, M. 2024. Exploring Reasoning Biases in Large Language Models Through Syllogism: Insights from the NeuBAROCO Dataset. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 16063–16077. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. ArXiv:2304.03442 [cs].

Park, J. S.; Popowski, L.; Cai, C.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–18.

Rajarathinam, R. J.; Palaguachi, C.; and Kang, J. 2024. Enhancing Multimodal Learning Analytics: A Comparative Study of Facial Features Captured Using Traditional vs 360-Degree Cameras in Collaborative Learning. In PaaÃŸen, B.; and Epp, C. D., eds., *Proceedings of the 17th International Conference on Educational Data Mining*, 551–558. Atlanta, Georgia, USA: International Educational Data Mining Society. ISBN 978-1-7336736-5-5.

Renkl, A.; and Atkinson, R. K. 2007. Interactive learning environments: Contemporary issues and trends. An introduction to the special issue. *Educational Psychology Review*, 19: 235–238.

Rieber, L. P. 1996. Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational technology research and development*, 44(2): 43–58.

Sainz, O.; Campos, J.; García-Ferrero, I.; Etxaniz, J.; de Lacalle, O. L.; and Agirre, E. 2023. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 10776–10787. Singapore: Association for Computational Linguistics.

Sandoval, W. 2014. Conjecture mapping: An approach to systematic educational design research. *Journal of the learning sciences*, 23(1): 18–36.

Sclar, M.; Choi, Y.; Tsvetkov, Y.; and Suhr, A. 2023. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Tian, X.; Mannekote, A.; Solomon, C. E.; Song, Y.; Wise, C. F.; Mcklin, T.; Barrett, J.; Boyer, K. E.; and Israel, M. 2024. Examining LLM Prompting Strategies for Automatic Evaluation of Learner-Created Computational Artifacts. In PaaÃŸen, B.; and Epp, C. D., eds., *Proceedings of the 17th International Conference on Educational Data Mining*, 698–706. Atlanta, Georgia, USA: International Educational Data Mining Society. ISBN 978-1-7336736-5-5.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xiao, Y.; Cheng, Y.; Fu, J.; Wang, J.; Li, W.; and Liu, P. 2023. How far are we from believable AI agents? A framework for evaluating the believability of human behavior simulation. *arXiv preprint arXiv:2312.17115*.

Yu, D.; Kaur, S.; Gupta, A.; Brown-Cohen, J.; Goyal, A.; and Arora, S. 2023. Skill-Mix: A flexible and expandable family of evaluations for AI models. *arXiv preprint arXiv:2310.17567*.

Zhao, T. Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate Before Use: Improving Few-Shot Performance of Language Models. ArXiv:2102.09690 [cs].