AgentBayes: Open-Ended Scientific Model Discovery

Alexander R. Farhang* **Anne L. Erickson*** **Atharva Sehgal*** **Yisong Yue***

Abstract

Scientific data modeling often requires more than predicting observations; it must recover the probabilistic data-generating structure: how observations are nested, how variation arises at each level, and how uncertainty propagates through to predictions. We introduce AgentBayes, an agentic system for Bayesian model discovery on real scientific datasets. AgentBayes automates the iterative Bayesian workflow through two roles: an Interactor that explores raw data and fitted posteriors through executable analysis, and a Modeler that converts these findings into hierarchical probabilistic programs. Because data and posterior samples remain in an executable sandbox rather than being serialized into LLM context, AgentBayes can critique and revise models at dataset scales where prior agentic Bayesian systems exhaust the context window. Across five `posteriordb` benchmarks and three scientific case studies, AgentBayes matches or improves on expert-written Bayesian models on most datasets, scales to larger datasets than prior agentic Bayesian systems, and outperforms symbolic regression baselines on data with experimental hierarchy. AgentBayes also surfaces overlooked data structure in existing benchmarks, and adapts its probabilistic programs accordingly. Together, these results show that LLM agents can aid scientists by performing the full Bayesian workflow, from exploratory data analysis to open-ended posterior predictive checks, on real-world, large-scale scientific data, broadening LLM-based scientific discovery from equation search to full probabilistic modeling.

1 Introduction

A central challenge in scientific data modeling is recovering the full data generating process, not just accurate predictions. Scientific measurements nest within individuals, individuals within groups, and noise enters at every level [Gelman and Hill, 2006, Gelman et al., 2013]. Without modeling this, individual variation collapses into group effects, signal into noise, and uncertainty fails to propagate. Hierarchical Bayesian models address this by jointly handling measurement uncertainty, selection effects, and population structure, and have enabled landmark results across astrophysics, planetary science, and epidemiology [Mandel et al., 2019, Thrane and Talbot, 2019, Foreman-Mackey et al., 2014, Faria et al., 2014].

But in practice, Bayesian modeling remains an expert-driven time intensive workflow. Scientists and statisticians explore the raw data, decide the latent experimental structure that is important, write a probabilistic program, fit the model, inspect diagnostics and posterior predictive simulations, and revise the model when it fails [Gelman et al., 2020]. Each pass through this loop can take days to weeks of statistician time, and the resulting overhead is a real barrier to adoption: scientists with hierarchically structured data often default to simpler flat models, or develop bespoke pipelines that are slow to iterate on.

*Caltech. Correspondence to `afarhang@caltech.edu`, `yyue@caltech.edu`.

Preprint.

This Bayesian workflow is powerful precisely because it is iterative and grounded in domain expertise, but those same properties make it difficult to automate.

Large language models (LLMs) are increasingly used to automate parts of scientific discovery, from hypothesis generation and code writing to equation search and tool-using agents [Huang et al., 2025a, Romera-Paredes et al., 2024, Huang et al., 2025b]. In data modeling, the most developed thread is equation discovery: finding a compact symbolic expression that explains observed data [Makke and Chawla, 2024, Shojaee et al., 2025, Xia et al., 2025]. This is useful when the modeling target is a single predictive relationship, but Bayesian workflow demands more: discovering probabilistic structure in the data, fitting candidate models, checking posterior behavior, and revising the model when those checks fail. Existing LLM-based Bayesian modeling systems automate only parts of this loop (Table 1). Some focus on generating candidate equations; others generate probabilistic programs, but still treat modeling as a mostly one-shot proposal-and-score problem [Domke, 2025]. Others support model critique, but depend on inserting the full dataset or hard-coded predictive pointwise statistics into the LLM's context window, preventing scaling to large dataset sizes [Li et al., 2024a]. This leaves the central challenge from the Bayesian workflow unresolved: the system must inspect the data and fitted model, diagnose what structure the model misses, and revise the next probabilistic program, all while being able to scale to large scientific datasets.

Our Contributions. We introduce AgentBayes, an agentic system that automates the iterative Bayesian workflow. The central design choice is to keep raw data and fitted posteriors accessible through programmatic interaction, with two LLM agents exchanging only compact structured findings. An Interactor agent explores data and critiques models, while a Modeler agent generates and fits probabilistic programs. This separation lets AgentBayes propose and refine models at dataset scales where prior LLM-driven Bayesian methods fail, while preserving open-ended iteration with both data and posteriors.

We evaluate AgentBayes in two ways. First, we use `posteriordb`, a benchmark containing a collection of Bayesian models and datasets to compare against expert-written probabilistic programs [Magnusson et al., 2024]. Second, we use AgentBayes to reanalyze real scientific datasets with rich hierarchical experimental structure, including olfactory dose responses, residential radon measurements, and animal-behavior kinematics [Si et al., 2019, Magnusson et al., 2024]. Across these settings, AgentBayes matches or improves on expert-written Bayesian models in most benchmarks, handles datasets beyond the scale of existing agentic approaches for Bayesian modeling, and outperforms recent LLM-based symbolic-regression baselines [Shojaee et al., 2025, Xia et al., 2025]. In the case studies, AgentBayes also discovers previously overlooked patterns and uses them to propose probabilistic model structures that better match the data.

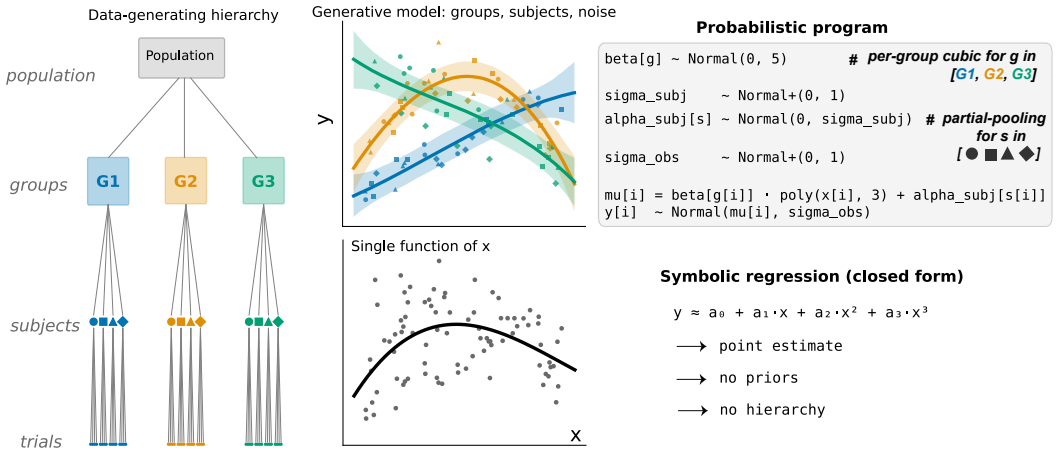

Figure 1: Modeling the data generating process. *Left*: schematic of the experimental hierarchy. *Top*: group-wise posterior mean functions with 90% credible intervals and corresponding pseudocode of the probabilistic program modeling the data (*right*). *Bottom*: fitting a model with a single equation.

2 Background

AgentBayes builds on two lines of work that have addressed complementary parts of scientific data modeling. Bayesian hierarchical modeling gives scientists a language for representing the full data generating process of observed data: experimental structure, uncertainty, and variation across related observations. LLM-based scientific discovery provides tools for automating parts of modeling, including code generation, equation search, and agentic data analysis. These capabilities have not yet been brought together into a scalable, automated Bayesian workflow.

2.1 Bayesian inference and probabilistic programming

Probabilistic modeling treats observed outcomes as samples from a data generating process. Given observations y and covariates X, a model specifies a likelihood $p(y \mid \theta, X)$ and a prior $p(\theta)$ over unknown parameters. Bayesian inference then yields a posterior distribution,

$$p(\theta \mid y, X) \propto p(y \mid \theta, X)p(\theta),$$

rather than a single point estimate, capturing uncertainty about parameters, which can then be propagated through to predictions and downstream scientific conclusions [Gelman et al., 2013]. In this paper, a candidate model is a probabilistic program m that specifies the likelihood, priors, latent variables, and hierarchical structure, together defining a joint distribution from which the posterior is obtained.

Figure 1 illustrates how a probabilistic program represents a data generating process and does more than predict y from x: group-specific coefficients, partially pooled subject effects, and explicit observation noise jointly specify the latent quantities, hierarchical dependencies, priors, and likelihood. A flat alternative collapses these sources of variation into a single function of the covariates, with no pooling and no uncertainty to propagate downstream.

When the probabilistic program encodes a hierarchical structure, as in Figure 1, it allows the sharing of information across related experimental units. Rather than fitting each subject independently (no pooling) or treating all subjects identically (complete pooling), the model draws lower-level parameters from shared group- or population-level distributions, enabling partial pooling [Gelman and Hill, 2006, Gelman et al., 2013]. Sparsely observed units can borrow statistical strength from related units while still retaining their own effects. Partial pooling is often the key modeling affordance for scientific data, where the relevant structure is not only the relationship between covariates and outcomes, but also how variation is organized across groups, subjects, and measurements.

Fitting a probabilistic program requires an inference procedure. Probabilistic programming languages provide both a compact modeling language and algorithms for drawing posterior samples from the specified model [Carpenter et al., 2017]. In this paper, AgentBayes generates Stan programs and uses the No-U-Turn sampler (NUTS) for posterior inference [Hoffman and Gelman, 2014, Carpenter et al., 2017, Stan Development Team, 2026]. Once fit, candidate models can be compared by predictive criteria such as the leave-one-out expected log predictive density (ELPD), and inspected through posterior predictive checks [Vehtari et al., 2017, Gelman et al., 2020]. These checks compare synthetically generated datasets from the fitted model against the observed data, testing whether the proposed data-generating structure captures the patterns the model is meant to explain. When they reveal a mismatch, deciding how to revise the likelihood, hierarchy, covariates, or priors is usually part of an iterative human modeling workflow.

2.2 LLM-based model discovery

Recent advances in machine learning incorporate LLMs into various stages of scientific modeling and discovery, including hypothesis generation, code writing, program search, equation discovery, and tool-using data analysis [Grayeli et al., 2024, Novikov et al., 2025, Shojaee et al., 2025, Xia et al., 2025, Wang et al., 2024, Majumder et al., 2024, Zheng et al., 2023, Ma et al., 2024, Huang et al., 2025a,b]. These systems show that language models can search large spaces of scientific artifacts when paired with execution, scoring, or feedback. For data modeling, the most relevant thread is symbolic regression: searching for a compact

Table 1: Desired capabilities of LLM-driven data modeling agents for real scientific tasks.

	Real-world data scale	Models multi-level experimental structure	Models uncertainty	Empirical interaction with data
Symbolic Regression				
LLM-SR	✔	✗	✗	✗
SR-Scientist	✔	✗	✗	✔
Bayesian Inference				
BoxLM	✗	✔	✔	✗
AgentBayes	✔	✔	✔	✔

equation or program that predicts observed measurements [Makke and Chawla, 2024, Shojaee et al., 2025, Xia et al., 2025, Merler et al., 2024].

Symbolic regression (SR) is a natural fit when the target object is a predictive relationship, such as a closed-form expression mapping covariates to outcomes. As illustrated in Figure 1, the goal in the Bayesian setting is not only to predict y from x, but to recover a probabilistic data-generating structure: likelihoods, latent variables, hierarchical dependencies, priors, and uncertainty about unknown quantities. This distinction matters for scientific datasets with repeated measurements, grouped observations, heterogeneous noise, or sparsely observed experimental units, where the structure of variation is part of the scientific model.

Recent work uses LLMs for Bayesian modeling directly, by eliciting priors, translating natural language descriptions into probabilistic programs, or generating candidate models [Selby et al., 2024, Domke, 2025, Li et al., 2024a,b, Capstick et al., 2025], building on pioneering pre-LLM efforts like the Automatic Statistician [Lloyd et al., 2014]. However, most existing systems still treat modeling as a proposal-and-score problem: they generate candidate models, evaluate them with fixed [Li et al., 2024a] or limited critique [Li et al., 2024b], and select among the candidates.

This capability gap is summarized in Table 1. LLM-driven SR methods can target large scale real-world data and can produce useful predictive equations, but they do not model uncertainty or multi-level experimental structure. Existing LLM-based Bayesian systems target probabilistic models, but do not yet combine scalable interaction with raw data, posterior samples, and diagnostics to enable model refinement. AgentBayes is designed to target all of these essential capabilities.

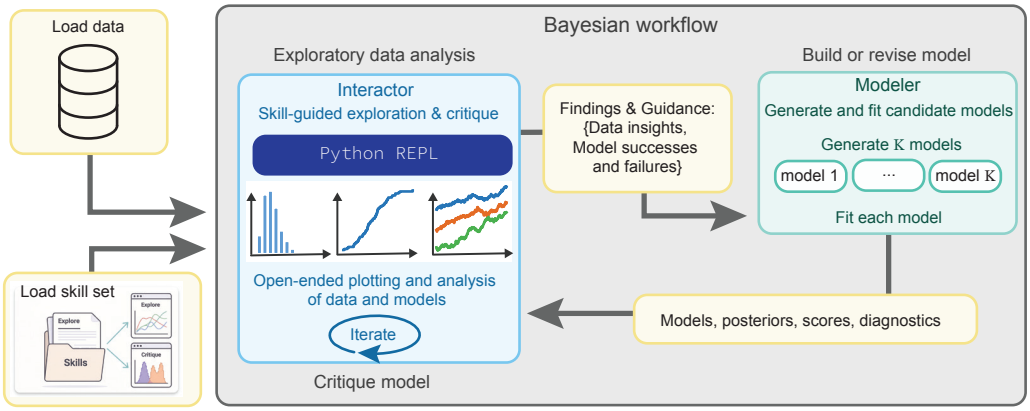

Figure 2: AgentBayes architecture. The Interactor agent (left) operates a multimodal Python sandbox prepopulated with raw data and, in critique mode, fitted posteriors; it emits a lightweight report to the Modeler agent (right) which synthesizes K modeling directions, generates Stan programs in parallel, and fits each candidate. ELPD and convergence diagnostics return to the Interactor for the next iteration.

3 AgentBayes

Real-world Bayesian modeling is rarely a one-shot fit. A scientist explores the data, fits a model, critiques it via posterior predictive checks, and revises the model when failures reveal missing or inappropriate model features [Blei, 2014, Gabry et al., 2019, Vehtari et al., 2017, Gelman et al., 2020]. AgentBayes automates this loop with two interacting agents, shown in Figure 2. The Interactor is responsible for exploratory data analysis and empirical critique: it queries raw data and fitted posteriors, produces plots and analyses, identifies patterns the current models capture or miss, and records reusable analyses and findings. The Modeler is responsible for structural revision: it converts the Interactor's findings into candidate probabilistic programs, fits those programs, and returns posterior samples, scores, and diagnostics for the next critique round. Only compact reports pass between the agents in natural language; raw data and posterior objects stay in the sandbox, accessed through executable queries rather than serialized into LLM context.

Interactor agent. The Interactor performs the empirical critique steps of the Bayesian workflow. It runs in a multi-turn, multimodal, persistent Python REPL (read-eval-print-loop) sandbox, so it can query raw data and fitted posteriors through interactive code execution. When the Interactor is loaded with the Explore skill, the sandbox is prepopulated with the dataset, so the Interactor can identify structure that candidate models should explain, such as grouping variables, trends, and heterogeneity across experimental conditions. When the Interactor is loaded with the Critique skill, the sandbox also contains the fitted candidate models, posterior samples, scores, and diagnostics from the previous iterations, allowing it to compare model predictions against the data and diagnose model failures. Plotted figures are rendered to the LLM the next turn, enabling visual posterior predictive checks and data-model comparisons. The Interactor can also save useful analysis and plotting functions into a shared analytical library $\mathcal{L}$ that is made available to later iterations. After completing a multiturn interrogation of the data or models, the Interactor emits a compact structured report containing findings, modeling recommendations, and per-candidate assessments; this report is the only conversational artifact passed to the Modeler.

Modeler agent. The Modeler converts the Interactor's empirical findings into candidate probabilistic programs. At each iteration, it reads the latest Interactor report together with the history of previous candidates and fits, then synthesizes K structurally diverse modeling directions, varying components like the likelihood family, hierarchy, covariates, or parametrization. Separate LLM calls generate Stan programs from these directions [Carpenter et al., 2017]; compilation or runtime failures are routed to a dedicated repair step with Stan-specific guidance. Surviving candidates are fit in parallel with Markov Chain Monte Carlo (MCMC) using the No-U-Turn Sampler [Hoffman and Gelman, 2014]. For each candidate, the system records posterior samples, leave-one-out expected log predictive density (ELPD-LOO) [Vehtari et al., 2017], convergence diagnostics ($\hat{R}$, effective sample size), HMC-specific pathology indicators (divergent transitions), and an audit tier indicating whether the fit converged and produced finite ELPD estimates. All candidates, including failed or nonconverged fits, are returned to the Interactor in the next critique session, since failures can reveal systematic misunderstandings about the data or model structure.

Cost. Because the Interactor queries data programmatically inside its sandbox and surfaces what it learns as structured, tagged findings, the per-turn payload passed to the Modeler is bounded by the report's content, not by $|\mathcal{D}|$. A run on N=12,573 observations (Radon's size) can emit the same per-turn payload as one on N=12. This removes the dataset-serialization barrier of prior agentic Bayesian work [Li et al., 2024a], which inserts the full dataset and per-observation posterior predictive means and variances into every agent prompt, overflowing the context window as dataset sizes grow.

4 Experiments

We evaluate AgentBayes using three comparison paradigms and one ablation study:

- In § 4.1, we evaluate on `posteriordb` [Magnusson et al., 2024], a benchmark pairing real datasets with expert-written Bayesian models, and compare against both the expert references and BoxLM [Li et al., 2024a]. We show that AgentBayes can recover

Algorithm 1 AGENTBAYES outer loop.

Input: dataset $\mathcal{D}$; iterations N; candidates per iteration K; library $\mathcal{L}$; skill set $\mathcal{S} = \{s_{\text{explore}}, s_{\text{critique}}\}$.
Output: selected program $m^\star$ and its fitted posterior.

$r_0 \leftarrow \textsc{Interactor}(\mathcal{D}, \mathcal{L};\ s_{\text{explore}})$
for $t = 1, \ldots, N$ **do**
  $\{m_1, \ldots, m_K\} \leftarrow \textsc{Modeler}(r_{t-1}, \text{history})$ ▷ plan, generate, compile-fix
  $\{(\text{fit}_k, \ell_k)\} \leftarrow \textsc{Fit}(\{m_k\})$ ▷ NUTS, ELPD, Diagnostics
  $r_t \leftarrow \textsc{Interactor}(\mathcal{D}, \{m_k, \text{fit}_k, \ell_k\}, \mathcal{L};\ s_{\text{critique}})$
end for
return $m^\star \leftarrow \arg\max_k \ell_k$ over surviving candidates.

competitive probabilistic programs on standard Bayesian benchmarks and scale beyond prior LLM-based Bayesian modeling systems.

- In § 4.2, we evaluate on real scientific datasets with nested experimental structure and compare against LLM-driven symbolic regression systems. We show that explicitly modeling likelihoods, hierarchy, and uncertainty yields gains that a single predictive equation cannot capture.
- In § 4.3, we present case studies that trace these gains to concrete discoveries. We show that AGENTBAYES can identify hidden structure in real datasets and use it to improve the probabilistic model. Additionally, in § 4.4, we show that removing interaction with data by ablating the Interactor results in a loss of these discoveries.

4.1 Comparisons to Expert Bayesian Models

Design. We evaluate AGENTBAYES on five datasets from `posteriordb` [Magnusson et al., 2024], a benchmark pairing real-world datasets with expert-written Stan programs [Carpenter et al., 2017]: **Eight Schools**, average SAT test score improvements after an intervention, **Dugongs**: the ages and lengths of dugongs, **Surgical**: pediatric cardiac surgery mortality rates in twelve hospitals, **Peregrine**: the counts of peregrine breeding pairs in eastern France from 1964 to 2003, and **Radon**: a large dataset measuring house-level radon and county-level uranium concentrations in US counties. We compare AGENTBAYES against the `posteriordb` expert reference and a reimplementation of BoxLM [Li et al., 2024a]. We report ELPD [Vehtari et al., 2017], the standard predictive criterion for Bayesian models, which scores held-out log-likelihood and is well-defined for all three methods.

Observations. AGENTBAYES is competitive with expert-written Bayesian programs on standard `posteriordb` benchmarks, matching or improving the expert reference on four of five datasets (Table 2). It is also able to scale to the full Radon dataset, where BoxLM's critic overflows the context window by serializing the pointwise posterior feedback into the prompt(§ **??**). The Radon result, a 730 point ELPD improvement over the expert, is driven by structural data features AGENTBAYES discovers through interactive data exploration; we show in §4.3 that removing the Interactor misses these features and degrades performance.

Table 2: ELPD comparison (higher is better) with expert-derived models on `posteriordb` benchmark datasets. Bold indicates the best value per row, underlined indicates the value outperformed the expert. AgentBayes beats the expert reference on the majority of datasets.

Dataset	Expert	BoxLM	AGENTBAYES (ours)
Eight Schools	−30.70	−30.89	**−30.51**
Dugongs	22.52	23.42	**24.90**
Surgical	−40.29	−38.97	**−38.49**
Peregrine	**−142.19**	−152.59	−148.23
Radon (all, $N = 12{,}573$)	−17,020.07	—[†]	**−16,285.98**

[†]BoxLM cannot scale to a dataset of this size; see §**??**.

4.2 Comparisons with Symbolic Regression

Motivation. §4.1 shows that AgentBayes matches or exceeds expert Bayesian models on standard Bayesian benchmarks. We now ask whether AgentBayes's representational machinery (e.g. partial pooling, explicit noise models, and hierarchical priors) yields concrete gains on scientific datasets with complex structure. LLM-driven symbolic regression is the closest automated model-discovery baseline, targeting a closed-form predictor $f(x)$ rather than probabilistic data generating process. This comparison tests whether the broader probabilistic modeling target yields gains beyond equation discovery.

Design. We compare AgentBayes against two state of the art LLM-driven symbolic regression methods, LLM-SR [Shojaee et al., 2025] and SR-Scientist [Xia et al., 2025], on three scientific datasets with complex experimental structure: **Radon**(see 4.1), **Odorant** (activity of *Drosophila* larval olfactory neurons across odorants and concentrations [Si et al., 2019]), and **Wingbeat** (per-wingbeat wingstroke-angle kinematics of flying flies; an unpublished dataset using the experimental technique of Melis et al. [2024]). We report training-set NMSE and R^2 as the common-denominator metric: ELPD-LOO is already out-of-sample via PSIS and needs no held-out split, while SR baselines produce point estimates for which ELPD is undefined. For AgentBayes, we compute R^2 of the posterior predictive mean. BoxLM cannot be evaluated because its critic payload exceeds the context window on all three datasets and cannot be compared (§3, **??**).

Table 3: Performance of AgentBayes vs state of the art LLM-driven symbolic regression methods on three scientific datasets.

	Odorant		**Wingbeat**		**Radon**	
	NMSE ↓	R^2 ↑	NMSE ↓	R^2 ↑	NMSE ↓	R^2 ↑
Symbolic Regression						
LLM-SR	0.760	0.240	0.242	0.758	0.832	0.168
SR-Scientist	0.820	0.180	0.305	0.695	0.841	0.159
Bayesian Inference						
BoxLM	—	—	—	—	—	—
Agent Bayes (ours)	**0.257**	**0.743**	**0.048**	**0.952**	**0.672**	**0.328**

Observations. AgentBayes outperforms both symbolic regression approaches on all three datasets (Table 3). The common feature is that each experiment has structure: counties in Radon, receptor–odorant pairs in Odorant, and wingbeats nested within trials, flies, and genotypes in Wingbeat (Figure **??**). On data with non-trivial generative structure, like many scientific experiments, recovering the data generating process (likelihood, hierarchy, and noise) yields measurable gains a flat functional form cannot match (analogous to the example in Fig. 1). Our case studies in 4.3 provide further analysis of AgentBayes's modeling choices, including how uncovering previously hidden structure, can lead to improved models.

4.3 Case Studies

4.3.1 Radon - Identifying and accommodating data sentinels

Motivation. A common challenge when modeling real-world data is the existence of unknown values encoded via sentinels, placeholder codes that look like ordinary measurements but are not. A system that treats every input at face value will silently fold those artifacts into its modeling. **Finding.** In the Radon benchmark from `posteriordb` [Magnusson et al., 2024, Gelman and Hill, 2006], the `floor_measure` variable corresponds to which household floor radon was detected on, with values typically between 0 (basement) and 3 (third floor); however, about 2% of the dataset has `floor_measure` values at 9. AgentBayes, through interactive data exploration, flagged 9 as *"a special code-like level (9), indicating it may be categorical with nonstandard coding rather than a continuous measurement."* Integrating this observation into the modeling process, AgentBayes encoded floors as per-level indicators with independent coefficients. **Significance.** The `posteriordb` expert model treats `floor_measure` as continuous and uses it as a numeric slope, forcing the model

to absorb a 9× artifactual contribution from these sentinel rows, biasing the basement-vs-first-floor effect estimate toward zero and penalizing its ELPD score. AgentBayes's empirical-first approach catches a class of data-quality issue that no amount of careful prior specification can fix downstream, illustrating that interactive data inspection can protect scientific modeling pipelines from preprocessing artifacts.

4.3.2 Radon - Identifying unexpected variations

Motivation. Thorough data exploration can surface violations of implicit assumptions about the data structure. When a covariate is believed to be constant within a grouping but isn't, a single-slope model will silently average over a real signal and an artifact, biasing the inferred effect. **Finding.** In the Radon benchmark, county-level uranium concentration should, by geology, be constant within each county. The expert model treats it as a row-level covariate with a single slope. AgentBayes, during data exploratory plotting, found a discrepancy: many counties have within-county uranium variation (Figure **??**). AgentBayes's explorer empirically decomposed uranium concentration via a Mundlak [Mundlak, 1978] between/within split and observed that 73 of 386 county groups exhibit within-group uranium variation that should not exist geologically. The two components had opposite signs (between = +0.2, within = −0.2): the between-county slope captures the real geological relationship, while the spurious within-county variation predicts radon in the opposite direction, a fingerprint of measurement artifact rather than signal. The expert's single-slope estimator implicitly averages these two components, dragging the inferred uranium effect toward zero. The Modeler implemented the decomposition directly: routing the within-group nuisance variance into a named parameter rather than the residual, so the between-county coefficient cleanly recovers the geological signal. **Significance.** Post hoc, we traced this to a preprocessing collision in `posteriordb`: county indices were built from name strings rather than unique FIPS codes, collapsing cross-state homonyms (e.g., multiple Franklin Counties) and affecting 28.2% of observations. Crucially, the agent's fix does not depend on identifying this cause; the Interactor observed only the symptom, and the Modeler's decomposition neutralizes the artifact whether the within-group variation is geologically real or introduced upstream. This demonstrates that empirical critique can produce a robust fix, one that handles a class of failure modes without requiring the agent to diagnose the upstream cause.

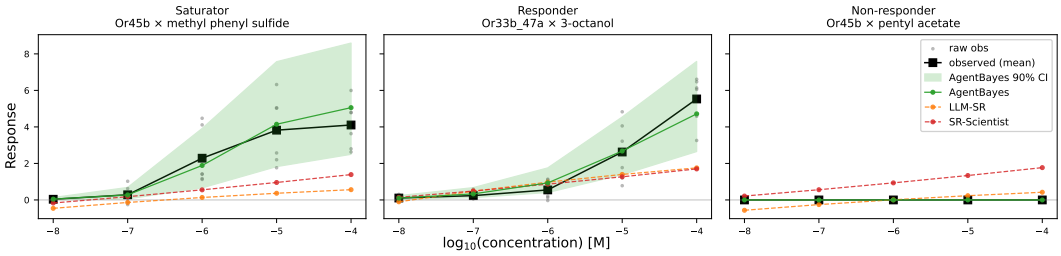

Figure 3: Odorant × Receptor examples, spanning non-responders to saturators.

4.3.3 Odorant - Hierarchical Bayesian characterization of dose-response

Motivation. Scientific datasets often contain heterogeneous response types within a single experiment, and modeling them well typically requires bespoke multi-stage pipelines that handle each subpopulation differently. Si et al. [2019] provide a representative example: characterizing neural responses of *Drosophila* larval olfactory receptor neurons to a panel of odorants, where receptor × odorant pairs span saturating responders, partial responders, and non-responders. Si et al. [2019] handle this heterogeneity through a sequence of expert-curated fitting tiers, a workflow that takes substantial researcher time and embeds many staged decisions. **Finding.** We tasked AgentBayes with modeling a representative subset of this dataset. The model it produced selects the Hill function as the response-shape primitive, an appropriate model for receptor-ligand saturation and the same one used by [Si et al., 2019]. AgentBayes proposed a single joint generative model: a hurdle component modeling whether a receptor-odorant pair responds at a given concentration, paired with a hierarchical Hill function modeling response shape conditional on activation, with crossed receptor, odorant, and pair random effects. Where the original pipeline handles weakly-responding pairs through bespoke expert classification, filtering, and fitting, AgentBayes

uses hierarchical priors and partial pooling: pairs with weak data automatically borrow strength from pairs with stronger data, without requiring upfront classification of response type. AgentBayes's formulation accommodates saturating, partial, and weak responders within one model, producing posterior predictive distributions across response types (Figure 3). The full agentic modeling loop completes in just hours compared to the typically substantial time investment of expert researchers. **Significance.** This case study illustrates AgentBayes's capacity to autonomously arrive at the modeling primitives that domain experts converge on and encode them in a unified Bayesian generative framework, replacing a staged, hand-tuned pipeline with a single hierarchical model that handles heterogeneity through partial pooling rather than upfront classification. By lowering the time cost of model construction to hours of agent time, AgentBayes can support scientists in exploring and comparing candidate models where established practice has relied on a single, staged pipeline.

4.4 Ablation Studies

To observe the direct influence of iterative code-based interaction with data and posteriors, we ablate AgentBayes's Interactor on three datasets: Radon, Odorant, and Wingbeat. All prompt information now flows directly to the Modeler. We find that ablating the Interactor agent reduces performance across the board (Table 4). Interestingly, in the Radon ablation, the method discovers neither of the features found in Sec. 4.3.1 or Sec. 4.3.2: 1) the data sentinel code (and in fact, erroneously assigns 1 as the basement code) nor 2) the within-county uranium variation.

Table 4: Interactivity improves AgentBayes performance (ELPD).

Dataset	**with** Interactor Agent	**without** Interactor Agent	ΔELPD	relative
Radon	**-16,285.98**	−16,571.64	+285.66	1.7%
Odorant	**-1,779.26**	−1,951.96	+172.70	8.8%
Wingbeat	**-17,773.12**	−19,015.41	+1,242.29	6.5%

To observe the contribution of the iterative process, we find that across the datasets, the highest performing model was found on average, at 90.5% through the iteration budget and 71.1% of datasets had the highest performing model generated on the final iteration.

5 Conclusion

AgentBayes is built upon the premise that automating scientific modeling means recovering the full data generating process, not compressing it into a single predictive function. The Interactor–Modeler split makes this tractable at scale: data and posterior samples stay in an executable sandbox, only structured findings flow between agents, and the workflow extends to datasets that overflow prior agentic Bayesian systems. AgentBayes fits into scientific practice as a collaborator that works alongside scientists to develop and compare candidate analyses. It cuts hierarchical Bayesian analysis from weeks to hours, surfaces structural anomalies in the data, and independently arrives at modeling primitives that domain experts converge on only after extensive bespoke analysis. Scientists can then explore a far broader space of candidate models than they could construct by hand, directing their expertise toward interpretation and continued improvement of models proposed by AgentBayes. In doing so, AgentBayes broadens the reach of LLM-driven scientific discovery from equation search to the full Bayesian workflow on real-world scientific data at scale.

Limitations and Societal Impacts AgentBayes targets the explicit-likelihood, MCMC-based subset of Bayesian modeling; intractable likelihoods, large discrete latent spaces, and very high-dimensional structured outputs fall outside its scope. AgentBayes inherits the biases of its underlying LLM, which may shape which models it proposes and which findings it surfaces; its outputs therefore serve as candidate models for scientist review.

References

David M. Blei. Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232, 2014. doi: 10.1146/annurev-statistics-022513-115657.

Alexander Capstick, Rahul G. Krishnan, and Payam Barnaghi. Autoelicit: Using large language models for expert prior elicitation in predictive modelling, 2025. URL https://arxiv.org/abs/2411.17284.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017. doi: 10.18637/jss.v076.i01.

Justin Domke. Large language bayes, 2025. URL https://arxiv.org/abs/2504.14025.

Nuno R. Faria, Andrew Rambaut, Marc A. Suchard, Guy Baele, Trevor Bedford, Melissa J. Ward, Andrew J. Tatem, João D. Sousa, Nimalan Arinaminpathy, Jacques Pépin, David Posada, Martine Peeters, Oliver G. Pybus, and Philippe Lemey. The early spread and epidemic ignition of HIV-1 in human populations. *Science*, 346(6205):56–61, October 2014. doi: 10.1126/science.1256739.

Daniel Foreman-Mackey, David W. Hogg, and Timothy D. Morton. Exoplanet population inference and the abundance of Earth analogs from noisy, incomplete catalogs. *The Astrophysical Journal*, 795(1):64, October 2014. doi: 10.1088/0004-637X/795/1/64.

Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2):389–402, 2019. doi: 10.1111/rssa.12378.

Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, Cambridge, 2006.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 0 edition, November 2013. ISBN 978-0-429-11307-9. doi: 10.1201/b16018. URL https://www.taylorfrancis.com/books/9781439898208.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow, 2020. URL https://arxiv.org/abs/2011.01808.

Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. 2024. URL https://arxiv.org/abs/2409.09359.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications, 2025a. URL https://arxiv.org/abs/2502.09858.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, Di Yin, Shruti Marwaha, Jennefer N. Carter, Xin Zhou, Matthew Wheeler, Jonathan A. Bernstein, Mengdi Wang, Peng He, Jingtian Zhou, Michael Snyder, Le Cong, Aviv Regev, and Jure Leskovec. Biomni: A General-Purpose Biomedical AI Agent, June 2025b. URL http://biorxiv.org/lookup/doi/10.1101/2025.05.30.656746.

Michael Y. Li, Emily B. Fox, and Noah D. Goodman. Automated statistical model discovery with language models. In *International Conference on Machine Learning (ICML)*, 2024a. arXiv:2402.17879.

Michael Y. Li, Vivek Vajipey, Noah D. Goodman, and Emily B. Fox. CriticAL: Critic automation with language models. In *NeurIPS 2024 Workshop on Statistical Frontiers in LLMs and Foundation Models*, 2024b. arXiv:2411.06590.

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models, 2014. URL https://arxiv.org/abs/1402.4304.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery, 2024. URL https://arxiv.org/abs/2405.09783.

Måns Magnusson, Jakob Torgander, Paul-Christian Bürkner, Lu Zhang, Bob Carpenter, and Aki Vehtari. posteriordb: Testing, benchmarking and developing bayesian inference algorithms. *arXiv preprint arXiv:2407.04967*, 2024. URL https://arxiv.org/abs/2407.04967.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, and Peter Clark. Data-driven discovery with large generative models, 2024. URL https://arxiv.org/abs/2402.13610.

Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1), January 2024. ISSN 1573-7462. doi: 10.1007/s10462-023-10622-0. URL http://dx.doi.org/10.1007/s10462-023-10622-0.

Ilya Mandel, Will M. Farr, and Jonathan R. Gair. Extracting distribution parameters from multiple uncertain observations with selection biases. *Monthly Notices of the Royal Astronomical Society*, 486(1):1086–1093, June 2019. doi: 10.1093/mnras/stz896.

Johan M. Melis, Igor Siwanowicz, and Michael H. Dickinson. Machine learning reveals the control mechanics of an insect wing hinge. *Nature*, 628(8009):795–803, 2024. doi: 10.1038/s41586-024-07293-4. URL https://doi.org/10.1038/s41586-024-07293-4.

Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. In-context symbolic regression: Leveraging large language models for function discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, page 589–606. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-srw.49. URL http://dx.doi.org/10.18653/v1/2024.acl-srw.49.

Yair Mundlak. On the pooling of time series and cross section data. *Econometrica*, 46(1):69–85, 1978. doi: 10.2307/1913646. URL https://www.jstor.org/stable/1913646.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, January 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-023-06924-6. URL https://www.nature.com/articles/s41586-023-06924-6.

David Antony Selby, Kai Spriestersbach, Yuichiro Iwashita, Dennis Bappert, Archana Warrier, Sumantrak Mukherjee, Muhammad Nabeel Asim, Koichi Kise, and Sebastian Josef Vollmer. Had enough of experts? elicitation and evaluation of bayesian priors from large language models. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024. URL https://openreview.net/forum?id=3iDxHRQfVy.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2404.18400.

Guangwei Si, Jessleen K. Kanwal, Yu Hu, Christopher J. Tabone, Jacob Baron, Matthew Berck, Gaetan Vignoud, and Aravinthan D. T. Samuel. Structured odorant response patterns across a complete olfactory receptor neuron population. *Neuron*, 101(5):950–962, 2019. doi: 10.1016/j.neuron.2018.12.030. URL https://doi.org/10.1016/j.neuron.2018.12.030.

Stan Development Team. CmdStanPy: A Python interface to CmdStan, 2026. URL https://github.com/stan-dev/cmdstanpy. Version 1.3.0.

Eric Thrane and Colm Talbot. An introduction to Bayesian inference in gravitational-wave astronomy: Parameter estimation, model selection, and hierarchical models. *Publications of the Astronomical Society of Australia*, 36:e010, 2019. doi: 10.1017/pasa.2019.2.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, 2017.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. Scimon: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 279–299. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.18. URL http://dx.doi.org/10.18653/v1/2024.acl-long.18.

Shijie Xia, Yuhan Sun, and Pengfei Liu. Sr-scientist: Scientific equation discovery with agentic ai, 2025. URL https://arxiv.org/abs/2510.11661.

Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh T. N. Nguyen, Lauren T. May, Geoffrey I. Webb, and Shirui Pan. Large language models for scientific synthesis, inference and explanation, 2023. URL https://arxiv.org/abs/2310.07984.