BoxingGym: Benchmarking Progress in Automated Experimental Design and Model Discovery

Kanishk Gandhi* **Michael Y. Li** * **Lyle Goodyear** **Agam Bhatia**

Louise Li **Aditi Bhaskar** **Mohammed Zaman** **Noah D. Goodman**

Stanford University

Abstract

Understanding the world and explaining it with scientific theories is a central aspiration of artificial intelligence research. Proposing theories, designing experiments to test them, and then revising them based on data are key to scientific discovery. Despite the promise of LLM-based scientific agents, no benchmarks systematically test their ability to propose scientific models, collect experimental data, and revise them in light of new data. We introduce `BoxingGym` , a benchmark with 10 environments for evaluating experimental design (*e.g.,* collecting data to test a scientific theory) and model discovery (*e.g.,* proposing and revising scientific theories). To enable quantitative and principled evaluation, we implement each environment as a generative probabilistic model with which a scientific agent can run interactive experiments. These probabilistic models are drawn from various real-world scientific domains ranging from psychology to ecology. To evaluate a scientific agent's ability to collect informative experimental data, we compute the expected information gain (EIG), an information-theoretic quantity which measures how much an experiment reduces uncertainty about the parameters of a generative model. A good scientific theory is a concise and predictive explanation. To quantitatively evaluate model discovery, we ask a scientific agent to explain their model and evaluate whether this explanation helps another scientific agent make more accurate predictions. We evaluate several open and closed-source language models of varying sizes. We find that larger models (32B) consistently outperform smaller variants (7B), and that closed-source models generally achieve better results than open-source alternatives. However, all current approaches struggle with both experimental design and model discovery, highlighting these as promising directions for future research. [2]

"To understand a system, you must perturb it."
– George Box *(ad sensum)*

1 Introduction

Helping humans understand the world (and themselves) by discovering scientific theories is a foundational goal of artificial intelligence research [31]. Proposing theories about the world, conducting experiments to test them, and revising them based on data is central to this process [9]. Recent advances in large language models (LLMs), have shown promising potential for accelerating scientific discovery. LLMs have extensive scientific knowledge [2], strong inductive reasoning capabilities [53, 43], and the ability to propose models of data [27, 28, 11]. These promising results suggest that LLMs, functioning as autonomous agents, could be well-suited for experimental design (*i.e.,* col-

*Equal Contribution. Corresponding author: `kanishk.gandhi@stanford.edu`

[2]Project: https://github.com/kanishkg/boxing-gym

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Scaling Environments for Agents (SEA).

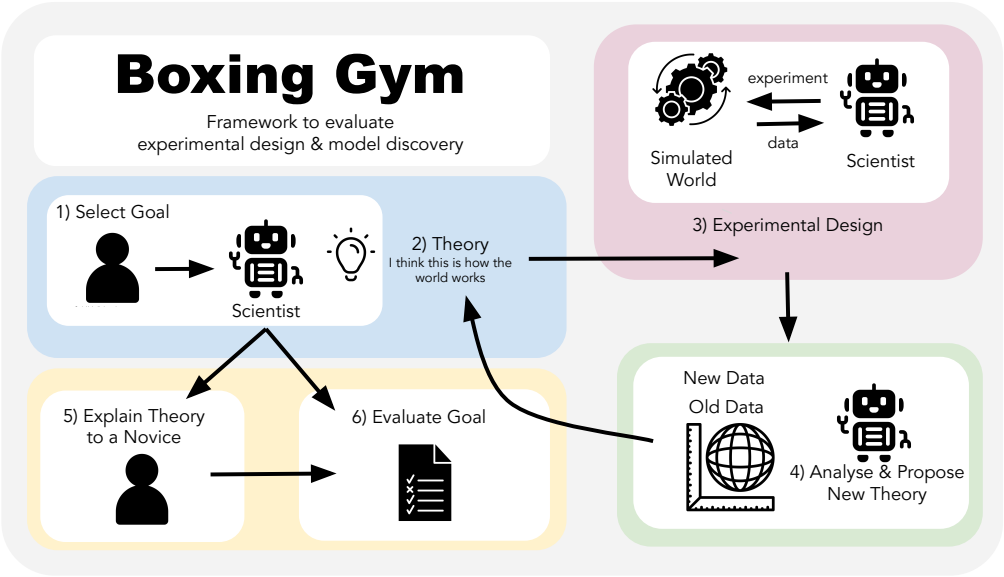

Figure 1: **Overview of** `BoxingGym`**.** The `BoxingGym` Framework is designed to holistically evaluate experimental design and model discovery capabilities in the spirit of George Box [9]. 1) The process starts with a user defining a goal for the scientist agent. 2) The scientist formulates a theory. 3) This theory guides the experimental design, where the scientist interacts with a simulated world to gather new data. 4) The scientist then analyzes the new and old data to propose and refine theories. This iterative process continues for several iterations. 5) The scientist is then asked to explain the findings to a novice. 6) We evaluate the novice and the scientist by casting the goal as a prediction problem.

lecting informative experiments to test scientific theories) and model discovery (*i.e.,* developing interpretable models based on experimental data).

Previous work has evaluated automated experimental design and model discovery in isolation [16, 17, 15, 27]. However, they are fundamentally coupled in real-world settings: scientists collect experimental data to build better models and better models inform better experiments. While scientific agents are promising, there is currently no systematic way to evaluate an agent's ability to propose scientific models, collect experimental data, and revise them in light of new data. This motivates the need for a benchmark that evaluates an agent's capabilities holistically in an integrated scientific discovery pipeline.

We outline the key desiderata for a framework that evaluates experimental design and model discovery: (1) The framework should enable the agent to *actively experiment* with the environment without requiring the agent to perform time-consuming and resource-intensive real-world lab experiments. (2) Since scientific theories come in different forms, the framework should flexibly accommodate *different representations of scientific theories*. (3) The framework should evaluate experimental design and model discovery in an *integrated* way. (4) Science is often *goal-directed* or driven by an inquiry. For example, a biologist might perform experiments with the goal of identifying cellular mechanisms underlying circadian rhythm in mammals. Our framework should allow users to specify high-level goals to guide the agent's discovery process. Our desiderata are inspired by the framework for scientific modeling introduced by George Box [7, 8], which emphasizes an iterative process of building models, designing experiments to test them, and revising them accordingly.

To achieve these desiderata, we introduce `BoxingGym` (Fig. 1) a flexible framework for evaluating experimental design and model discovery with autonomous agents. Our benchmark consists of 10 *environments* grounded in real-world scientific models. To enable agents to actively experiment, we implement each environment as a generative model. This key design choice makes simulating active experimentation tractable because it corresponds to sampling from the underlying generative model, conditioned on the experimental interventions. To accommodate various representations of scientific theories, all environments are designed with a flexible language based interface (Fig. 2). Finally, our environments can be instantiated with different goals, or intents for inquiry, that encourage the agent to adapt their experimentation towards accomplishing the goal (*e.g.,* understand the parameters underlying participant behavior in a psychology study) by specifying the goal in language.

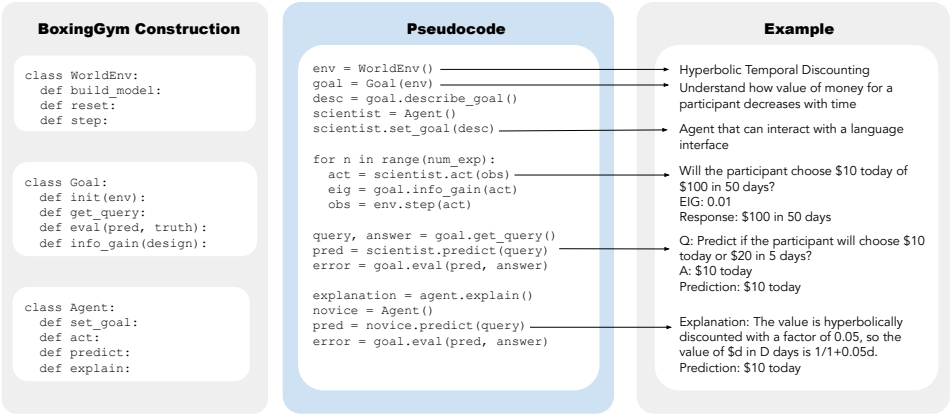

Figure 2: **Python pseudocode examples. (left)** `BoxingGym` is instantiated as modular classes and methods for the environment (WorldEnv), goals (Goal), and agents (Agent). **(center)** Pseudocode illustrating the workflow of setting goals, performing experiments, predicting outcomes, and providing explanations. **(right)** An example, hyperbolic temporal discounting, where the agent predicts a participant's choice between immediate and delayed rewards and explains the concept to a novice.

We introduce principled evaluation metrics that measure the quality of experiments and discovered models. To evaluate experimental design, we draw from *Bayesian optimal experimental* (BOED) design [44] and use *expected information gain* (EIG) to measure the informativeness of an experiment. EIG captures how much an experiment reduces uncertainty in the parameters of a generative model and, importantly, this measure complements our decision to implement environments as generative models. To evaluate model discovery, we take inspiration from the fact that science is a communicative endeavor. We propose a *communication-based* evaluation strategy: we ask a scientist agent to distill their experiments into a natural language explanation and evaluate how much that explanation empowers a novice agent, who does not have access to the experiments conducted by the scientist, to make accurate predictions about the environment.

We evaluate several open and closed-source language models ranging from 7B to 32B parameters. We find that larger models consistently outperform smaller variants, and closed-source models generally achieve better results than open-source alternatives. We also evaluate Box's Apprentice [27], which augments language models with statistical modeling capabilities, but find that this augmentation does not reliably improve performance. Notably, we observe substantial variation in difficulty across environments, which remaining challenging even for the strongest models. Promisingly, some environments show clear performance improvements with model scale. These results highlight significant opportunities for improving automated scientific reasoning.

2 Related Works

Optimal Experimental Design. Bayesian optimal experimental design (BOED) is a principled framework for designing maximally informative experiments across various disciplines [49, 12, 35]. While theoretically appealing, BOED's practical implementation is challenging due to the intractability of information gain metrics like expected information gain (EIG). Although several methods [44, 16, 17] exist to approximate EIG, they assume the data follows a fixed generative model—limiting their utility when model revision is needed as new data is collected.

Automated Model Discovery. Automated model discovery from data has been a long-standing goal in AI, aiming to build interpretable models that capture underlying patterns in data—from physical laws [6, 32] to nonparametric regression [15]. Recent work [27, 28] has integrated language models into this process, leveraging their ability to both propose and critique candidate models, demonstrating their potential as tools for automated model discovery. This work highlights the potential of using language models as a powerful tool for model discovery.

Reasoning and Exploration with LLMs. Language models have shown promising capabilities in both deductive reasoning (deriving consequences from hypotheses) [47, 46, 42] and inductive reasoning (inferring hypotheses from observations) [53, 43]. While reinforcement learning has improved LLMs' reasoning abilities [24, 21, 20, 22], these advances have primarily focused on deterministic, verfiable systems rather than the stochastic data typical in scientific discovery. Efficient exploration and information-seeking are crucial for experimental design and model building. Recent

work [37, 33, 19, 18, 48, 26] has investigated in-context exploration strategies and shown how language models can learn how to search and explore directly through sequence modeling, developing effective search strategies in language.

Interactive Environments. Drawing inspiration from established reinforcement learning principles [10, 34], `BoxingGym` adopts the modularity and simplicity of classic environments like OpenAI Gym while shifting focus to evaluation rather than agent training. While recent work has expanded interactive benchmarks to language agents —spanning tasks from software debugging [25] to automated scientific research[36, 29], our work advances this direction by introducing a principled framework for evaluating language agents' capabilities in iterative experimental design and model discovery.

3 Boxing Gym

3.1 Problem Formulation.

We formalize experimental design and model discovery using probabilistic modeling and Bayesian optimal experimental design (BOED). In `BoxingGym` , each environment is implemented as a generative model defining a joint distribution over the experimental outcome y, experimental design d, and unobserved parameters θ. This joint distribution is defined in terms of a prior distribution over θ, $p(\theta)$ and a *simulator* $p(y|\theta, d)$ which is a model of the experimental outcome y given parameters θ and design d. For example, in a psychology experiment, θ could be the parameters of a behavioral model of participants, d could be the questions posed to participants, and y could be the participant's response to d. Running an experiment corresponds to choosing a design d and observing a sample y from the marginal predictive distribution conditioned on that design, *i.e.,* $y \sim p(y|d) = E_{p(\theta)}[p(y|\theta, d)])$ [3].

3.2 Evaluation

3.2.1 Evaluating experimental design via Expected Information Gain

To evaluate experimental design, we take inspiration from the Bayesian OED literature [16, 17]. Crucially, our choice to implement environments as generative models enables us to leverage this literature. For each domain, we have an underlying predictive model $p(y|\theta, d)$. We quantify the *informativeness* of a design d through the expected information gain (EIG), that measures the reduction in posterior uncertainty about the model parameters θ after running an experiment d. Below, H is the Shannon entropy.

$$\text{EIG}(d) = \mathbb{E}_{p(y|d)}\left[H[p(\theta)] - H[p(\theta|y, d)]\right]$$

Since the EIG is typically not available in closed-form, we use a Nested Monte Carlo estimator

$$\hat{\mu}_{\text{NMC}}(d) = \frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{p(y_n|\theta_{n,0}, d)}{\frac{1}{M}\sum_{m=1}^{M} p(y_n|\theta_{n,m}, d)}\right) \quad \text{where} \quad \theta_{n,m} \overset{\text{i.i.d.}}{\sim} p(\theta),\ y_n \sim p(y|\theta = \theta_{n,0}, d)$$

We chose this estimator because it is a consistent estimator of the true EIG [44] and is straightforward to implement. EIG measures the value of an experiment under the assumption that the true distribution of experimental outcomes is modeled by $p(y|d)$. In general, this assumption is not true, but EIG is still a useful measure since we generate data from an underlying model in our benchmarks.

3.2.2 Evaluating model discovery via communication

To evaluate the quality of a model, we use standard model evaluation metrics (*e.g.,* prediction MSE) and a communication-based metric that takes advantage of the natural language interface. In particular, a *scientist agent* interacts with an environment through experiments. After these experiments, we ask the scientist agent to synthesize their findings through an *explanation*. We then evaluate how much that explanation enables a *novice* agent to make more accurate predictions about the environment without any additional experiments. Since a good explanation is both *predictive* and *parsimonious*, we set a token limit on the explanation. Crucially, this evaluation method can accommodate different forms of scientific theories. In our experiments, we ask the scientist agent to produce a statistical model and then distill the model into a natural language explanation to guide the novice agent.

[3] In the sequential setting, we replace the prior $p(\theta)$ with the posterior $p(\theta|y, d)$.

3.2.3 Evaluating goals via prediction

To evaluate success at achieving a specific goal (*e.g.,* how do the populations of predator and prey change with time) we employ a prediction target (*e.g.,* predict the population of predators at a particular time) and calculate a standardized prediction error. First, we compute the error between the predicted and true values. Then, we standardize this error with respect to the prior predictive mean, which is obtained by assuming a uniform prior over the design space. Specifically, for each domain, we sample a design d uniformly from the design space and a parameter θ from the prior distribution $p(\theta)$. We then generate samples from the predictive model $p(y|\theta, d)$ and average over multiple d and θ to obtain the prior predictive mean μ_0 and variance σ_0. Let $\{y_i\}_{i=1}^n$ be the ground truth outputs for inputs $\{x_i\}_{i=1}^n$. and let $\{\hat{y}_i\}_{i=1}^n$ be the predictions of the agent. The standardized prediction error is then calculated using these quantities, providing a measure of the agent's performance relative to the prior predictive mean. We use a domain-specific function f computing the discrepancy between a prediction $\hat{y}_i$ and ground truth value y_i (*e.g.,* MSE). We compute the errors $\epsilon_i = f(\hat{y}_i, y_i)$ and $\epsilon_{\mu_0} = f(\mu_0, y_i)$. Finally, we compute the standardized error as $\frac{\epsilon_i - \epsilon_{\mu_0}}{\sigma_0}$. Crucially, since this metric is computed with respect to the prior predictive, this metric can be negative.

3.3 Design Decisions in Constructing `BoxingGym`

We outline the key design decisions of `BoxingGym` that allow it to capture key aspects of scientific discovery within a flexible, simulated, and extensible environment.

Discovery via active experimentation. The agent actively interacts with the environment by conducting experiments, reflecting the real-world coupling of experimentation and model discovery. This approach assesses the agent's ability to gather relevant data and refine its models based on experimental results.

Real-world scientific models. Our environments are grounded in real-world scientific models from several domains, ensuring the benchmark tests the agent's ability to handle realistic scenarios. We implement these environment as `pymc` generative models to make active experimentation an automatic and tractable process.

Goal-driven discovery. Each environment has a specific goal, mirroring the inquiry-driven nature of scientific research. This encourages the agent to engage in targeted experimentation.

Language-based interface for experiments. We use a language-based interface for our experiments because it's flexible (*i.e.,* scientific domains can generally be described in language), easily integrates with LLMs, and interpretable to humans.

Emphasis on Measuring Discovery with Explanations. `BoxingGym` places a strong emphasis on measuring the quality of the agent's discoveries through the explanations it can provide after experimentation (§3.2.2). This design decision is motivated by two considerations. From a theoretical perspective, science is fundamentally about developing better theories, and scientific theories are explanations of observed phenomena. From a practical perspective, communicating findings to the broader scientific community is an essential aspect of scientific research. By using language, we do not have to commit to a particular representation of a scientific theory. We illustrate this flexibility, by showing how different representations can be easily integrated within our method for measuring natural language explanations.

Extensible/modular environments for benchmarking agents. `BoxingGym` is easily extensible and modular, enabling researchers to integrate new environments and test different agents with minimal effort. We illustrate this in Fig. 2 which provides a pseudo-code example of how to implement a new environment and goal in `BoxingGym` .

3.4 Domains

`BoxingGym` consists of 10 environments (see App. D for full details) that cover a range of scientific domains and test different aspects of experimental design and model discovery. Some environments are designed to test optimal experiment design, while others focus on model discovery or involve simulated neuro-symbolic human participants.

Location finding. [17] In an n-dimensional space with k signal-emitting sources, the scientist measure signals at any grid location. Goals include predicting the signal at any point or locating the sources.

Hyperbolic temporal discounting. [17] The scientist observes a participant's choices for different immediate rewards (ir), delayed rewards (dr), and delay periods (D days) Fig. 2 (right). Goals include predicting choices of a participant or discount factors.

Death process. [17] A disease spreads at an infection rate. The scientist can measure the number of infected individuals at different points of time to predict future infections or the infection rate.

Item Response Theory (IRT). [45] In this environment, there is a set of students and a set of questions. The experimenter can observe the correctness of a student's response to a particular question. The goal is to discover the underlying model that relates student ability and question difficulty to the probability of a correct response.

Animal growth curves. [30] An experimenter can observe the length of a dugong at a particular age. The goal is to discover the underlying growth model of dugongs.

Population growth dynamics. [30] An experimenter can observe the population of peregrines at a particular point in time. The goal is to discover the underlying population dynamics model. This is tested by asking the experimenter to predict population dynamics at a particular point in time.

Mastectomy Survival analysis. [13] The experimenter can observe if a patient is alive after a mastectomy, including metastasis status and time since surgery. The goal is to predict survival probabilities for new patients.

Predator-Prey dynamics. [52] This simulates predator-prey populations over time. The goal is to discover models like the Lotka-Volterra equations to predict future populations.

Emotion from outcome. [38] Participants guess a player's emotions after a gambling game's outcome. The experimenter designs games with varied probabilities and prizes to model how participants judge the emotions of a player from outcomes. Human participants are simulated using a probabilistic model translated into natural language by a language model.

Moral Machines. [5] Participants face moral dilemmas, choosing which group an autonomous car should save. Experimenters manipulate group compositions and required actions to model moral decision-making. Human participants are simulated with a probabilistic model, and their actions are translated into natural language by a language model.

4 Experiments

We conduct experiments to evaluate the performance of two baseline agents on `BoxingGym` . Our goal is to assess their ability to perform experimental design and theory building across a diverse set of environments. We benchmark two types of agents: a standard language model (GPT-4o, OpenAI [39]) and a language model augmented with symbolic reasoning capabilities (Box's Apprentice).

LLM Agent. We consider 6 LLMs, GPT-4o [39], Claude-3.7-sonnet [3], Qwen-2.5-32b-instruct, Qwen-2.5-7b-instruct [55], and reasoning variants OpenThinker-32b, and OpenThinker-7b [51]; the reasoning variants are finetuned on math and coding task. We prompt these models to interact with our environment, purely through natural language, without additional tools (see Fig. 2, see App. B for details).

Box's Apprentice. The apprentice agent augments language models by enabling them to implement generative models of observed data. For model discovery, the agent writes a `pymc` program [27] after 10 experiments, which is then fit and provided to the scientist explaining findings to the novice. For experimental design, the agent creates and uses these models to guide subsequent experiments.

Experiment Setup. For each environment, we run the agents for 5 independent trials. At each step, the agent chooses to perform an experiment, by specifying a design, and observes the outcome.

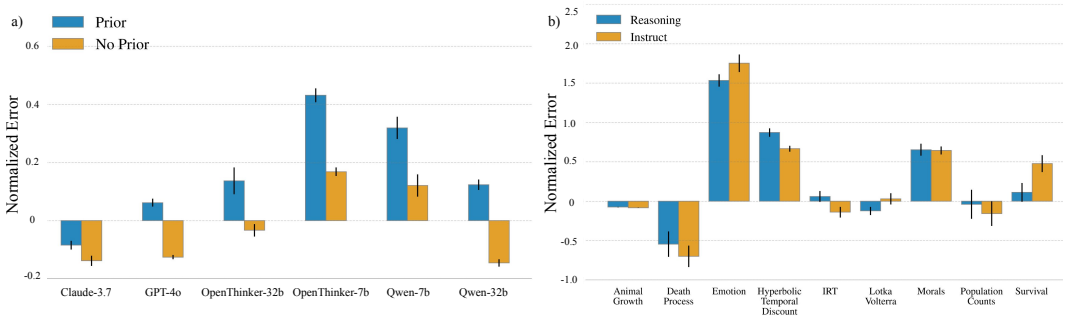

Figure 3: **Normalized Error Compared across Models.** (a) Comparison of the normalized errors for different LLMs with or without prior information included in the prompt. (b) Comparison of reasoning models (OpenThinker) and instruct models (Qwen) across environments. Error bars are the standard error across 5 runs.

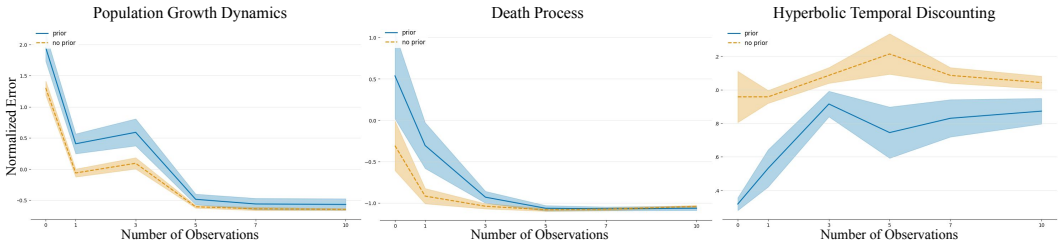

Figure 4: **Normalized Errors Over Number of Observations.** Normalized errors for the LLM agent with `gpt-4o` with prior information (solid blue) and without prior information (dotted yellow) across three domains: Population Growth Dynamics (left), IRT (center) and Hyperbolic Discounting (right). Error bars are the standard error across 5 runs.

After a fixed number of steps (0, 1, 3, 5, 7, 10), we evaluate the agent's performance using the metrics described earlier §3.2. The performance of models is averaged across 5 runs and over 10 evaluation points. We also explore a *prior* vs *no prior* condition to investigate whether domain knowledge helps or hinders scientific discovery. In the prior condition, we give the LM full context about the problem domain (*e.g.,* "you are observing how participants balance delayed vs immediate rewards"), simulating scientists with background knowledge. In the no prior condition, we remove this context and describe the setting in a domain-agnostic way (*e.g.,* "you receive a tuple of three values"), resembling reasoning from raw observations without preconceptions. This tests whether prior knowledge scaffolds discovery or creates biases that constrain exploration.

4.1 Experimental Design Evaluation

Setup. To evaluate the agents' performance, we first assess their ability to gather valuable information through their experiment selection and then measure how effectively they use this information to predict the environment. The Expected Information Regret (EI Regret) compares the Expected Information Gain (EIG) (§3.2.1) of the agent's chosen experiments to the maximum EIG achievable from 100 random experiments. Lower EI Regret indicates more informative experiment selection.

Prior information does not improve performance. We find that models often perform better when given no prior information after 10 experiments (Fig. 3a). In some cases, this is because the LLM makes an overly strong assumption about the environment (*e.g.,* the signal decay is symmetric around the origin) and does not revise the assumption after more experiments; this is consistent with findings reported by Li et al. [27]. In other cases, such as the hyperbolic discounting environment (Fig. 4, right), the model overfits to limited observations.

More experiments generally lead to better predictions. We plot the learning trajectories for three environments in (Fig. 4). The agent's average prediction error decreases as it performs more

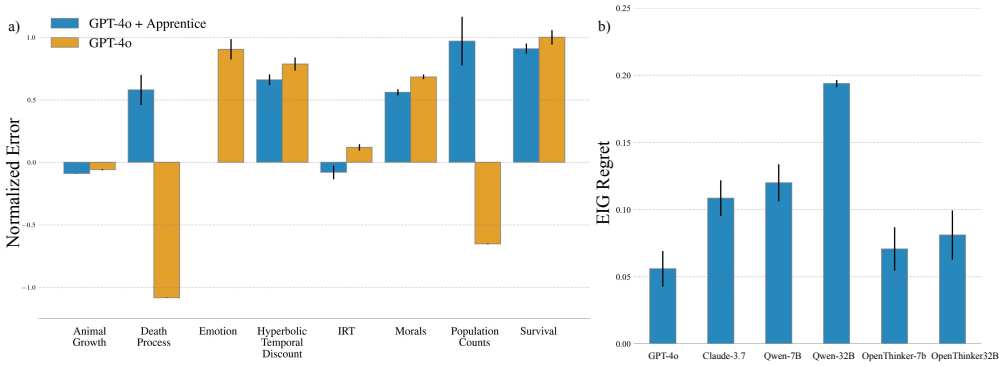

Figure 5: (a) Comparison of the Box's Apprentice with an LLM agent. (b) EIG Regret scores for six large language models, with lower values indicating better performance.

experiments. The Hyperbolic Temporal Discounting environments shows an unexpected trends where more experiments actually increases error. This may again be related to how prior knowledge interferes with effective learning from data.

Models Improve with Scale. Larger models consistently outperform their smaller counterparts within the same model family. Both OpenThinker-32B and Qwen2.5-32B demonstrate significantly better performance than their respective 7B variants across environments (Fig. 3a), highlighting the benefits of scale for experimental design tasks.

Instruction-Tuned Models outperform Reasoning Models. Surprisingly, the instruction-tuned Qwen2.5 models outperform the reasoning-focused OpenThinker models (Fig. 3b). This may be because OpenThinker models are finetuned to perform well on a relatively narrow set of verifiable problems in math and code, while instruction-tuned models retain broader capabilities that could be useful for experimental design.

Models performance varies substantially across environments. Models show varying performance across different environments (Fig. 3b). Performance is strongest on environments like population growth dynamics and death process, where the LM agent achieves negative standardized error, indicating that the LM successfully leveraged information gained through experimentation. However, in environments like hyperbolic discounting, performance is low even after experimentation, suggesting that some domains are inherently more challenging for current models.

EIG Regret reveals relationship between experimental design and prediction. Our EIG regret analysis (Fig. 5b) provides insight into the relationship between two key components of scientific reasoning: designing informative experiments and making accurate predictions from collected data. GPT-4o achieves both the lowest EIG regret and strong predictive performance across several environments, suggesting these capabilities can be aligned. However, the varying performance of other models is informative — for instance, Qwen-32B shows higher EIG regret despite good predictive performance in some domains, indicating that while these abilities may be related, excellence in prediction doesn't automatically translate to optimal experimental design.

LLMs cannot always optimally leverage statistical models. While Box's Apprentice can propose and fit explicit statistical models to observed data, it does not consistently improve over the non-augmented LLM (GPT-4o) (Fig. 5a) From qualitative analysis of the models, we find that Box's Apprentice tends to favor overly simple functional forms due to limited data, such as using linear approximations for inherently nonlinear phenomena.

4.2 Evaluating Model Discovery via Communication

Setup. Next, we evaluate the agents' ability to build and communicate models that capture the underlying phenomena in each environment. To test this, we have the agents interact with the environment for 10 steps (scientist phase) and then generate a natural language explanation of their

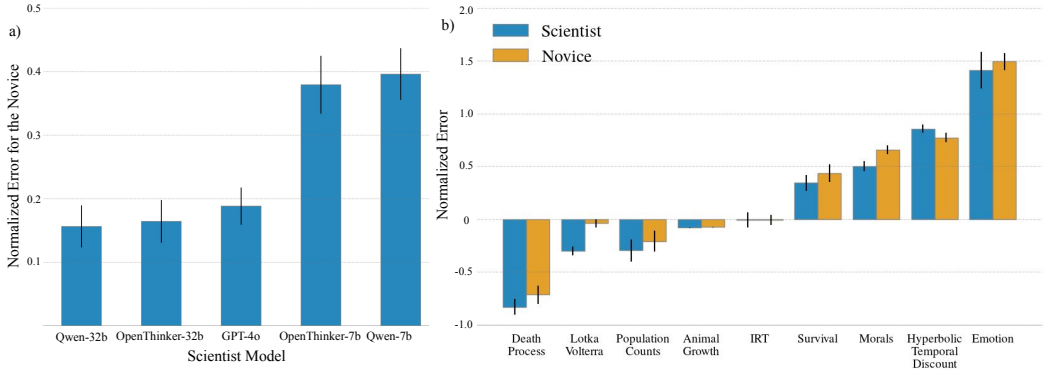

Figure 6: **Evaluation of Model Discovery via Communication.** (a) Comparison of the standardized error of the Novice (`gpt-4o`) with different Scientist models. (b) Comparison of errors made by the Novice and the Scientist (both models are `gpt-4o`). Error bars are standard error.

findings. We then provide this explanation to a *novice* agent, which must make predictions about the environment without any direct interaction (novice phase by using the explanation from the scientist; §3.2.2). The novice agent is always `gpt-4o`. The scientist's prediction after 10 observations (Error After Experiments) acts as a weak positive control. Ideally, if the scientist's explanation is effective, the novice's error should approach the positive control.

Explanations improve with scale. Larger models generally produce more effective explanations, as evidenced by better novice performance when using explanations from 32B variants compared to 7B models (Fig. 6a). This suggests that increased model scale improves not just experimentation but also the ability to distill and communicate findings.

Explanations are not as good as experiments. As expected, novice agents perform worse than scientists who directly interacted with the environment (Fig. 6b). The gap suggests that current explanation methods do not fully capture the knowledge gained through experimentation.

Explanations are more helpful for some environments. However, the effectiveness of explanations varies substantially across domains (Fig. 6b). For instance, explanations are helpful for animal growth, but struggle with complex domains like moral judgments. This variation likely reflects the complexity of different domains and the current limitations of language models in capturing and communicating certain types of patterns.

5 Discussion

We introduced `BoxingGym` , a benchmark measuring language-based agents' capabilities in experimental design and model discovery across 10 real-world-based environments. We evaluated experimental design using information gain metrics and developed a novel model discovery metric based on an agent's ability to explain its model to a novice agent. Our evaluation across multiple model scales (7B-32B parameters) shows that while larger and closed-source models generally perform better, fundamental challenges persist. Neither domain-specific prior knowledge nor statistical modeling capabilities consistently improved performance. Some environments yielded strong results with larger models, while others remained challenging for all approaches. `BoxingGym` has limitations: it uses pre-defined experimental paradigms rather than requiring design from scratch [14], ignores resource constraints, and covers limited scientific domains. Future work should address these limitations by incorporating experiment design from scratch, resource constraints, and more diverse fields [23]. We could also expand the human behavior environments (Moral Machines, Emotions) with more sophisticated participant simulations [4, 1, 50, 40, 41]. While our experiments demonstrated potential for interfaces that augment language models' scientific reasoning capabilities, future research should explore data visualization, model validation [28], and web-based research strategies to enhance experimental guidance and discovery.

References

[1] Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pages 337–371. PMLR, 2023.

[2] Microsoft Research AI4Science and Microsoft Azure Quantum. The impact of large language models on scientific discovery: a preliminary study using gpt-4, 2023.

[3] Anthropic. Claude 3.7 sonnet. https://www.anthropic.com, 2024.

[4] Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023.

[5] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. The moral machine experiment. *Nature*, 563(7729): 59–64, 2018.

[6] Josh C. Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104:9943 – 9948, 2007.

[7] G. E. P. Box and William G. Hunter. A Useful Method for Model-Building. *Technometrics*, 4: 301–318, 1962.

[8] George E. P. Box. Sampling and Bayes' Inference in Scientific Modelling and Robustness. *Journal of the Royal Statistical Society. Series A (General)*, 143(4):383–430, 1980. ISSN 00359238.

[9] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71 (356):791–799, 1976.

[10] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[11] Pablo Samuel Castro, Nenad Tomasev, Ankit Anand, Navodita Sharma, Rishika Mohanta, Aparna Dev, Kuba Perlin, Siddhant Jain, Kyle Levin, Noémi Éltető, Will Dabney, Alexander Novikov, Glenn C Turner, Maria K Eckstein, Nathaniel D Daw, Kevin J Miller, and Kimberly L Stachenfeld. Discovering symbolic cognitive models from human and animal behavior. *bioRxiv*, 2025. doi: 10.1101/2025.02.05.636732.

[12] Kathryn Chaloner and Isabella Verdinelli. Bayesian Experimental Design: A Review. *Statistical Science*, 10(3):273 – 304, 1995. doi: 10.1214/ss/1177009939. URL https://doi.org/10.1214/ss/1177009939.

[13] David Roxbee Cox. *Analysis of survival data*. Chapman and Hall/CRC, 2018.

[14] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[15] David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1166–1174, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[16] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[17] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021.

[18] Kanishk Gandhi, Dorsa Sadigh, and Noah D Goodman. Strategic reasoning with language models. *arXiv preprint arXiv:2305.19165*, 2023.

[19] Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

[20] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[21] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[22] Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*, 2024.

[23] Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications, 2025.

[24] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[25] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

[26] Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul Mcvay, Michael Rabbat, and Yuandong Tian. Beyond a*: Better planning with transformers via search dynamics bootstrapping. *arXiv preprint arXiv:2402.14083*, 2024.

[27] Michael Y Li, Emily B Fox, and Noah D Goodman. Automated Statistical Model Discovery with Language Models. In *International Conference on Machine Learning (ICML)*, 2024.

[28] Michael Y. Li, Vivek Vajipey, Noah D. Goodman, and Emily B. Fox. Critical: Critic automation with language models, 2024. URL https://arxiv.org/abs/2411.06590.

[29] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[30] Måns Magnusson, Paul Bürkner, and Aki Vehtari. posteriordb: a set of posteriors for Bayesian inference and probabilistic programming, October 2023.

[31] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12–12, 1955.

[32] B. A. McKinney, J. E. Crowe, H. U. Voss, P. S. Crooke, N. Barney, and J. H. Moore. Hybrid grammar-based approach to nonlinear dynamical system identification from biological time series. *Phys. Rev. E*, 73:021912, Feb 2006. doi: 10.1103/PhysRevE.73.021912.

[33] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

[34] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[35] Jay I. Myung, Daniel R. Cavagnaro, and Mark A. Pitt. A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, 57(3):53–67, 2013. ISSN 0022-2496. doi: https://doi.org/10.1016/j.jmp.2013.05.005. URL https://www.sciencedirect.com/science/article/pii/S0022249613000503.

[36] Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, et al. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*, 2025.

[37] Allen Nie, Yi Su, Bo Chang, Jonathan N Lee, Ed H Chi, Quoc V Le, and Minmin Chen. Evolve: Evaluating and optimizing llms for exploration. *arXiv preprint arXiv:2410.06238*, 2024.

[38] Desmond C Ong, Jamil Zaki, and Noah D Goodman. Affective cognition: Exploring lay theories of emotion. *Cognition*, 143:141–162, 2015.

[39] OpenAI. Hello, GPT-4. https://openai.com/index/hello-gpt-4o/, 2024. Accessed: 2024-06-04.

[40] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18, 2022.

[41] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[42] Gabriel Poesia, Kanishk Gandhi, Eric Zelikman, and Noah D Goodman. Certified deductive reasoning with language models. *arXiv preprint arXiv:2306.04031*, 2023.

[43] Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*, 2024.

[44] Tom Rainforth, Robert Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On Nesting Monte Carlo Estimators. *International Conference on Machine Learning (ICML)*, 2018.

[45] Georg Rasch. *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993.

[46] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2022.

[47] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36, 2024.

[48] John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada Lewis, Anian Ruoss, et al. Mastering board games by external and internal planning with language models. *arXiv preprint arXiv:2412.12119*, 2024.

[49] Ben Shababo, Brooks Paige, Ari Pakman, and Liam Paninski. Bayesian inference and online experimental design for mapping neural microcircuits. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/17c276c8e723eb46aef576537e9d56d0-Paper.pdf.

[50] Omar Shaikh, Valentino Chai, Michele J Gelfand, Diyi Yang, and Michael S Bernstein. Rehearsal: Simulating conflict to teach conflict resolution. *arXiv preprint arXiv:2309.12309*, 2023.

[51] Open Thoughts Team. Open Thoughts, January 2025.

[52] Vito Volterra. Variations and fluctuations of the number of individuals in animal species living together. *ICES Journal of Marine Science*, 3(1):3–51, 1928.

[53] Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. Hypothesis search: Inductive reasoning with language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[54] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[55] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe the design of our benchmark accurately, summarize results with different models.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See discussion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No proofs or new theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, further, all our code, results and scripts are available on github.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the code is accessible on the github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe this in detail in experimental setup and have the full specification in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report statistical significance in all our results...

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See appendix section B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Single blind submission and we follow the code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: We don't discuss these as there are no direct negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not relevant for the paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models have been cited appropriately. The papers that inspired the environments have been credited too.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We add documentation to the `BoxingGym` code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human participants were recruited.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Paper does not use human participants.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: None of the core methods used LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Env	Goal	Error@0	Error@10	Discovery@10
Hyperbolic Discounting	Choice	0.32±0.04 0.96±0.15	0.87±0.08 1.04±0.04	0.79±0.37 0.96±0.07
Hyperbolic Discounting	Discount	-0.06±0.00 -	-0.06±0.00 -	- -
Location Finding	Signal	0.30±0.25 0.63±0.39	0.59±0.55 0.86±0.47	4.75±4.51 1.52±1.28
Location Finding	Source Location	1.29±1.3 -	-0.15 ±0.4 -	- -
Death Process	Num Infected	0.54±0.52 -0.31±0.30	-1.06±0.03 -1.04±0.01	-1.08±0.01 -1.00±0.11
Death Process	Infection Rate	0.13±0.37 -	1.64±1.12 -	- -
IRT	Correctness	0.12±0.07 0.08±0.15	-0.24±0.10 0.00±0.13	0.12±0.18 0.12±0.14
Dugongs	Length	-0.04±0.02 -0.04±0.02	-0.08±0.00 -0.08±0.00	-0.06±0.04 -0.07±0.02
Peregrines	Population	1.95±0.22 1.30±0.11	-0.57±0.09 -0.65±0.01	-0.65±0.02 -0.66±0.03
Mastectomy	Survival	0.04±0.14 0.32±0.08	0.36±0.10 0.27±0.12	1.00±0.41 0.45±0.18
Predator-Prey	Population	0.38±0.04 0.75±0.02	-0.31±0.05 -0.42±0.01	-0.01±0.12 -0.07±0.40
Emotions	Prediction	1.04±0.21 N/A	1.22±0.29 N/A	0.90±0.58 N/A
Moral Machines	Judgement	0.40±0.07 N/A	0.36±0.04 N/A	0.68±0.13 N/A

Table 1: **Performance of GPT-4o Across Different Tasks.** Numbers shown are normalized-0 errors. Errors with prior (top line) and without prior (bottom line) appear on different lines. Errors are averaged across 5 runs.

A Full Results

See Tab. 1, and Tab. 2 for [4] prediction errors across all environments for GPT-4o and the Box's apprentice with GPT-4o. Full results are available in the Github Repository.

B LLM Agent

The LLM agent provides an easy way for a large language model (LLM) to interact with `BoxingGym` . By tailoring the system message to the specific environment, we can clearly define goals for the LLM, elicit experimental designs from it, make accurate predictions for queries, and generate explanations for a novice. This agent class also incorporates a simple retry mechanism that allows the LLM to correct its designs if they are initially invalid.

Models were configured with a temperature parameter of 0.0 to ensure deterministic outputs. Maximum token limits were set to 512 tokens for instruct models and 1024 tokens for thinking variants, providing sufficient thinking tokens for generating an answer without multiple retries.

[4]We omit the predatory-prey and Emotions domains for Box's Apprentice, since GPT-4o could not reliably produce `pymc` programs

Env	Goal	Error@0	Error@10	Discovery@10
Hyperbolic Discounting	Choice	0.66 ± 0.25 0.66 ± 0.25	1.17 ± 0.14 0.91 ± 0.09	0.66 ± 0.30 0.74 ± 0.42
Location Finding	Signal	0.99 ± 0.58 1.18 ± 0.64	1.45 ± 1.60 0.83 ± 0.60	1.18 ± 1.12 -0.01 ± 0.30
Death Process	Num Infected	3.79 ± 1.68 -0.90 ± 0.05	-1.02 ± 0.05 -0.61 ± 0.30	0.58 ± 0.85 0.50 ± 1.26
IRT	Correctness	0.44 ± 0.36 0.12 ± 0.24	-0.12 ± 0.14 0.12 ± 0.14	-0.08 ± 0.39 0.2 ± 0.40
Dugongs	Length	0.26 ± 0.12 0.05 ± 0.10	-0.08 ± 0.02 -0.09 ± 0.004	-0.09 ± 0.005 -0.08 ± 0.004
Peregrines	Population	2.71 ± 0.60 1.62 ± 0.47	0.04 ± 0.21 0.95 ± 0.86	0.97 ± 1.38 -0.19 ± 0.79
Mastectomy	Survival	0.14 ± 0.41 0.73 ± 0.15	0.55 ± 0.24 0.64 ± 0.15	0.91 ± 0.28 0.27 ± 0.23
Moral Machines	Judgement	0.97 ± 0.33	0.89 ± 0.21	0.56 ± 0.18

Table 2: **Performance of Box's Apprentice Across Different Tasks.** Standardized errors shown here. Errors with prior (top line) and without prior (bottom line) appear on different lines. Errors are averaged across 5 runs.

GPT-4o and Claude-3.7-Sonnet were accessed via their APIs, while all other models were deployed using vLLM. For the vLLM-served models, we utilized a dual A40 GPU configuration: one GPU dedicated to model serving and the other for inference execution through the vLLM endpoint. This architecture ensured optimal resource allocation and performance stability throughout the experimental process.

Each OED experimental run consisted of 10 predictions conducted after 0, 1, 3, 5, 7, and 10 observations, respectively. Comprehensive log files were generated for each set of predictions to facilitate subsequent analysis. Execution time varied across model architectures, with most configurations requiring approximately 2-3 minutes per run (defined as a single seed, configuration, and environment combination). Models accessed through external APIs typically required longer execution times due to network latency and rate limiting considerations. Discovery experiments reduced execution times compared to OED experiments due to the decreased number of required API calls.

C Box's Apprentice

We closely follow Li et al. [27]. In particular, to generate a candidate, we sample a single probabilistic program z from the proposal LM, $q_{\text{LM}}(\cdot)$. For the model discovery experiments, we perform this once after 10 experiments. For the OED experiments, we perform this three times over the course of 10 experiments. In all experiments, we use `GPT-4o` (`gpt-4o-2024-05-13`). The proposal LM q_{LM} "conditions" on $h^t \in \Sigma^*$, a natural language instruction synthesizing previous modeling approaches and suggesting new approaches, the previous program z^{t-1}, and a textual representation of the dataset $\mathcal{D}$.

$$z^t \sim q_{\text{LM}}(\cdot | z^{t-1}, h^{t-1}, \mathcal{D}).$$

We run this at a temperature of 0.0. Chain-of-thought reasoning, or generating intermediate reasoning steps, improves the performance of LMs [54]. Motivated by this, we instruct q_{LM} to reflect on the properties of the dataset, sketch a high-level modeling approach, state the hypotheses that it will address before writing a program, and add comments to code. See the system prompt in Figure 7.

D Domains

D.1 Location Finding

The location finding environment has hidden signal sources that emit a signal. The scientist can makeg measurements of the superimposed signal at various points. The experiment is directly taken from Foster et al. [16]. In table 3, we describe the inputs and outputs of the experiment.

Parameter	Description
Model	Superposition of K signal sources in d-dim space
Setup Parameters	Num signal sources K, dim of space d, base signal b, max signal m, noise σ
Observations	Total noisy signal at point of measurement
Goals	Predicting signal intensity at new points and source locations

Table 3: Location Finding

We define $k = 3$ signal sources in $\mathbb{R}^d = \mathbb{R}^2$ space with locations at θ_k. The number of sources is predefined and is known to the agent. Each source emits a signal strength α_k. In our implementation, we choose α_k to be fixed for all sources. The signal strength decays according to the inverse square law–if an agent measures at point ξ, then the noisy superimposed signal observed will be distributed according to $\mathcal{N}(\mu(\theta, \xi), \sigma)$ where σ is the signal noise, $\mu(\theta, \xi)$ is the total intensity at point ξ,

$$\mu(\theta, \xi) = b + \sum_{k=1}^{K} \frac{\alpha_k}{m + || \theta_k - \xi ||^2} \tag{1}$$

and $b, m > 0$ are constants governing background and maximum signal. Note that unlike Foster et al. [17], we observe the total intensity, not the log total intensity.

D.2 Hyperbolic Discounting

The hyperbolic discounting domain has two hidden variables (k, α) to describe a participant's behavior, where each participant is asked to choose between an immediate reward $\$iR$ or a delayed reward $\$dR$ in D days. The experiment is outlined in table 4 below.

Parameter	Description
Model	Human decision-making in temporal discounting of rewards
Setup Parameters	Params of the discount function (ϵ, mean and std for $\log k$, scale for α)
Observations	Choice between immediate iR and delayed reward dR at delay D
Goals	Predicting choices and the value of the discount factor

Table 4: Hyperbolic Discounting

In each measurement, we require iR is strictly smaller than dR and all three values have to be positive, because we assume a rational participant would always choose a higher immediate reward over a lower delayed reward. We follow the prior distribution of the latent variables given by Foster et al. [16]:

$$\log k \sim N(-4.25, 1.5), \alpha \sim HalfNormal(0, 2) \tag{2}$$

where the HalfNormal distribution is a normal distribution truncated at 0. For each test, there are three variables in design: iR, dR, and D. We give values to each choice: receiving the immediate reward $\$iR$ has value $V_i = iR$, while receiving the delayed reward $\$dR$ in D days has value $V_d = \frac{dR}{1+kD}$. Then, whether each participant's chooses the delayed reward in each scenario is characterized as a Bernoulli random variable $X \sim Bernoulli(p)$ where the probability of choosing the delayed reward is given by

Box's Apprentice system prompt

```
You are a brilliant statistician modeling a dataset.
Your job is to come up with a generative model that explains the true data by writing a pymc probabilistic program.
Here is a description of the dataset {dataset_description}
{dataset_text_representation}
Here is a description of the columns {column_description}
If you are in the first round, you will not receive any additional information.
However, for the second round and beyond, I will give you the model you proposed previously.
Please import pymc NOT pymc3!
Note that there are differences in the arguments pymc expects.
IMPORTANT: do not use sd as an argument use sigma instead!
It is crucial that you pass the idata_kwargs argument to pm.sample!!
IMPORTANT: Use the variable name "y_obs" for the observations when you define it!
IMPORTANT: Use the variable name "y_obs" for the observations when you define it!
IMPORTANT: Index the appropriate column names when grabbing data from observed_data. These column names are indicated in the column description.

Your answer should follow the template in the following order.
1. First, sketch a high-level probabilistic program for the data.
  You will go through multiple rounds of revision.
  If there's a previous program in your context window and a list of hypotheses, revise based on this information!
  Explicitly cite the hypotheses (if there are any) that you address in your sketch.
2. After coming up with a plan, write your program and add comments to lines of code that address certain hypotheses.
```python
  import pymc as pm
  import numpy as np
  def gen_model(observed_data):
      # convert observed_data columns to numpy arrays
      # index the appropriate column names

      ....
      rng1 = np.random.default_rng(42)
      rng2 = np.random.default_rng(314)
      with pm.Model as model():
          # create a pm.MutableData object for each non-observation column
          ...Your code here...
          # Copy the rest of this code verbatim but remember to have this indented in scope of model()!
          trace = pm.sample(1000, tune=500, target_accept=0.90, chains=3, cores=1, random_seed=rng1)
          posterior_predictive = pm.sample_posterior_predictive(trace, random_seed=rng2, return_inferencedata=False)
          return model, posterior_predictive, trace
```
```

Figure 7: BoxLM **system prompt** The system prompt for the proposal p_{LM}. We also include some additional instructions on pymc syntax such as wrapping features in a MutableData container.

$$p(X = 1|k, \alpha, iR, dR, D) = \epsilon + (1 - 2\epsilon)\Phi(\frac{V_d - V_i}{\alpha}) \tag{3}$$

where Φ is the cumulative distribution function of the standard normal distribution. In our implementation, we set $\epsilon = 0.01$ for all scenarios.

D.3 Death Process

The death process environment models an infection spreading among a healthy population of N individuals. The infection rate θ determines how the probability of infection increases over time. The environment is outlined in table 5 below.

Parameter	Description
Model	The spread of an infection over time
Setup Parameters	Pop size N, params of the infetion rate (μ, σ, upper and lower bounds)
Observations	Number of infected individuals at observation time
Goals	Predicting the number of infected individuals at a time and the infection rate

Table 5: Death Process

In our model, θ is given by the prior distribution outlined in Foster et al. [17].

$$\theta \sim \text{TruncatedNormal}(\mu = 1, \sigma = 1, min = 0, max = \infty) \tag{4}$$

The number of infected individuals Y at time t is distributed as a binomial random variable:

$$Y|\theta, t \sim \text{Binomial}(N, \eta) \tag{5}$$

where $\eta = 1 - e^{-\theta t}$, and N is the population size. We ask the agent to make observations sequentially by giving a time $t > 0$ at each step.

D.4 IRT

1PL IRT Model The one parameter IRT (or Rasch) domain models the performance of multiple students on multi-question exams. The binary outcome (whether the student is correct) of a student-question pair is determined by latent variables governing the student's proficiency and the question's difficulty (Figure 2). The agent's goal is to predict the outcome of a particular student-question pair. The agent may observe other student-question pairs to view their outcome. Table 6 below details the inputs, outputs, and target for every variation of the IRT model.

Param	Description
Model	Student performance on multi-question exams
Setup Parameters	Number of students N, number of questions Q, student-question pair to predict
Observations	Outcomes of various student-question pairs
Goals	Predicting the correctness of student responses to questions

Table 6: IRT Model

We define the ability α_j of student j and the difficulty β_k of question k. In our implementation, α and β are standard normals. The outcome O_{jk} of a student j on question k is determined by a Bernoulli trial where the probability of success p_{jk} is determined by the logit function of $z_{jk} = \alpha_j - \beta_k$.

$$p_{jk} = \frac{1}{1 + e^{-z_{jk}}} \tag{6}$$

In summary, for a given student-question pair, we compute the probability of the student getting the question correct and return the result of the corresponding Bernoulli trial.

2PL IRT Model The two parameter IRT model is identical to the 1PL variant with an additional variable governing the discriminability γ_k of question k. The discriminability models how sensitive the question is to incorrect answers. For higher values of γ, the probability of a student's answer being correct is higher. Thus the outcome O_{jk} of a student j on question k is determined by a Bernoulli trial where the probability of success p_{jk} is determined by the logit function of $z_{jk} = \gamma_k(\alpha_j - \beta_k)$.

3PL IRT Model The three parameter IRT model is identical to the 2PL variant with an additional variable modeling how susceptible a question is to guessing. For question k, c_k determines the probability that a student gets the question right by guessing. Thus the outcome O_{jk} of a student j on question k is determined by a Bernoulli trial where the probability of success p_{jk} is determined by

$$p_{jk} = c_k + (1 - c_k)\frac{1}{1 + e^{-z_{jk}}} \tag{7}$$

where $z_{jk} = \gamma_k(\alpha_j - \beta_k)$ as in 2PL.

We use the 2PL model in `BoxingGym` .

D.5 Dugongs

The dugongs environment has the ages and lengths of dugongs (sea cows)[30]. The goal is to model the length of a dugong based on its age. The following table describes the inputs and outputs of the experiment:

Parameter	Description
Model	Bayesian hierarchical model
Setup Parameters	alpha, beta, lambda, lower limit, upper limit
Observations	Length of dugong at a given age
Goals	Predicting the length of dugongs at different ages

Table 7: Dugongs Environment

In this environment, the length of a dugong at age x is modeled using a hierarchical Bayesian model with parameters α, β, and λ. The age values range between 0 and 5. The observed length Y at a given age x is generated from a normal distribution with a mean that is a function of x and the parameters α, β, and λ, and a fixed standard deviation. The function representing the mean length m is defined as:

$$m = \alpha - \beta \cdot |\lambda|^x \tag{8}$$

The observed lengths are then drawn from a normal distribution:

$$Y \sim \mathcal{N}(m, \sigma) \tag{9}$$

where σ is the noise in the observed lengths, set to a fixed value (e.g., 0.25).

D.6 Peregrines

The peregrine environment models the population count of peregrine falcons at different times [30]. The goal is to understand how the population changes over time. The following table describes the inputs and outputs of the experiment:

In this environment, the population count of peregrine falcons at time t is modeled using a Poisson regression model with parameters α, β_1, β_2, and β_3 . The time values range between 0 and 5. The population count C at a given time t is generated from a Poisson distribution with a mean that is a

Parameter	Description
Model	Poisson regression model
Setup Parameters	Regression params: α, β_1, β_2, and β_3
Observations	Population count of peregrine falcons at a given time
Goals	Predicting the population of peregrines at different times

Table 8: Peregrine Environment

function of t and the parameters α, β_1, β_2, and β_3. The function representing the log of the mean population count λ is defined as:

$$\log \lambda = \alpha + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \tag{10}$$

The observed population counts are then drawn from a Poisson distribution:

$$C \sim \text{Poisson}(\exp(\log \lambda)) \tag{11}$$

This model allows for capturing the non-linear trends in the population data over time.

D.7 Survival Analysis: Mastectomy

The survival analysis environment models the outcomes of breast cancer patients based on the time since surgery and the metastasized status. The following table describes the inputs and outputs of the experiment:

Parameter	Description
Model	Survival analysis using a Bayesian approach
Setup Parameters	num_patients, time_upper_bound, lambda, beta
Observations	Whether a selected patient is alive or dead
Goals	Predict survival based on time since surgery and if the cancer had metastasized

Table 9: Survival Analysis Environment

In this environment, the outcome (alive or dead) of a patient is modeled based on the time since surgery and whether the cancer metastasized [13]. The outcomes are generated using a Bayesian model with parameters λ_0 and β. The number of patients and the upper bound of the time since surgery are configurable. At the start of an episode, we sample a set of patients that have undergone mastectomy, with varying times since they had surgery and if their cancer had metastasized or not. The experimenter can then choose to observe specific patients to see if they are alive or dead. The probability of death is calculated using the following model:

$$\lambda = \exp(\beta \cdot \text{metastasized}) \cdot \lambda_0 \mu = \text{time_since_surgery} \cdot \lambda \tag{12}$$

The probability of death for a patient is given by the logistic function:

$$p(\text{death}) = \frac{1}{1 + \exp(-\mu)} \tag{13}$$

Each patient's outcome is simulated from a Bernoulli distribution with the calculated death probability. The observed data consists of tuples indicating whether the patient died, the time since surgery, and the metastasized status.

For example, for a patient with a given time since surgery and metastasized status, the death outcome is sampled as follows:

$$\text{death_outcome} \sim \text{Bernoulli}(p(\text{death})) \tag{14}$$

D.8 Predator-Prey Dynamics

The predator-prey environment models the interaction between populations of predators and prey over time using the Lotka-Volterra equations [52]. The following table describes the inputs and outputs of the experiment:

Parameter	Description
Model	Lotka-Volterra equations
Setup Parameters	Initial prey population, initial predator population, α, β, γ, and δ
Observations	Populations of prey and predators at a given time
Goals	Predicting populations

Table 10: Predator-Prey Environment

In this environment, the populations of prey and predators at time t are modeled using the Lotka-Volterra equations. The initial populations of prey and predators are given by the parameters 'prey_init' and 'predator_init', respectively. The interaction between the populations is governed by the parameters α, β, γ, and δ. The time values range between 0 and 50. The Lotka-Volterra system of differential equations is defined as follows:

$$\frac{d\text{prey}}{dt} = \alpha \cdot \text{prey} - \beta \cdot \text{prey} \cdot \text{predator} \tag{15}$$

$$\frac{d\text{predator}}{dt} = \delta \cdot \text{prey} \cdot \text{predator} - \gamma \cdot \text{predator} \tag{16}$$

The populations of prey and predators at any given time t are obtained by solving these differential equations. The observed data consists of tuples indicating the time and the populations of prey and predators at that time.

For example, for a given time t, the populations of prey and predators are computed by solving the Lotka-Volterra equations with the specified parameters and initial populations. The resulting populations are nonnegative integers representing realistic population counts.

D.9 Emotions from Outcomes

The Emotions from Outcomes environment models a participant's predictions of a players emotions after spinning a wheel with three possible monetary outcomes [38]. The model considers the actual outcome, the expected outcome, and the absolute difference between the actual and expected outcomes. The following table describes the inputs and outputs of the experiment:

Parameter	Description
Model	Forward regression model with priors for emotional response
Setup Parameters	Prize values, probabilities, outcome, LLM
Observations	Prediction in natural language of how a player feels and why
Goals	Predicting what a participant thinks a player feels on a likert scale of 8 emotions.

Table 11: Emotions From Outcomes Environment

In this environment, the participant's predictions of a player's emotions are modelled after observing the outcome of the player spinning a wheel with three possible prizes. Each outcome has a known probability and monetary value. The emotion predictions are influenced by the actual outcome, the difference between the actual outcome and the expected outcome, and the absolute difference between the actual outcome and the expected outcome.

The model uses the following parameters:

1. Prize values: The monetary values of the three possible outcomes.
2. Probabilities: The probabilities of each outcome occurring.
3. Outcome: The actual outcome of the wheel spin.

The emotions are measured on a Likert scale from 1 to 9 for the following eight emotions: Happiness, Sadness, Anger, Surprise, Fear, Disgust, Contentment, Disappointment

The emotional response is generated based on the following model:

$$\text{mean} = \alpha + \beta_{\text{win}} \cdot \text{win} + \beta_{\text{PE}} \cdot \text{PE} + \beta_{\text{absPE}} \cdot \text{absPE} \tag{17}$$

where:

- α are the intercepts for each emotion.
- β_{win} are the coefficients for the actual outcome.
- β_{PE} are the coefficients for the prediction error (PE).
- β_{absPE} are the coefficients for the absolute prediction error (absPE).

For each emotion, the value is sampled from a normal distribution with the computed mean and a predefined standard deviation.

The generative model produces Likert scale ratings for the 8 emotions for the participant's predictions of what a player would feel. These predictions are translated into free-form natural language observations by a language model with the prompt shown in Fig. 8. For example, an observation when the prizes are \$50, \$20, \$10 with probabilities 0.1, 0.4, 0.5, and the player wins \$50, the simulated participant responds with "The player might be feeling quite happy and content because they landed on the highest possible outcome, which was unexpected given its low probability."

D.10 Moral Machines

The Moral Machine environment Awad et al. [5] models participants' decisions in moral dilemmas involving autonomous vehicles. Participants are presented with scenarios where the vehicle must decide between two outcomes, each involving the death of a different group of characters. The following table describes the inputs and outputs of the experiment:

Parameter	Description
Model	Logistic regression model with priors for moral decision-making
Setup Parameters	Character attributes, intervention type, LLM
Observations	Prediction in natural language of which group to save and why
Goals	Predicting which group participants choose to save

Table 12: Moral Machines Environment

In this environment, participants must decide which group of characters to save in a moral dilemma involving autonomous vehicles. The characters in each group can be any of the following: stroller, boy, girl, pregnant_woman, male_doctor, female_doctor, female_athlete, male_athlete, female_executive, male_executive, large_woman, large_man, homeless, old_man, old_woman, criminal, dog, cat.

The model uses the following parameters:

1. Character attributes: gender, age, social status, fitness, species (human or pet).
2. Intervention type: 'swerve' or 'stay'.

The decision to save a group is influenced by the difference in attributes between the two groups and the intervention required. The logistic regression model considers the following coefficients:

LLM prompt to translate predictions from the generative model to observations

```
You are observing a user play a game where they spin a wheel.
The wheel has three possible outcomes (monetary values), and the probabilities of landing on each are known to you and the player.
You are observing the player play the game and the outcomes.
You are asked to predict how the player feels after each spin of the wheel.
Translate the values for emotions to a sentence that describes the player.
The decisions are based on the following model and features:
- Your predition of the player's happiness, sadness, anger, surprise, fear, disgust, contentment, and disappointment are influenced by a few factors.
- The player's emotions are influenced by the actual outcome of the spin.
- The player's emotions are influenced by the difference between the actual outcome and the expected outcome.
- The player's emotions are influenced by the absolute difference between the actual outcome and the expected outcome.
The wheel has three possible outcomes with the following probabilities:
{v1:0.2f}: {p1:0.2f}
{v2:0.2f}: {p2:0.2f}
{v3:0.2f}: {p3:0.2f}
The player has spun the wheel and landed on {outcome}.
This is how you think the player feels:
Happiness: {happiness}/9
Sadness: {sadness}/9
Anger: {anger}/9
Surprise: {surprise}/9
Fear: {fear}/9
Disgust: {disgust}/9
Contentment: {contentment}/9
Disappointment: {disappointment}/9
Translate the values for emotions to a sentence that describes the player.
1: Not at all, 9: Very much
This sentence should be concise and describe the player's emotions after the spin.
The sentence should be a few words long and should not contain any numbers or refer to the numbers directly.
Only talk about the most salient emotions.
Start with "The player might be feeling...because..." and provide a description of the player's emotions and a reason.
```

Figure 8: **LLM prompt for simulated participant.** LLM prompt to translate predictions from the generative model to observations in free-form natural language.

- $\beta_{\text{intervention}}$: Preference for inaction.
- β_{group}: Preference for group 1 (passengers).
- β_{gender}: Preference for sparing females.
- β_{fitness}: Preference for sparing the fit.
- $\beta_{\text{social_status}}$: Preference for sparing higher status individuals.
- β_{age}: Preference for sparing the young.
- $\beta_{\text{human_count}}$: Preference for sparing more characters.

- β_{species}: Preference for sparing humans over pets.

The logistic regression model generates a choice for which group to save based on the computed attributes and intervention. These predictions are translated into free-form natural language observations by a language model with the prompt shown in Fig. 9. For example, in a scenario where group 1 consists of a boy and a girl, and group 2 consists of an elderly man and an elderly woman, with the intervention being 'swerve', the simulated participant responds with "I choose to save group 1 because they are younger and have more potential life ahead of them."

LLM prompt to translate predictions from the logistic regression model to observations

```
You have to make choices in a moral dilemma involving autonomous vehicles. In each dilemma, the vehicle must decide between two outcomes, each involving the death of a different group of characters.
There are two groups of characters, and the vehicle must decide which group to save. You can query the participant for their decision by providing the descriptions of the characters in each group and the intervention the car would take ('swerve' or 'stay' to save group 1).
The characters can be any of the following: {characters}
Translate the decision that is shown to you in natural language and provide reasons for it.
The importance that you place on the different attributes of the characters affects your decision. Values range from -1 to 1, with 0 being neutral.
Your preferences are as follows:
- Preference for human count: {beta_human_count}
- Preference for group 1 (passengers): {beta_group}
- Preference for males: {beta_gender}
- Preference for young: {beta_age}
- Social status: {beta_social_status}
- Preference for Fitness: {beta_fitness}
- Preference for humans over pets: {beta_species}
- Preference for having to take an action: {beta_intervention}
DO NOT talk about the values of preferences directly in your response. Only talk about the most salient reasons for the decision. Be concise, write one sentence only.
Translate the decision that is shown to you in natural language and provide reasons for it.
The scenario presented to you was as follows:
Group 1: {group1}
Group 2: {group2}
Intervention: {intervention}
Choice made: {choice}
Translate the decision that is shown to you in natural language and provide reasons for it.
Only talk about the most salient reasons for the decision.
The sentence should be concise and should not refer to the importance factors/values directly.
Start with "I choose to save group 1/group 2 because...". Be concise and faithful to the importance weights.
```

Figure 9: **LLM prompt for simulated participant.** LLM prompt to translate predictions from the logistic regression model to observations in free-form natural language.

E Qualitative Examples

Box's Apprentice proposed programs

```
with pm.Model() as model:
        # Priors for student abilities and question difficulties
        student_ability = pm.Normal("student_ability", mu=0, sigma=1, shape=6)
        question_difficulty = pm.Normal("question_difficulty", mu=0, sigma=1, shape=6)

        # Linear combination of student ability and question difficulty
        logit_p = student_ability[student_id] - question_difficulty[question_id]

        # Likelihood
        y_obs = pm.Bernoulli("y_obs", logit_p=logit_p, observed=correctness, dims="obs_id")
```

```
with pm.Model() as model:
        # Define priors for the parameters of the linear model
        intercept = pm.Normal("intercept", mu=0, sigma=10)
        slope = pm.Normal("slope", mu=0, sigma=10)

        # Define the linear model for the mean of the Poisson distribution
        lambda_ = pm.math.exp(intercept + slope * year_data)

        # Define the likelihood
        y_obs = pm.Poisson("y_obs", mu=lambda_, observed=C, dims="obs_id")
```

```
with pm.Model() as model:
        # Define the locations of the three sources as latent variables
        source_x1 = pm.Uniform("source_x1", lower=0, upper=1, shape=3)
        source_x2 = pm.Uniform("source_x2", lower=0, upper=1, shape=3)

        # Calculate the distances from each source to each observation point
        distances = pm.math.sqrt((x1_data[:, None] - source_x1[None, :])**2 +
                                 (x2_data[:, None] - source_x2[None, :])**2)

        # Model the signal strength as a function of the distances
        signal_strength_mu = pm.math.sum(1 / distances, axis=1)

        # Define the likelihood of the observations
        y_obs = pm.Normal("y_obs", mu=signal_strength_mu, sigma=1, observed=signal_strength, dims="obs_id")
```

Figure 10: BoxLM **proposed programs. (top)** IRT **(middle)** Peregrines **(bottom)** Location finding

Example of an explanation produced by the LLM Agent for the IRT Environment

```
Based on the observations made, we can infer some patterns about
     the students and questions:

1. **Question 1**: Students 0, 1, 2, and 3 answered question 1
     correctly, while student 4 answered it incorrectly. This
     suggests that question 1 is generally easier for most
     students, but not universally so.

2. **Question 0**: Students 2, 3, and 4 answered question 0
     correctly. This indicates that question 0 might also be
     relatively easy for students, though we lack data for
     students 0, 1, and 5.

3. **Student 1**: Answered question 1 correctly but answered
     question 2 incorrectly. This suggests that student 1's
     performance may vary depending on the question.

4. **Student 4**: Answered question 0 correctly but answered
     question 1 incorrectly. This indicates that student 4's
     performance also varies by question.

From these observations, it appears that some questions (like
     question 1 and question 0) might be easier for a majority of
     students, but individual performance can vary. Without more
     data, it's challenging to make definitive predictions for all
      student-question pairs, but these patterns provide a
     starting point for understanding the environment.
```

Figure 11: **Example Explanation.** Example of an explanation produced by the LLM Agent for the IRT Environment.

Example of an explanation produced by Box's Apprentice for the IRT Environment

```
Based on the observations and the statistical model, we can
    infer that each student's ability and each question's
    difficulty play significant roles in determining the
    correctness of answers. The model uses these factors to
    predict the probability of a correct response.

**Student Abilities:**
- Students have varying abilities, with some being more likely
    to answer correctly than others. For instance, student 2 has
    a higher ability (0.544) compared to student 0 (-0.405).

**Question Difficulties:**
- Questions also vary in difficulty. For example, question 1 is
    relatively easier (-0.873) compared to question 3 (0.492).

**Predicted Probabilities:**
- The model calculates the probability of a correct answer by
    combining student ability and question difficulty. For
    example, student 2 answering question 1 has a high
    probability of correctness (0.708), while student 0 answering
     question 3 has a lower probability (0.414).

**General Patterns:**
- Students with higher abilities are more likely to answer
    correctly across various questions.
- Easier questions are more likely to be answered correctly by
    most students.

To predict if a student will answer a question correctly,
    consider both the student's ability and the question's
    difficulty. Higher student ability and lower question
    difficulty increase the likelihood of a correct answer.
```

Figure 12: **Example Explanation.** Example of an explanation produced by the Box's Apprentice for the IRT Environment.