TIME-SERIES AS FEEDBACK: EVALUATING ADAPTIVE REASONING IN LLM AGENTS

Ryo Kuroki[1,2*] **Amin Mansouri**[1] **Philippe Schwaller**[1]
[1] EPFL, Lausanne, Switzerland [2] Asahi Kasei Corporation, Tokyo, Japan

ABSTRACT

Time-series interpretation and reasoning are essential for inferring the state of physical systems and remain a key challenge for autonomous scientific discovery. We introduce a benchmark to evaluate whether large language model (LLM) agents can perform such reasoning in adaptive experiment-planning settings where time-series observations serve as feedback and experimental conditions constitute agent actions that generate new trajectories. Using kinetic mechanism identification as a motivating testbed, we construct an agent–environment loop in which an agent iteratively proposes experiments, receives time-series data, and refines hypotheses over competing mechanisms while selecting new experimental conditions that best discriminate among them. We show that agents with likelihood-based (NLL) feedback consistently outperform adaptive and non-adaptive baselines, demonstrating effective hypothesis-aware adaptive experimental design. Agents operating directly on raw time-series feedback also outperform the same baselines, indicating a non-trivial capability to extract task-relevant information from noisy trajectories without hand-engineered analysis tools. However, raw-feedback performance remains below NLL-feedback performance, highlighting current limitations in direct time-series interpretation by LLM agents without structured signals. Overall, this work contributes both (i) a benchmark for interactive time-series reasoning in adaptive experimental settings, and (ii) an empirical study of LLM agents' strengths and limitations in hypothesis-driven scientific experimentation.

Track: Research

1 INTRODUCTION

Inferring the state of dynamical physical and chemical systems from time-series observations is a fundamental problem in scientific discovery. This class of problems arises in multiple scientific domains where competing dynamical models must be discriminated through adaptive experimentation. Examples include kinetic mechanism identification in chemistry (Blackmond, 2005; Burés, 2016a), adaptive Hamiltonian learning in quantum systems (Wang et al., 2017; Wiebe et al., 2014), gene regulatory network inference under perturbation experiments in systems biology (Gardner et al., 2003; Bonneau et al., 2006), neural circuit identification using controlled stimulation and time-resolved recordings (Paninski et al., 2007; Brunton et al., 2016), adaptive epidemiological model discrimination (Funk et al., 2015), and active system identification in robotics (Ljung, 1999). In these settings, inference rarely proceeds from a single experiment. Instead, researchers iteratively interpret time-resolved observations, maintain and refine hypotheses over competing system models, and design new experiments or interventions that most effectively discriminate among them. By repeating this closed-loop process – experiment, observe, update, redesign – uncertainty over the underlying dynamical mechanism is progressively reduced.

Recent work has begun to explore large language models (LLMs) as components of autonomous scientific discovery systems, including closed-loop experimentation, materials design, and scientific hypothesis generation (Song et al., 2025; Mitchener et al., 2025; Gottweis et al., 2025; M. Bran et al., 2024; Boiko et al., 2023; Yin et al., 2025; Darvish et al., 2025). Some existing agent benchmarks involve iterative interaction with simulated environments (Cissé et al., 2025; Song et al.,

*Correspondence to: kuroki.rf@asahi-kasei.co.jp

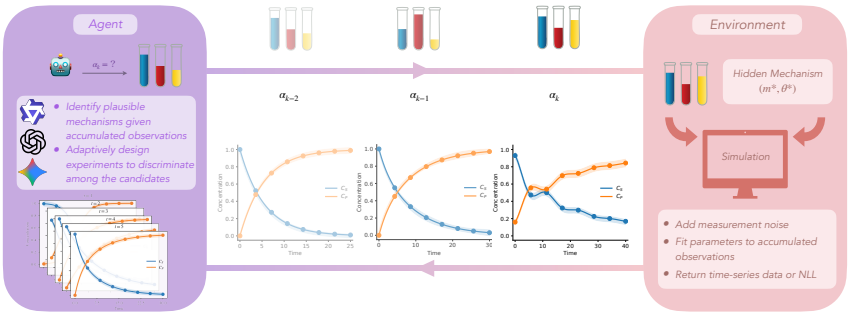

Figure 1: Overview of the iterative agent–environment loop for kinetic mechanism identification. At each iteration, the agent proposes an experimental condition α_k, the environment simulates noisy kinetic time-series data under a hidden mechanism, and feedback is returned either as raw time-series data or as NLL values computed by fitting candidate mechanisms.

2025; MacKnight et al., 2025). However, these benchmarks typically emphasize optimization objectives – such as maximizing target material properties or scalar performance metrics rather than hypothesis-driven model discrimination under dynamical, time-series feedback.

Despite the importance of time-series reasoning in scientific practice, there is currently no benchmark that explicitly evaluates whether LLM agents can (i) extract task-relevant information from raw time-series observations and (ii) reason over competing dynamical hypotheses to design adaptive, discriminative experiments. Moreover, traditional system identification pipelines rely heavily on domain-specific analytical tools (Blackmond, 2005; Burés, 2016a;b; Ringleb et al., 2025).

To address this gap, we introduce a benchmark for evaluating LLM agents in adaptive time-series–based system identification. We use kinetic mechanism identification (Burés & Larrosa, 2023) as a concrete and well-studied testbed for this general problem. In our setup, an agent interacts with an environment representing a dynamical system governed by a hidden ground-truth mechanism selected from a set of candidates. At each iteration, the agent proposes experimental conditions; the environment returns either raw time-series observations or likelihood-based feedback derived from fitting candidate models. The agent must iteratively refine hypotheses over competing mechanisms and select new experiments that best reduce uncertainty.

Our benchmark isolates two core capabilities: (i) extracting discriminative information directly from noisy time-series trajectories without task-specific analysis tools, and (ii) performing hypothesis-aware adaptive experiment planning. By contrasting raw time-series feedback with likelihood-based feedback and comparing against non-adaptive baselines, we provide a controlled evaluation of both the strengths and current limitations of LLM agents in interactive time-series reasoning.

2 METHODOLOGY

Benchmark Overview We formalize adaptive system identification as an interactive agent–environment loop in which *actions* correspond to experimental conditions and *observations* correspond to noisy time-series trajectories. The agent's goal is to iteratively design experiments that discriminate among competing candidate mechanisms, thereby reducing uncertainty over the underlying dynamical system. We instantiate this benchmark using kinetic mechanism identification as a concrete testbed. The benchmark consists of 100 independent reaction systems, each defined by a hidden ground-truth mechanism m^* and associated kinetic parameters θ^*. We consider a hypothesis class of 20 candidate mechanisms $\mathcal{M} = \{m_1, \ldots, m_{20}\}$ as described by Burés & Larrosa (2023), and construct 5 independent parameterizations per mechanism, yielding 100 total systems. All candidate kinetic mechanisms describe catalytic substrate-to-product transformations, in which a catalyst mediates and accelerates conversion from substrate to product. Details of the candidate mechanisms are provided in Appendix A.1.2.

Agent–Environment Interaction Loop For each chemical reaction system, we run an iterative agent–environment loop for kinetic mechanism identification for up to five rounds, as illustrated in Fig. 1. At iteration k, the agent proposes an experimental condition α_k, corresponding to initial concentrations of substrate, product, and catalyst within predefined ranges. Given α_k, the environment

simulates a kinetic experiment by generating a concentration trajectory $\mathbf{C}(t) = (C_S(t), C_P(t))$ under the ground-truth mechanism m^* and parameters θ^*. From this continuous trajectory, the environment returns a discrete set of noisy time-series observations $O(\alpha_k) = \{(t_j, \tilde{\mathbf{C}}(t_j; \alpha_k))\}_{j=1}^{6}$, where the six observation times $\{t_j\}$ are sampled at log-uniform intervals and Gaussian measurement noise is added to concentrations. $C_S(t)$ and $C_P(t)$ denote the substrate and product concentrations at time t, respectively. $\tilde{\mathbf{C}}(t_j; \alpha)$ is the measured concentration vector at time t_j.

The environment provides feedback to the agent through two interaction modes. i) Negative log-likelihood (NLL): In this mode, a fitting tool is applied to the accumulated kinetic time-series observations $\mathcal{O}_{1:k} = \{O(\alpha_1), \ldots, O(\alpha_k)\}$. Each candidate mechanism $m_i \in \mathcal{M}$ is fitted independently, producing a set of NLL values $\{\mathrm{NLL}(m_i)\}_{i=1}^{20}$, which are returned to the agent as structured feedback. ii) Raw time-series: In this mode, the environment returns the accumulated collection of time-series observations $\mathcal{O}_{1:k}$ directly, without any intermediate fitting or summarization. Thus, the agent must extract discriminative information from noisy trajectories alone and reason over plausible kinetic mechanisms to propose α_{k+1} for the next iteration.

The agent is implemented as an LLM guided by a fixed system prompt specifying high-level objectives: maintaining plausible hypotheses, selecting informative experiments, and reasoning about how candidate mechanisms would differ under proposed conditions. To avoid encoding domain-specific heuristics, no task-specific analysis rules are provided. Unless otherwise stated, we use Gemini 2.5 Pro as the agent. The prompt is provided in Appendix A.4.2.

As baselines, we consider three methods.

i) Random non-adaptive baseline This baseline samples each experiment independently from the same log-uniform condition domain over substrate, catalyst, and product concentrations.

ii) Maximin Latin hypercube sampling baseline This baseline provides a strong non-adaptive space-filling design. For each reaction system, we generate a fixed 5 point Latin hypercube design over 3D condition space and retain the design with the largest minimum pairwise distance in log-transformed space.

iii) Bayesian optimization non-adaptive baseline After each iteration, we fit a Gaussian process surrogate mapping log-transformed experimental conditions to the resulting mechanism entropy, i.e., the entropy of the candidate-mechanism distribution induced by per-mechanism NLL values (Appendix A.6), and select the next experiment by minimizing a lower confidence bound (LCB) acquisition function. The first two experiments are initialized by Latin hypercube sampling.

For evaluation, we report two primary metric families.

i) Environment-identification metrics. In both feedback modes, we compute fitting-based accuracy, defined as the probability that the true mechanism attains the minimum NLL among all candidates when fitted to the accumulated observations. We additionally report mechanism entropy over candidates, measuring uncertainty reduction across iterations.

ii) Agent diagnostic metrics (raw mode only). In raw feedback mode, we also compute Top-1 and Top-5 agent accuracy based on the mechanism rankings explicitly proposed by the agent. These diagnostics test whether improved environment-level identification is accompanied by more accurate hypothesis refinement by the agent itself.

iii) Proposal-space metrics. In both feedback modes, we report two proposal-space metrics: occupancy entropy and corner mass. Occupancy entropy is computed from the cumulative set of proposed conditions after mapping substrate, product, and catalyst concentrations into normalized log space and binning them into a fixed $3 \times 3 \times 2$ grid; we then compute the normalized Shannon entropy of the resulting occupancy distribution. Corner mass is the fraction of cumulative proposals that fall within 15% of a boundary along all three dimensions in the same normalized log-space representation. These metrics characterize how each proposer explores the experimental design space, distinguishing diffuse coverage from corner-seeking behavior.

3 EXPERIMENTS

Adaptive Experiment Planning vs. Non-Adaptive Baselines.

Fig. 2 shows that the agent achieves consistently higher fitting-based accuracy and lower hypothesis entropy than all baselines across iterations, indicating more effective uncertainty reduction over candidate mechanisms. Shaded regions denote 95% bootstrap confidence intervals over 100 reaction systems. These results suggest that the agent can reason over competing hypotheses to select experimental conditions within the predefined ranges that yield more discriminative time-series observations, thereby accelerating mechanism identification compared to the non-adaptive and adaptive designs. Note that the non-adaptive baselines also exhibits gradual accuracy improvement over iterations. This is expected as additional experiments – although chosen non-adaptively – accumulate more time-series observations, improving model discrimination through increased data volume. However, without adaptive selection, uncertainty reduction proceeds more slowly than with hypothesis-aware experiment planning.

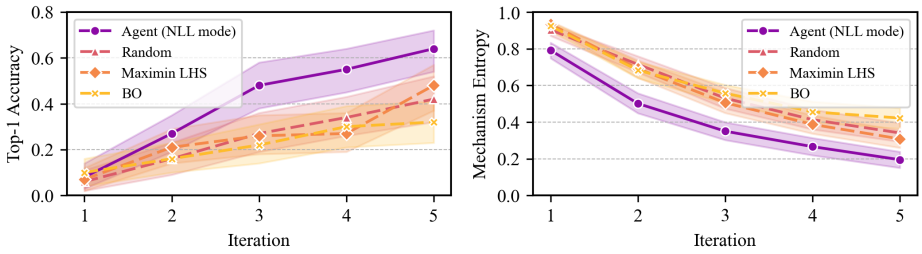

Figure 2: Agent versus non-adaptive baseline. Left: fitting-based accuracy over iterations. Right: entropy over candidate mechanisms.

Raw Time-Series Feedback vs. Likelihood-Based Feedback. Fig. 3 (top left) shows that the agent, operating in raw mode, consistently outperforms the strongest non-adaptive baseline, Maximin LHS, and exhibits a faster improvement in fitting-based accuracy over iterations. This indicates that the agent can extract task-relevant information for experimental planning directly from raw kinetic time-series observations. However, the agent in NLL mode attains higher accuracy throughout, reflecting the additional information provided by likelihood-based fitting. Further insight is provided by the top-k accuracy in Fig. 3 (upper right). Although the agent's top-1 mechanism predictions in raw mode remain inaccurate, the top-5 accuracy is substantially higher and is comparable to the fitting-based accuracy up to iteration 4. This suggests that, rather than identifying a single correct mechanism, the agent narrows the hypothesis space to a small set of plausible candidates using raw time-series data. Taken together, these results indicate that raw kinetic time-series provide sufficient feedback for hypothesis refinement and adaptive experimental planning, while the remaining gap to NLL feedback highlights the room for improvement in direct time-series interpretation with LLM agents.

Fig. 3 (bottom) further clarifies that this advantage is not explained by simple geometric heuristics alone. Compared with all baselines, the agent exhibits lower occupancy entropy, indicating that its proposals are not merely more space-filling across the design space. At the same time, its corner mass is higher than that of the baselines but remains below 0.2 throughout, showing that the agent does not simply concentrate on extreme corner conditions either. Taken together, this pattern is consistent with the agent performing hypothesis-aware adaptive experimental design, rather than relying on either diffuse space-filling exploration or a simple corner-seeking heuristic.

A Representative Raw Time-Series Feedback Case Study A representative raw-feedback trajectory (Appendix Table 1) provides a qualitative example of raw time-series reasoning by the agent. In this example, the ground-truth mechanism $M12$ belongs to the catalyst-deactivation class, and the agent correctly infers plausible deactivation from the initial trajectory before proposing a product-addition experiment that directly probes product-dependent deactivation. The main error arises later in finer-grained interpretation: after observing only a modest change under added product, the agent downweights $M12$ and shifts toward other deactivation pathways. Nevertheless, the experiment sequence it designs progressively improves environment-level identification of the true mechanism, with the fitting-based rank of $M12$ improving from 18 to 6 and then to 1. This is consistent with the broader quantitative pattern in raw mode. Top-1 agent accuracy remains limited, while fitting-based identification improves substantially.

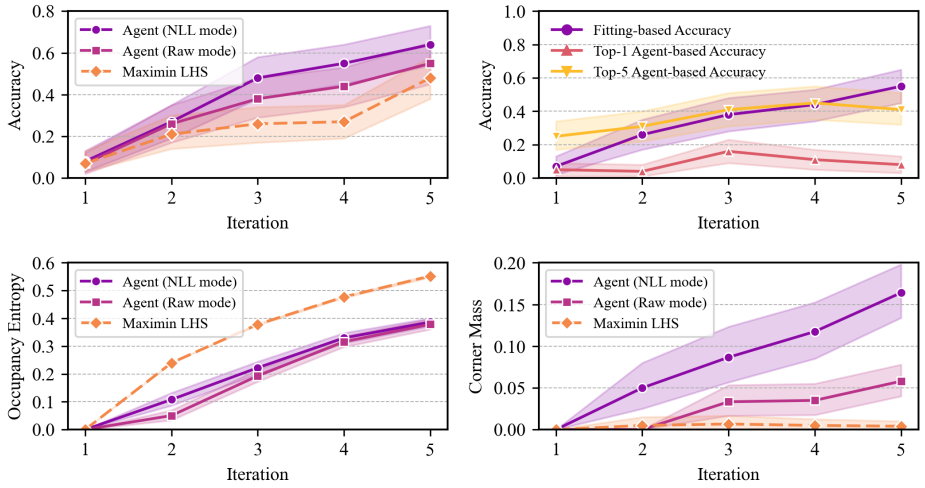

Figure 3: **Top:** Feedback-mode comparison and raw-mode diagnostics. Left: fitting-based accuracy for raw feedback, NLL feedback, and Maximin LHS baseline. Right: In raw mode, fitting-based accuracy versus agent Top-1/Top-5 mechanism accuracy. **Bottom** Proposal diagnostics. Left: Occupancy entropy for raw feedback, NLL feedback, and Maximin LHS baseline. Right: Corner mass for raw feedback, NLL feedback, and Maximin LHS baseline.

Model Comparison Across Feedback Modes. Consistent performance differences are observed across LLMs in both NLL and raw modes, with Gemini 2.5 Pro outperforming GPT-4o, which in turn outperforms Qwen3-32B (Fig. 4). The performance ranking remains stable across feedback settings. These results indicate that the benchmark meaningfully differentiates LLM capabilities in adaptive time-series reasoning and hypothesis-driven experimental planning.

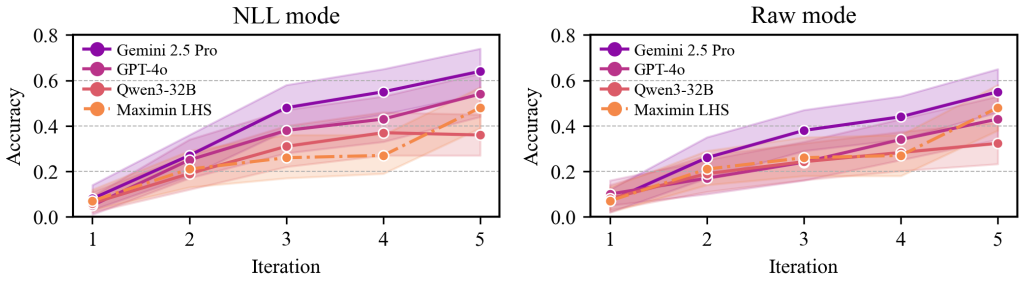

Figure 4: LLM performance across feedback modes. Left: Fitting-based accuracy in NLL mode. Right: Fitting-based accuracy in raw time-series mode. Shaded regions show 95% bootstrap confidence intervals over 100 systems.

4 Conclusion

We introduced a kinetic mechanism identification benchmark for evaluating LLM agents in interactive time-series reasoning and adaptive experiment planning. Agents with likelihood-based feedback substantially outperform adaptive and non-adaptive baselines, demonstrating effective hypothesis-aware experimental design. Agents operating directly on raw time-series feedback also outperform the same baselines, indicating meaningful capability to guide experiment selection. However, the persistent gap between raw and likelihood-based performance highlights current limitations in direct time-series interpretation without structured model-fitting signals. Finally, the benchmark differentiates performance across multiple LLMs, supporting its utility as a testbed for future work on adaptive time-series reasoning in scientific discovery.

REFERENCES

Donna G Blackmond. Reaction progress kinetic analysis: a powerful methodology for mechanistic studies of complex catalytic reactions. *Angewandte Chemie International Edition*, 44(28):4302–4320, 2005.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Richard Bonneau, David J Reiss, Paul Shannon, Marc T Facciotti, Leroy Hood, Nitin S Baliga, and Vesteinn Thorsson. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, 7(5):R36, 2006.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Jordi Burés. A simple graphical method to determine the order in catalyst. *Angewandte Chemie International Edition*, 55(6):2028–2031, 2016a.

Jordi Burés. Variable time normalization analysis: general graphical elucidation of reaction orders from concentration profiles. *Angewandte Chemie*, 128(52):16318–16321, 2016b.

Jordi Burés and Igor Larrosa. Organic reaction mechanism classification using machine learning. *Nature*, 613(7945):689–695, 2023.

Abdoulatif Cissé, Xenophon Evangelopoulos, Vladimir V. Gusev, and Andrew I. Cooper. Language-based bayesian optimization research assistant (bora). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI-2025, pp. 4967–4975. International Joint Conferences on Artificial Intelligence Organization, September 2025. doi: 10.24963/ijcai.2025/553. URL http://dx.doi.org/10.24963/ijcai.2025/553.

Kourosh Darvish, Marta Skreta, Yuchi Zhao, Naruki Yoshikawa, Sagnik Som, Miroslav Bogdanovic, Yang Cao, Han Hao, Haoping Xu, Alán Aspuru-Guzik, et al. Organa: A robotic assistant for automated chemistry experimentation and characterization. *Matter*, 8(2), 2025.

Sebastian Funk, Shweta Bansal, Chris T Bauch, Ken TD Eames, W John Edmunds, Alison P Galvani, and Petra Klepac. Nine challenges in incorporating the dynamics of behaviour in infectious diseases models. *Epidemics*, 10:21–25, 2015. doi: 10.1016/j.epidem.2014.09.005.

Timothy S Gardner, Diego di Bernardo, David Lorenz, and James J Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006. doi: 10.1093/bioinformatics/btl485.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2 edition, 1999.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Robert MacKnight, Jose Emilio Regio, Jeffrey G Ethier, Luke A Baldwin, and Gabe Gomes. Pretrained knowledge elevates large language models beyond traditional chemical reaction optimizers. *arXiv preprint arXiv:2509.00103*, 2025.

Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C Landsness, Daniel L Barabasi, Siddharth Narayanan, Nicky Evans, et al. Kosmos: An ai scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Liam Paninski, Jonathan W Pillow, and Jeremy Lewi. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in Brain Research*, 165:493–507, 2007.

Michael Ringleb, Alexander Eith, Stefan Zechel, Ulrich Schubert, Kevin Jablonka, and Jacob Schneidewind. Kinetically guided exploration of photocatalytic reactions by combining automation with in situ measurements, 2025.

Zhangde Song, Jieyu Lu, Yuanqi Du, Botao Yu, Thomas M. Pruyn, Yue Huang, Kehan Guo, Xiuzhe Luo, Yuanhao Qu, Yi Qu, Yinkai Wang, Haorui Wang, Jeff Guo, Jingru Gan, Parshin Shojaee, Di Luo, Andres M Bran, Gen Li, Qiyuan Zhao, Shao-Xiong Lennon Luo, Yuxuan Zhang, Xiang Zou, Wanru Zhao, Yifan F. Zhang, Wucheng Zhang, Shunan Zheng, Saiyang Zhang, Sartaaj Takrim Khan, Mahyar Rajabi-Kochi, Samantha Paradi-Maropakis, Tony Baltoiu, Fengyu Xie, Tianyang Chen, Kexin Huang, Weiliang Luo, Meijing Fang, Xin Yang, Lixue Cheng, Jiajun He, Soha Hassoun, Xiangliang Zhang, Wei Wang, Chandan K. Reddy, Chao Zhang, Zhiling Zheng, Mengdi Wang, Le Cong, Carla P. Gomes, Chang-Yu Hsieh, Aditya Nandy, Philippe Schwaller, Heather J. Kulik, Haojun Jia, Huan Sun, Seyed Mohamad Moosavi, and Chenru Duan. Evaluating large language models in scientific discovery, 2025. URL `https://arxiv.org/abs/2512.15567`.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Jianwei Wang, Stefano Paesani, Raffaele Santagati, Sebastian Knauer, Antonio A. Gentile, Nathan Wiebe, Maurangelo Petruzzella, Jeremy L. O'Brien, John G. Rarity, Anthony Laing, and Mark G. Thompson. Experimental quantum hamiltonian learning. *Nature Physics*, 13(6):551–555, 2017. doi: 10.1038/nphys4074.

Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David G Cory. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*, 112(19):190501, 2014.

TianZhixi Yin, Ruozhu Feng, Jie Bao, Peiyuan Gao, Yangang Liang, Job Heather, Alan Aspuru-Guzik, and Wei Wang. Learning advance: Robotics-llm guided hypotheses generation for the discovery of chemical knowledge. ChemRxiv, 2025. URL `https://chemrxiv.org/doi/full/10.26434/chemrxiv-2025-n1b4l`. Preprint.

A Appendix

A.1 Candidate mechanisms

A.1.1 Notation and modeling assumptions

We model all mechanisms as sets of elementary reactions and derive dynamics under standard mass-action kinetics. Reversibility is specified solely by the arrow type: <=> denotes a reversible elementary step and => denotes an irreversible elementary step. Observations include only substrate and product concentrations ($C_S(t)$ and $C_P(t)$); all catalyst-related intermediates are latent. Initial conditions: $C_S(0)$, $C_P(0)$, and $C_{\mathrm{cat}}(0)$ are chosen by the agent/baseline. All intermediate complexes (e.g., `catS`, `catP`, `cat2`, `cat*`, and all `inactive_*` species) are initialized to 0. Additional species that appear only in certain mechanisms are initialized to 1: $C_X(0) = 1$, $C_{\mathrm{inhibitor}}(0) = 1$, and $C_L(0) = 1$.

A.1.2 CANDIDATE MECHANISMS

These mechanisms are grouped into four broad classes: (a) a core mechanism without side reactions, (b) mechanisms containing bicatalytic steps, (c) mechanisms involving catalyst activation, and (d) mechanisms with catalyst deactivation pathways.

Category mapping We use the following mapping between candidate mechanisms and the four mechanism classes: (a) `M1`; (b) `M2-M5`; (c) `M6-M8`; and (d) `M9-M20`.

Mechanism definitions Each mechanism is specified by a sequence of elementary reactions in a compact reaction-string form. The ODEs are derived under mass-action kinetics from the reaction graph and arrow directions alone (reversible <=> vs irreversible =>). We use | to separate elementary steps within a mechanism.

```
M1:  S + cat <=> catS | catS <=> P + cat
M2:  S + cat <=> catS | catS <=> P + cat | 2cat <=> cat2
M3:  S + cat2 <=> cat2S | cat2S <=> P + cat2 | 2cat <=> cat2
M4:  X + catS <=> S + cat | X + catS <=> P + cat
M5:  S + cat <=> catS | catS + cat <=> catP + cat | catP <=> P + cat
M6:  cat => cat* | S + cat* <=> cat*S | cat*S <=> P + cat*
M7:  S + cat <=> catS | S + catS <=> catS2 | catS2 <=> P + catS
M8:  S + cat* <=> cat*S | cat*S <=> P + cat* | cat <=> cat* + L
M9:  S + cat <=> catS | catS <=> P + cat | cat => inactive_cat
M10: S + cat <=> catS | catS <=> P + cat | inhibitor + cat => inactive_catI
M11: S + cat <=> catS | catS <=> P + cat | S + cat => inactive_catS
M12: S + cat <=> catS | catS <=> P + cat | P + cat => inactive_catP
M13: S + cat <=> catS | catS <=> P + cat | 2cat => inactive_cat2
M14: S + cat <=> catS | catS <=> P + cat | catS => inactive_catS
M15: S + cat <=> catS | catS <=> P + cat | inhibitor + catS => inactive_catSI
M16: S + cat <=> catS | catS <=> P + cat | S + catS => inactive_catS2
M17: S + cat <=> catS | catS <=> P + cat | P + catS => inactive_catSP
M18: S + cat <=> catS | catS <=> P + cat | 2catS => inactive_cat2S2
M19: S + cat <=> catS | catS <=> P + cat | cat + catS => inactive_cat2S
M20: S + cat <=> catS | catS <=> P + cat | cat => inactive_cat | catS => inactive_catS
```

Mechanism definition JSON provided to agents

```
{
  "M1": "S+cat<=>catS|catS<=>P+cat",
  "M2": "S+cat<=>catS|catS<=>P+cat|2cat<=>cat2",
  "M3": "S+cat2<=>cat2S|cat2S<=>P+cat2|2cat<=>cat2",
  "M4": "X+catS<=>S+cat|X+catS<=>P+cat",
  "M5": "S+cat<=>catS|catS+cat<=>catP+cat|catP<=>P+cat",
  "M6": "cat=>cat*|S+cat*<=>cat*S|cat*S<=>P+cat*",
  "M7": "S+cat<=>catS|S+catS<=>catS2|catS2<=>P+catS",
  "M8": "S+cat*<=>cat*S|cat*S<=>P+cat*|cat<=>cat*+L",
  "M9": "S+cat<=>catS|catS<=>P+cat|cat=>inactive_cat",
  "M10": "S+cat<=>catS|catS<=>P+cat|inhibitor+cat=>inactive_catI",
  "M11": "S+cat<=>catS|catS<=>P+cat|S+cat=>inactive_catS",
  "M12": "S+cat<=>catS|catS<=>P+cat|P+cat=>inactive_catP",
  "M13": "S+cat<=>catS|catS<=>P+cat|2cat=>inactive_cat2",
  "M14": "S+cat<=>catS|catS<=>P+cat|catS=>inactive_catS",
  "M15": "S+cat<=>catS|catS<=>P+cat|inhibitor+catS=>inactive_catSI",
  "M16": "S+cat<=>catS|catS<=>P+cat|S+catS=>inactive_catS2",
  "M17": "S+cat<=>catS|catS<=>P+cat|P+catS=>inactive_catSP",
  "M18": "S+cat<=>catS|catS<=>P+cat|2catS=>inactive_cat2S2",
  "M19": "S+cat<=>catS|catS<=>P+cat|cat+catS=>inactive_cat2S",
  "M20": "S+cat<=>catS|catS<=>P+cat|cat=>inactive_cat|catS=>inactive_catS"
}
```

In addition to `S`, `P`, and `cat`, the candidate mechanism set introduces the following extra species that are present at initialization: `X` (a different catalytic species, used in `M4`), `L` (a ligand, used in `M8`), and

`inhibitor` (an inhibitor species, used in `M10` and `M15`). All other species in the reaction strings are mechanism-specific intermediates (including activated/deactivated forms and complexes) that are initialized to 0 and are defined implicitly by the reaction topology above.

A.1.3 Parameter priors and constraints

All kinetic constants are sampled independently from a log-uniform prior, $\theta \sim \mathrm{LogUniform}(10^{-2}, 10^{2})$. After sampling, we filter parameter sets using mechanism-specific dynamical conditions to ensure that the defining catalyst-related species for each mechanism attain non-negligible concentrations under a reference simulation.

We define conversion as fractional substrate consumption, $x(t) = \big(C_S(0) - C_S(t)\big)/C_S(0)$. For mechanisms with multiple catalyst-containing species, we define the total catalyst pool as $C_{\mathrm{tot}}(t) = \sum_{s\in\mathcal{S}_{\mathrm{cat}}} C_s(t)$, where $\mathcal{S}_{\mathrm{cat}}$ is the set of catalyst-related species for that mechanism, and define a catalyst-species fraction for any numerator set $\mathcal{N}$ as

$$r_{\mathcal{N}}(t) \;=\; \frac{\sum_{s\in\mathcal{N}} C_s(t)}{C_{\mathrm{tot}}(t)}\,.$$

Reference simulation conditions for filtering Unless otherwise stated, filtering conditions are evaluated under a reference initial condition with $C_S(0) = 1$ and $C_P(0) = 0$. Catalyst "mol%" is defined as the ratio $C_{\mathrm{cat}}(0)/C_S(0)$. Therefore, 5 mol% corresponds to $C_{\mathrm{cat}}(0) = 0.05$ in the reference run, and a 3–7 mol% sweep corresponds to $C_{\mathrm{cat}}(0) \in [0.03, 0.07]$ with $C_S(0) = 1$.

Mechanism-specific filters The mechanism-specific filtering criteria below are adopted from Burés & Larrosa (2023).

- `M1`, `M4`, `M5`: no additional filtering beyond the prior.
- `M2` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{cat2}\}$): in a reference run with 5 mol% catalyst, require a significant dimeric fraction during the early-to-mid reaction, i.e., $\max_{x\in[0.2,0.5]} r_{\{\mathtt{cat2}\}} \geq 0.1$.
- `M3` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{cat2}, \mathtt{cat2S}\}$): in a reference run with 5 mol% catalyst, require that free `cat` is not vanishingly small at the start of the reaction, i.e., $\min_{x\in[0.0,0.1]} r_{\{\mathtt{cat}\}} \geq 0.05$.
- `M6` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{cat*}, \mathtt{cat*S}\}$): for at least one catalyst loading in 3–7 mol%, require that activated/complexed catalyst has an intermediate fraction at 20% conversion, i.e., $r_{\{\mathtt{cat*},\mathtt{cat*S}\}} \in [0.1, 0.9]$ at $x = 0.2$.
- `M7` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{catS2}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{catS},\mathtt{catS2}\}} \in [0.1, 0.8]$ at $x = 0.2$.
- `M8` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{cat*}, \mathtt{cat*S}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{cat*},\mathtt{cat*S}\}} \in [0.1, 0.9]$ at $x = 0.5$.
- `M9` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_cat}\}$): for at least one catalyst loading in 3–7 mol%, require an intermediate active-catalyst fraction, $r_{\{\mathtt{cat},\mathtt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M10` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_catI}\}$): for at least one catalyst loading in 3–7 mol%, require appreciable inhibited catalyst, $r_{\{\mathtt{inactive\_catI}\}} \geq 0.1$ at $x = 0.5$.
- `M11` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_catS}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{cat},\mathtt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M12` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_catP}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{cat},\mathtt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M13` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_cat2}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{cat},\mathtt{catS}\}} \in [0.5, 0.8]$ at $x = 0.5$.
- `M14` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_catS}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{cat},\mathtt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M15` ($\mathcal{S}_{\mathrm{cat}} = \{\mathtt{cat}, \mathtt{catS}, \mathtt{inactive\_catSI}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\mathtt{inactive\_catSI}\}} \geq 0.1$ at $x = 0.5$.

- `M16` ($\mathcal{S}_{\text{cat}} = \{\texttt{cat}, \texttt{catS}, \texttt{inactive\_catS2}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\texttt{cat},\texttt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M17` ($\mathcal{S}_{\text{cat}} = \{\texttt{cat}, \texttt{catS}, \texttt{inactive\_catSP}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\texttt{cat},\texttt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$.
- `M18` ($\mathcal{S}_{\text{cat}} = \{\texttt{cat}, \texttt{catS}, \texttt{inactive\_cat2S2}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\texttt{cat},\texttt{catS}\}} \in [0.5, 0.8]$ at $x = 0.5$.
- `M19` ($\mathcal{S}_{\text{cat}} = \{\texttt{cat}, \texttt{catS}, \texttt{inactive\_cat2S}\}$): for at least one catalyst loading in 3–7 mol%, require $r_{\{\texttt{cat},\texttt{catS}\}} \in [0.5, 0.8]$ at $x = 0.5$.
- `M20` ($\mathcal{S}_{\text{cat}} = \{\texttt{cat}, \texttt{catS}, \texttt{inactive\_cat}, \texttt{inactive\_catS}\}$): require (i) an intermediate active-catalyst fraction, $r_{\{\texttt{cat},\texttt{catS}\}} \in [0.5, 0.9]$ at $x = 0.5$, and (ii) appreciable deactivated fractions, $r_{\{\texttt{inactive\_cat}\}} > 0.05$ and $r_{\{\texttt{inactive\_catS}\}} > 0.05$ at $x = 0.5$.

A.2 Simulator and data generation

A.2.1 Experimental condition space

The experimental condition is $\alpha = \big(C_S(0),\, C_{\text{cat}}(0),\, C_P(0)\big)$, i.e., the initial concentrations of substrate, catalyst, and product. Allowed ranges are the same for the agent and the random baseline: $C_S(0) \in [0.1, 5.0]$, $C_{\text{cat}}(0) \in [5 \times 10^{-4}, 0.2]$, and $C_P(0) \in [0, 0.5]$. The non-adaptive random baseline samples conditions from this domain using log-uniform sampling: $C_S(0)$ and $C_{\text{cat}}(0)$ are sampled log-uniformly over their strictly-positive ranges; for $C_P(0)$, we sample log-uniformly over $[\varepsilon, 0.5 + \varepsilon]$ with $\varepsilon = 10^{-9}$ and then set $C_P(0) \leftarrow C_P(0) - \varepsilon$ to map samples back to $[0, 0.5]$.

A.2.2 ODE simulation and observation times

We first simulate a dense trajectory up to a fixed maximum time `max_t`. Let $P_\infty = P(\texttt{max\_t})$ denote the final product concentration at the end of this dense simulation, and define the effective horizon as the earliest time $t_{\max} \in [0, \texttt{max\_t}]$ such that $P(t_{\max}) \geq 0.95\, P_\infty$. We then sample 6 sparse observation points log-uniformly on $[t_{\text{start}}, t_{\max}]$ with $t_{\text{start}} = 10^{-6}\, t_{\max}$ (and using machine epsilon if $t_{\text{start}} = 0$). Kinetic time-series data are generated using COPASI (Hoops et al., 2006) via `basico.run_time_course` with `num_steps=5000`, `output_event=True`, `max_steps=1,000,000`, absolute tolerance 10^{-12}, and relative tolerance 10^{-8}; solver exceptions are treated as failed simulations. When computing NLL values by fitting candidate mechanisms, we integrate ODEs with SciPy (Virtanen et al., 2020) `solve_ivp` using `method=LSODA` with `rtol=1e-4` and `atol=1e-8` at the experimental time points.

A.2.3 Observation model

Noisy observations are obtained by adding Gaussian noise to the concentrations at each observation time. The observation noise level $\sigma = 0.05$ is defined with respect to the normalized concentration scale used in the simulation and is applied consistently across all experiments. Specifically, the observed concentrations are given by

$$\tilde{\mathbf{C}}(\tau_j; \alpha) = \mathbf{C}(\tau_j; \alpha) + \boldsymbol{\varepsilon}_j, \quad \boldsymbol{\varepsilon}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}).$$

A.3 Likelihood-Based Evaluation and Parameter Fitting

A.3.1 Gaussian Observation Model and NLL

We define a Gaussian observation model with independent noise across data points. Let SSE denote the summed squared error between simulated and observed concentrations across the accumulated collection of kinetic time-series observations (i.e., across all iterations included in $\mathcal{O}_{1:k}$, all observed species, and all observation time points), and let n be the total number of scalar data points. The log-likelihood used in our implementation is

$$\log p(\mathcal{O}_{1:k} \mid \theta, \sigma) = -\tfrac{1}{2} n \log(2\pi\sigma^2) - \tfrac{1}{2}\, \text{SSE}/\sigma^2.$$

We report negative log-likelihood (NLL) as $\text{NLL} = -\log p(\mathcal{O}_{1:k} \mid \theta, \sigma)$.

During fitting, σ is treated as a free parameter and is optimized jointly with kinetic constants. We parameterize θ and σ in log-space with uniform box constraints: $\log_{10}\theta \in [-2, 2]$ and $\log_{10}\sigma \in [-2.5, -1.5]$.

A.3.2 PARAMETER ESTIMATION VIA CMA-ES

Parameter estimation is performed using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Hansen (2016), which is well suited for nonlinear, nonconvex optimization problems arising in kinetic modeling. All fitting procedures use identical optimization settings across mechanisms. The objective minimized by CMA-ES is the NLL defined above. We use a log-parameterization with box constraints for $\log_{10}\theta$ and $\log_{10}\sigma$ as specified above and initialize CMA-ES with $\sigma_0 = 1.0$ in the internal parameter space. CMA-ES options used in our implementation are:

- `ftarget=-10000` (a sufficiently small target so optimization effectively runs as unconstrained minimization),
- `tolstagnation=20`,
- `popsize_factor=7`,
- `timeout=60` seconds,
- `verb_disp=0`.

Optimizer failures (including ODE integration failures during objective evaluation) are treated as fitting failures.

A.4 AGENTS

A.4.1 INTERACTION PROTOCOL

In raw mode, after each proposed experiment, the environment provides the accumulated raw time-course dataset from all runs so far as a JSON object mapping run identifiers (`"run_1"`, `"run_2"`, ...) to per-run records. Each run record contains `"initial_conditions"` with keys `"S"`, `"cat"`, `"P"`, `"time_data"` (a length 6 list of time points), and `"substrate_data"` / `"product_data"` (length 6 lists of the corresponding concentrations). In NLL mode, after each proposed experiment, the environment provides the proposed initial conditions and a JSON object of per-mechanism NLL values for the 20 candidates indexed by `"M1"`–`"M20"` (lower NLL indicates a better fit to the accumulated observations); if fitting fails for all mechanisms for a given condition, NLL values are returned as `null` for that iteration (see Appendix A.7). In both modes, the agent proposes the next experimental condition in a `<suggestion>...</suggestion>` JSON object with keys `"initial_concentration_of_substrate"`, `"initial_concentration_of_catalyst"`, and `"initial_concentration_of_product"` (within the allowed ranges), and outputs a `<candidate_mechanisms>` field containing mechanism names (`M1`–`M20`) for diagnostics.

A.4.2 PROMPTS AND LLM CONFIGURATION

We run agents via OpenRouter (`https://openrouter.ai/api/v1`) using the following model identifiers: `google/gemini-2.5-pro`, `openai/gpt-4o`, and `qwen/qwen3-32b`. We use greedy decoding with temperature 0, do not set top-p explicitly and use provider's default values, set the maximum completion length to 20,000 tokens, use a request timeout of 120 seconds, and disable streaming.

NLL mode system prompt

```
You are an expert in chemical kinetics and experimental design.
Your mission is to hypothesize plausible mechanisms, and design a single, maximally
informative experiment to help distinguish between the plausible candidate mechanisms.
You may propose 5 new experiments.

Execute the following same steps iteratively.
At the beginning of each turn, before executing any step, check how many iterations are
done based on the conversation history.
```

```
1. **Hypothesize about plausible mechanisms**
    *   You are provided with the fitting results unless this is the first turn. Based on
    the fitting results, identify all mechanisms that still have non-negligible support.
    *   Even if some mechanisms appear less likely, do NOT prematurely commit to a single
    leading hypothesis.
    *   Treat hypothesis diversity as important information to preserve.
2. **Hypothesize optimal experiment for discrimination**
    *   Your goal is NOT to maximize the distinction between the single most likely pair
    of mechanisms.
    *   Instead, your goal is to reduce uncertainty over the entire set of remaining
    plausible mechanisms.
    *   Prefer experiments that are informative for many mechanisms simultaneously.
    *   Avoid experiments that would only distinguish the top 1-2 candidates while leaving
    the others indistinguishable.
    *   Consider how the proposed condition helps differentiate multiple mechanisms at
    once and why it avoids overly myopic or extreme designs.
    *   Reflect on the history of conditions: avoid repeating similar regimes, but also
    avoid extreme conditions that collapse mechanistic differences (e.g., saturation,
    depletion).
3.  **Propose Experiment**:
    *   The experimental conditions that you can propose are initial concentration of
    substrate, catalyst and product.
    *   Conclude with <suggestion> ... </suggestion> and <<DONE>>.

After you propose a new experiment, you would obtain the fitting result of all runs
including new experimental data based on your suggestion.

For each turn, you must follow this format:
<iteration>
Specify the iteration number. Start from 1 and increment by 1 iteration.
</iteration>
<reflection>
- Your observations and reasoning about the previous turn's results.
- A clear statement on whether your aim for the turn was achieved.
</reflection>
<hypothesis>
Your detailed reasoning for the proposed experiment. This should include your scientific
hypothesis, how the experiment will distinguish between mechanisms, and why the design is
effective.
</hypothesis>
<candidate_mechanisms>
The current plausible candidate mechanisms under consideration. Provide them
comma-separated as a list (e.g. M1, M2).
</candidate_mechanisms>
<suggestion>
{
    "initial_concentration_of_substrate": <float 0.1 - 5.0>,
    "initial_concentration_of_catalyst": <float 5e-4 - 0.20>,
    "initial_concentration_of_product": <float 0.0 - 0.5>,
}
</suggestion>
<<DONE>>

# 20 Candidate Mechanisms
[MECHANISM-DEFINITION JSON OMITTED FOR BREVITY.]
# Data Structure
The following describes the structure of the data. This is not provided to you directly,
but you can access the fitting results of all runs through the conversation history.
At the first run, there is only "run_1". As you propose new experiments and obtain new
data, additional runs (e.g., "run_2", "run_3") will be added.

```python
{
  "run_1": {
    "initial_concentration_of_catalyst": float, # Example: 0.01
```
```

```
    "time_data": list[float], # List of time points
    "product_data": list[float], # List of product concentrations at each time point
    "substrate_data": list[float] # List of substrate concentrations at each time point
  },
  "run_2": {
    # ... same structure as run_1
  },
  # ... additional runs
}
```

Fitting Results
Everytime you propose a new experiment, you will receive the fitting results for all runs including the new experimental data based on your suggestion.
The fitting results are provided as negative log-likelihood values for each candidate mechanism.
Here are example fitting results from the previous runs:
```json
{
    "M1": -66.4739509769402,
    "M2": -66.18465035678079,
    "M3": -118.16423286937294,
    "M4": -131.86593439082938,
    "M5": 35.36495497243471,
    "M6": -78.6224862051261,
    "M7": -77.53127516407199,
    "M8": 69.72349746655837,
    "M9": -93.85551875223946,
    "M10": -106.0115552520685,
    "M11": -111.90660321199617,
    "M12": -117.2823521990672,
    "M13": -127.68391782844185,
    "M14": -66.05561653283398,
    "M15": -135.0936662137469,
    "M16": -71.82102553018892,
    "M17": -66.07488048869106,
    "M18": -63.6419421066368,
    "M19": -88.57669991706247,
    "M20": -48.13563459951533
}
```

The omitted mechanism-definition JSON is provided in Appendix A.1.2.

Raw mode system prompt

```
You are an expert in chemical kinetics and experimental design.
Your mission is to hypothesize plausible mechanisms, and design a single, maximally
informative experiment to help distinguish between the plausible candidate mechanisms.
You may propose 5 new experiments.

Execute the following same steps iteratively.
At the beginning of each turn, before executing any step, check how many iterations are
done based on the conversation history.

1. **Hypothesize about plausible mechanisms**
    *   You are provided with raw kinetic data unless this is the first turn. Based on the
    data, identify all mechanisms that still have non-negligible support based on your
    chemical kinetics knowledge.
    *   Even if some mechanisms appear less likely, do NOT prematurely commit to a single
    leading hypothesis.
    *   Treat hypothesis diversity as important information to preserve.
2. **Hypothesize optimal experiment for discrimination**
    *   Your goal is NOT to maximize the distinction between the single most likely pair
    of mechanisms.
```

```
    *   Instead, your goal is to reduce uncertainty over the entire set of remaining
    plausible mechanisms.
    *   Prefer experiments that are informative for many mechanisms simultaneously.
    *   Avoid experiments that would only distinguish the top 1-2 candidates while leaving
    the others indistinguishable.
    *   Consider how the proposed condition helps differentiate multiple mechanisms at
    once and why it avoids overly myopic or extreme designs.
    *   Reflect on the history of conditions: avoid repeating similar regimes, but also
    avoid extreme conditions that collapse mechanistic differences (e.g., saturation,
    depletion).
3.  **Propose Experiment**:
    *   The experimental conditions that you can propose are initial concentration of
    substrate, catalyst and product.
    *   Conclude with <suggestion> ... </suggestion> and <<DONE>>.

After you propose a new experiment, you would obtain the raw kinetic data of all runs
including new experimental data based on your suggestion.

For each turn, you must follow this format:
<iteration>
Specify the iteration number. Start from 1 and increment by 1 iteration.
</iteration>
<reflection>
- Your observations and reasoning about the previous turn's results.
- A clear statement on whether your aim for the turn was achieved.
</reflection>
<hypothesis_plausible_mechanisms>
- Your detailed reasoning for the plausible mechanisms. This should include your
scientific hypothesis on why these mechanisms are plausible given the data.
</hypothesis_plausible_mechanisms>
<hypothesis_experiment>
- Your detailed reasoning for the proposed experiment. This should include your scientific
hypothesis, how the experiment will distinguish between mechanisms, and why the design is
effective.
</hypothesis_experiment>
<candidate_mechanisms>
- List the plausible candidate mechanisms, sorted in descending order of plausibility
(most likely first). Provide them as a JSON list of strings (e.g. ["M1", "M5", "M2"]).
</candidate_mechanisms>
<suggestion>
{
    "initial_concentration_of_substrate": <float 0.1 - 5.0>,
    "initial_concentration_of_catalyst": <float 5e-4 - 0.20>,
    "initial_concentration_of_product": <float 0.0 - 0.5>,
}
</suggestion>
<<DONE>>

# 20 Candidate Mechanisms
[MECHANISM-DEFINITION JSON OMITTED FOR BREVITY.]
# Data Structure
The following describes the structure of the data. This kind of data is provided after you
propose a new experiment.
At the first run, there is only "run_1". As you propose new experiments and obtain new
data, additional runs (e.g., "run_2", "run_3") will be added.

```python
{
  "run_1": {
    "initial_concentration_of_catalyst": float, # Example: 0.01
    "time_data": list[float], # List of time points
    "product_data": list[float], # List of product concentrations at each time point
    "substrate_data": list[float] # List of substrate concentrations at each time point
  },
  "run_2": {
    # ... same structure as run_1
```
```

```
  },
  # ... additional runs
}
```

The omitted mechanism-definition JSON is provided in Appendix A.1.2.

A.5 Representative Raw-Feedback Case Study

This table provides the detailed step-by-step summary referenced in the main text.

Step	Observed feature from raw trajectories	Mechanistic implication / hypothesis update	Next experiment proposed	Effect on true-mechanism identification
$1 \to 2$	In the initial run (S=1.0, cat=0.02, P=0), the apparent rate decreases more strongly than expected from substrate depletion alone.	The agent infers plausible catalyst deactivation and downweights non-deactivation mechanisms.	Add initial product while keeping S and cat fixed: S=1.0, cat=0.02, P=0.2.	After the second run, the fitting-based rank of the true mechanism ($M12$) improves from 18 to 6.
$2 \to 3$	Adding product changes the trajectory only modestly relative to run 1, without an obvious increase in deactivation severity.	The agent interprets this as evidence against product-induced deactivation and shifts attention to other deactivation pathways.	Increase catalyst loading strongly: S=3.0, cat=0.1, P=0.	After the third run, the fitting-based rank of $M12$ improves to 1.
$3 \to 4$	At higher catalyst loading, the agent observes lower apparent catalyst efficiency and interprets this as stronger catalyst loss.	The agent updates toward bimolecular deactivation hypotheses ($M13/M18/M19$) over first-order deactivation pathways.	Lower substrate strongly while restoring moderate catalyst: S=0.1, cat=0.02, P=0.	After the fourth run, the true mechanism remains rank 1.
$4 \to 5$	At low substrate, the reaction stalls at very low conversion; the agent interprets this as severe deactivation when free catalyst is abundant.	The agent favors free-catalyst bimolecular deactivation ($M13$) and moves further away from the true product-dependent mechanism $M12$.	Use saturating substrate to suppress free catalyst: S=5.0, cat=0.02, P=0.	After five runs, the true mechanism still remains rank 1 under fitting-based evaluation.

Table 1: Representative raw-feedback case study for a system whose ground-truth mechanism is $M12$. The agent correctly identifies catalyst deactivation as a plausible mechanistic class and proposes targeted follow-up experiments probing product dependence, catalyst-order effects, and the balance between free and substrate-bound catalyst. Although the agent's own mechanism ranking later moves away from $M12$, the sequence illustrates useful feature extraction from raw trajectories and hypothesis-aware experiment planning under noisy feedback.

A.6 Metrics and statistics

Mechanism entropy is computed from per-mechanism negative log-likelihood values $\{\mathrm{NLL}_i\}_{i=1}^{|\mathcal{M}|}$ (lower is better) by converting them to a normalized distribution via a softmax over $-\mathrm{NLL}$,

$$p_i \;=\; \frac{\exp(-\mathrm{NLL}_i)}{\sum_{j=1}^{|\mathcal{M}|}\exp(-\mathrm{NLL}_j)}\,,$$

and then computing the (natural-log) Shannon entropy

$$H \;=\; -\sum_{i=1}^{|\mathcal{M}|} p_i \log p_i\,.$$

We report 95% bootstrap confidence intervals over the 100 systems.

A.7 FAILURES AND EDGE CASES

If the fitting procedure fails to return valid NLL values for all 20 mechanisms for a proposed condition α (e.g., due to numerical instability or divergence in ODE simulation), then in NLL mode we return `null` NLL values to the agent for that iteration. For all subsequent iterations, we exclude the corresponding observation $O(\alpha)$ from the accumulated collection of kinetic time-series observations $\mathcal{O}_{1:k}$ used for fitting and for computing evaluation metrics (accuracy and entropy). In the agent loop, we additionally provide an environment message indicating that the previous condition diverged and requesting a different proposal. For the random baseline, no such message is required; we simply proceed with the next sampled condition under the same exclusion rule.