Do LLM Judges Evaluate What We Ask Them to Measure? A Controlled Benchmark for Complexity and Chaos in Scientific Time Series

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) are increasingly used as judges, yet it remains unclear whether they evaluate the requested property or rely on correlated features of the data. We introduce a controlled benchmark for LLM judges of scientific time series in which complexity and chaos can be varied independently with known ground truth. Across ranking, outlier detection, clustering, and pairwise evaluation tasks, we find striking differences among three leading LLMs. Claude achieved near-perfect performance, whereas GPT and Gemini showed strong task-dependent failures, particularly in distinguishing complexity from chaos. Analysis of model reasoning shows that successful judgments rely on features directly related to the requested dynamical property, while failures often arise from plausible but incorrect proxies. These results show that extracting meaningful information from scientific data is not sufficient for reliable LLM-based judgment: models must identify the features corresponding to the scientific quantity they are asked to evaluate.

1 Introduction

Large language models (LLMs) are increasingly being used not only to analyze and summarize information, but also as judges that evaluate and rank data, model outputs, and scientific results[1–7]. Extending this role to scientific time series is crucial, as they provide one of the primary means to analyze dynamical behavior in complex systems. Large experimental and observational datasets in areas ranging from physiology and neuroscience to climate, astronomy, and other complex systems often contain signals whose dynamical properties, such as regularity, complexity, or chaotic behavior, must be reliably quantified[8–10]. Here, we use a transformation of the Lorenz system[11] that we recently derived, with details in Appendix A.1, to determine whether LLM judges can independently and correctly evaluate complexity and chaos in scientific time series, two distinct dynamical properties that are often conflated[12, 13]. A key advantage of this system is that geometrical complexity and chaotic instability (Lyapunov spectrum) can be varied independently, thereby providing controlled ground truth to test whether LLMs measure the intended dynamical property or rely on correlated proxies. We evaluate multiple LLMs to examine construct validity, criteria consistency, inter-judge agreement, and reasoning validity[14–19]. This controlled framework allows us to assess both the accuracy of LLM judgments and the extent to which they reflect the intended property rather than correlated or incidental features of the data.

2 Controlled Complexity–Chaos Benchmark with LLM Judges

In the N-Lorenz system (see A.1), the parameter N controls the number of unstable fixed points around which the trajectory evolves (and thus the number of attractor lobes). Increasing N increases

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

the geometrical and temporal complexity of the resulting time series while preserving the Lyapunov spectrum. Conversely, varying ρ at fixed N changes the degree of chaos while preserving geometrical complexity. Thus, complexity can be systematically varied while chaotic instability remains unchanged, and vice versa, providing a controlled ground truth for independently evaluating the two properties.

Motivated by recent studies that applied LLMs directly to raw time-series data for tasks such as forecasting, reasoning, and explanation [20–22, 1], we evaluated three leading LLMs (GPT-5.6-sol, Claude-Opus-5, and Gemini-3.7-flash) as judges of complexity and chaos directly from raw time-series data. Each model was tested directly, so that its judgments could be attributed to the model's intrinsic reasoning rather than to supporting numerical analysis. Across randomized realizations and initial conditions, we tested the judges on complexity ranking, complexity–chaos discrimination, and identification of signals that differ from controlled reference groups. Model responses were evaluated against the known ground truth of the N-Lorenz system and compared across models, trials, and evaluation conditions.

3 Experimental Tasks and Evaluation Metrics

We evaluated four classes of tasks of different difficulty levels, described briefly below, but with complete dataset compositions and prompts provided in the Appendix A.2, A.3.

Task 1: Ranking. To test whether LLMs can order time series by a specific dynamical property, 20 time series were ranked by either *a) complexity*, using time series with 2–21 lobes, or *b) chaotic behavior*, using time series with the same number of lobes but different degrees of chaos.

Task 2: Outlier detection. To test whether LLMs can identify differences relative to a common reference, among 20 time series generated from different initial conditions, the models identified *a) a single complexity outlier* among 19 time series with the same number of lobes, or *b) multiple complexity outliers* relative to a majority group.

Task 3: Complexity-based clustering. To test whether LLMs can discover and order groups by complexity, 20 time series with different numbers of lobes and initial conditions were clustered into groups of equal complexity and ordered from least to most complex under two settings: *(a) equally sized groups, with the number and size of the groups specified*, and *(b) unequally sized groups, with both the number and size of the groups unknown*.

Task 4: Pairwise evaluation. To test whether LLMs can make independent judgments of complexity and chaos, pairs of time series were evaluated simultaneously for both properties, with the models determining which time series was more complex and which was more chaotic. Three cases were considered: *a) equal complexity but different chaos*, *b) equal chaos but different complexity*, and *c) differences in both complexity and chaos*.

Evaluations Metrics: The lobe count N associated with each time series was used as the ground-truth measure of complexity. To obtain the ground-truth measure of chaos, we estimated the largest Lyapunov exponent, λ_1 [23], following the standard renormalization procedure of Wolf et al. [24]. Normalized Kendall concordance, $C = (\tau + 1)/2$, was used to assess ranking performance in Task 1, where τ is the conventional Kendall rank correlation coefficient [25]. Tasks 2 and 4 were assessed using the F1 score. For Task 3, clustering was scored using pairwise accuracy. More details are given in Appendix A.4. Every score reported in Table 1 and Figure 1 is the mean of the corresponding per-trial metric over 10 independently repeated trials for each model and task.

4 Results

Ranking. In the complexity-ranking experiment (Task 1(a)), Claude and Gemini produced the correct ranking in all 10 trials, whereas GPT was correct in 8/10. Claude measured the angular separation θ between large-radius lobe tips and estimated the lobe count using $N \approx 360^\circ/|\Delta\theta|$. Gemini primarily used the same geometric method. This directly targets the geometrical quantity defining complexity in the benchmark, rather than a correlated dynamical feature. GPT instead ranked trajectories mainly by their angular advance during an early time window. Its errors illustrate an important failure mode: angular speed describes how the system evolves, whereas angular spacing reflects the geometrical structure being used here to define complexity.

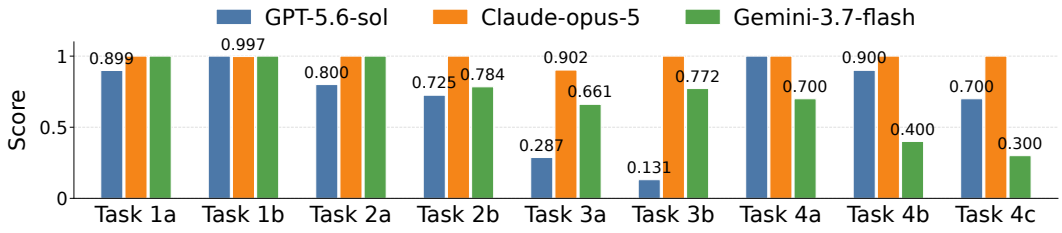

Figure 1: Scores by task. Labels for perfect scores (1) are omitted for simplicity.

In the chaos-ranking experiment (Task 1(b)), GPT and Gemini were correct in all 10 trials, while Claude was correct in 9/10. GPT ranked trajectories by their early separation from the common initial state and cross-checked the result using long-term irregularity, excursions, recurrence, and occasional formal diagnostics. Gemini used monotonic early-time growth as a proxy for the control parameter and, consequently, the degree of chaos. Claude adopted essentially the same proxy, but in one trial overrode it with subjective judgments based on apparent spikes, bursts, and regularity. The resulting inversion shows that apparent regularity in a finite time series can be misleading when used in place of a measure of dynamical instability.

Outlier detection. For single-outlier detection (Task 2(a)), all three models used the same core geometric approach: centering each trajectory, converting it to polar coordinates, isolating large-radius excursions, and clustering the corresponding angular directions. Claude and Gemini consistently used the cross-trajectory distinction between five $72°$-spaced clusters and ten $36°$-spaced clusters. GPT made two errors by overinterpreting secondary features, including apparent additional angular sectors, winding differences, and harmonic structure. Its inconsistent lobe estimates suggested sensitivity to centering, radial thresholds, and angular clustering, which could split or merge genuine lobes or admit spurious radial peaks. Thus, even when the appropriate geometric structure was identified, secondary dynamical features could override the feature relevant to the requested judgment.

For multiple-outlier detection (Task 2(b)), Claude applied the most reliable procedure: it estimated the lobe count independently for every trajectory from recurring angular directions and only then compared those estimates with the majority count, which correctly identified all six non-majority trajectories. GPT and Gemini were less reliable when they substituted dynamical proxies such as winding, recurrence, or radial structure for direct lobe counting. These measures sometimes captured visits, winding, or radial oscillations rather than distinct spatial lobes, producing frequent false negatives and occasional false positives. Gemini performed best when it used direct angular spacing, but errors increased when it reduced the geometry to coarse spatial regions or grouped time series by overall resemblance. These shortcuts discarded angular resolution and allowed atypical trajectories to be absorbed into the majority group.

Complexity-based clustering. For clustering with a fixed group structure (Task 3(a)), GPT, Claude, and Gemini were fully correct in 1/10, 9/10, and 5/10 trials, respectively. The dominant failure mode was again the use of dynamical proxies, GPT frequently relied on speed, turning rate, recurrence, radial behavior, and initial-condition matching instead of directly counting angularly separated outer lobes. Claude's only error similarly arose from using angular speed, radial oscillations, and sign changes, which describe timing or coordinate projections rather than spatial lobe count. Gemini's radial-maximum and symmetry detection was inconsistent, and its reliance on matching trajectories across initial-condition classes caused missed or repeated lobe visits to produce incorrect group assignments, particularly for closely spaced, high-N lobes.

For clustering with an unknown group structure (Task 3(b)), Claude was fully correct in all 10 trials. It consistently identified stable lobe directions from radial maxima, angular spikes, or dwell regions and estimated N from their angular separations or the number of angular transitions per revolution. GPT was not fully correct in any of the trials. Although it identified the correct number of groups in 7/10 trials, it never recovered the complete group-size sequence. In its best trial, it correctly assigned 16 of 20 labels and recovered 25 of 32 within-group pairs. Gemini was fully correct in 2/10 trials, with one additional trial recovering the correct memberships but not their complexity ordering. Across all trials, it recovered 48 of 70 exact groups, identified the correct seven-group total in 6/10 trials, and matched all group sizes in 4/10. GPT's and Gemini's errors primarily involved splitting or merging

groups because of inconsistent lobe counting and excessive reliance on similarities associated with initial conditions.

Pairwise evaluation of complexity and chaos. When the trajectories had ***equal complexity but different chaos*** (Task 4(a)), GPT and Claude selected the correct claims in all 10 trials. Gemini was fully correct in 4/10 trials, although it evaluated complexity correctly in 9/10 and chaos correctly in 5/10. Its errors arose mainly from inconsistent lobe definitions and from treating shared attractor geometry or the presence of chaos in both trajectories as evidence of equal chaos.

When the trajectories had ***equal chaos but different complexity*** (Task 4(b)), Claude and GPT selected the correct claims in 10/10 and 9/10 trials, respectively. GPT's single error resulted from miscounting the lobes as equal and interpreting lower repeatability as greater chaos. Gemini was fully correct in 0/10 trials: it assessed chaos correctly in 8/10 but complexity incorrectly in every trial. Its coarse representation of the geometry systematically concealed the greater number of recurring angular sectors in the more complex time series.

When ***both complexity and chaos differed*** (Task 4(c)), Claude selected the correct claims in all 10 trials. GPT was fully correct in 5/10 trials, with complexity and chaos evaluated correctly in 5/10 and 9/10 trials, respectively. Gemini was fully correct in 0/10 trials, with complexity correct in 0/10 and chaos in 6/10. GPT's errors were dominated by inconsistent visual lobe counting, whereas Gemini did not reliably isolate recurring angular lobes and sometimes treated shared attractor structure as evidence of equal dynamical instability.

The results show that successful judgments were associated with representations that directly exposed the requested dynamical property. Failures often occurred not because the models extracted no useful information from the time series, but because they relied on plausible features that did not correspond to the quantity being evaluated. Thus, having access to relevant information is not sufficient; it must be represented or analyzed in a way that makes the intended dynamical property apparent.

5 Conclusion and Future Work

Claude-Opus-5 was the best-performing model, with a total average score of 0.989. GPT-5.6-sol and Gemini-3.7-flash obtained average scores of 0.716 and 0.735, respectively. Their performance was strongly task-specific. Overall, all three LLMs handled the ranking tasks (Task 1) and single-outlier detection (Task 2(a)) relatively easily. GPT struggled most with complexity-based clustering, particularly the unknown-group variant (Task 3(b)), whereas Gemini struggled most with pairwise claim evaluation, especially when both complexity and chaotic behavior differed (Task 4(c)). More importantly, the controlled nature of the benchmark allowed us to examine not only whether the LLM judges reached the correct answer, but whether their judgments were based on the intended dynamical property. Successful judgments generally relied on features directly related to the requested quantity, such as angular lobe spacing for geometrical complexity or trajectory divergence for chaotic instability. In contrast, failures frequently occurred when models relied on plausible but incorrect proxies, such as angular speed, recurrence, radial structure, or overall geometric resemblance. **Thus, an LLM can extract meaningful structure from scientific time series while still failing to evaluate the construct it was asked to measure. For LLMs to serve as reliable judges of scientific data, having access to relevant information is not sufficient; they must identify and evaluate the features that correspond to the intended scientific quantity**.

Future work should evaluate additional LLMs to determine whether the successful strategies and failure modes identified here are model-specific or represent more general properties of LLM-based scientific judgment. We should also systematically vary time-series length to determine how the amount of available dynamical information affects model judgments and whether longer observations improve the accuracy of GPT and Gemini or reduce their reliance on misleading proxies. A wider range of tasks, such as forecasting, regime classification, parameter inference, and anomaly localization, will further assess the ability of LLMs to understand time series and dynamical systems. For scientific tasks in which reliable expert judgments are available, comparisons with human experts could further reveal whether LLMs use similar physically meaningful features or arrive at correct conclusions through different proxies. Finally, this framework should be extended to comparisons with state-of-the-art time-series machine-learning methods [26] to other dynamical systems, such as the Mackey–Glass systems [27], and ultimately to real-world time-series, where the relevant dynamical properties may not be independently controlled or known a priori.

References

[1] Preetham Sivalingam, Murari Mandal, Saurabh Deshpande, and Dhruv Kumar. LLM-as-a-judge for time series explanations. *arXiv preprint arXiv:2604.02118*, 2026.

[2] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, 2025.

[3] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[4] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.

[5] Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*, 2023.

[6] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[7] Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. *arXiv preprint arXiv:2411.16594*, 2024.

[8] Xiaodong An, Mikael Toye, and Flavio H. Fenton. Quantifying the complex spatiotemporal chaos of cardiac fibrillation in ionic models across parameter regimes. *arXiv preprint arXiv:2508.14303*, 2025.

[9] Ling Tang, Huiling Lv, Fengmei Yang, and Lean Yu. Complexity testing techniques for time series data: A comprehensive literature review. *Chaos, Solitons & Fractals*, 81:117–135, 2015.

[10] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical Review Letters*, 88(17):174102, 2002.

[11] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[12] David P. Feldman and James P. Crutchfield. Measures of statistical complexity: Why? *Physics Letters A*, 238(4-5):244–252, 1998.

[13] Guido Boffetta, Massimo Cencini, Massimo Falcioni, and Angelo Vulpiani. Predictability: a way to characterize complexity. *Physics Reports*, 356(6):367–474, 2002.

[14] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[15] Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*, 2023.

[16] Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the everything in the whole wide world benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2021.

[17] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V. Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in LLM-as-a-judge. *arXiv preprint arXiv:2410.02736*, 2024.

[18] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[19] Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. *arXiv preprint arXiv:2406.12624*, 2024.

[20] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[21] Mike A. Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

[22] Ching Chang, Yidan Shi, Defu Cao, Wei Yang, Jeehyun Hwang, Haixin Wang, Jiacheng Pang, Wei Wang, Yan Liu, Wen-Chih Peng, and Tien-Fu Chen. A survey of reasoning and agentic systems in time series with large language models. *Transactions on Machine Learning Research*, 2026.

[23] Jean-Pierre Eckmann and David Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57(3):617–656, 1985.

[24] Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317, 1985.

[25] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[26] Afrouz Delshad and Elizabeth M Cherry. Predicting complex time series with deep echo state networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(9), 2025.

[27] Michael C. Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197 (4300):287–289, 1977.

[28] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

A Appendix

A.1 N-Lorenz System

All time series used in our experiments were generated from the anchored N-Lorenz system, an angular reparameterization of the ordinary Lorenz system [11]

$$\dot{u} = \sigma(v - u), \qquad \dot{v} = u(\rho - z) - v, \qquad \dot{z} = uv - \beta z, \tag{1}$$

with parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$. Writing (u, v) in polar form using $u = r\cos\theta$, $v = r\sin\theta$, the equivalent cylindrical-coordinate system can be obtained:

$$\dot{r} = r\left[(\sigma + \rho - z)\sin\theta\cos\theta - \sigma\cos^2\theta - \sin^2\theta\right], \tag{2}$$

$$\dot{\theta} = (\rho - z)\cos^2\theta - \sigma\sin^2\theta + (\sigma - 1)\sin\theta\cos\theta, \tag{3}$$

$$\dot{z} = r^2\sin\theta\cos\theta - \beta z. \tag{4}$$

The N-lobed system is obtained by anchoring $N/2$ copies of this same angular dependence around the circle: replacing θ with the auxiliary angle $\phi = \frac{N}{2}(\theta - \frac{\pi}{4}) + \frac{\pi}{4}$ and rescaling $\dot{\theta}$ (rather than $\dot{\phi}$) by $d\theta/d\phi = 2/N$, gives

$$\dot{r} = r\left[(\sigma + \rho - z)\sin\phi\cos\phi - \sigma\cos^2\phi - \sin^2\phi\right], \tag{5}$$

$$\dot{\theta} = \frac{2}{N}\left[(\rho - z)\cos^2\phi - \sigma\sin^2\phi + (\sigma - 1)\sin\phi\cos\phi\right], \tag{6}$$

$$\dot{z} = r^2\sin\phi\cos\phi - \beta z. \tag{7}$$

N is a positive integer that sets the number of lobes; for $N = 2$, $\phi = \theta$ and the system reduces exactly to the ordinary Lorenz system above. Trajectories are converted to Cartesian coordinates via $x = r\cos\theta$, $y = r\sin\theta$, which is the (x, y) representation used throughout the input data.

The system has N nonzero fixed points, anchored at

$$\theta_k = \frac{\pi}{4} + \frac{2\pi k}{N}, \qquad r_k = r_L = \sqrt{2\beta(\rho - 1)}, \qquad z_k = \rho - 1, \qquad k = 0, \ldots, N - 1, \tag{8}$$

so that the $k = 0$ fixed point always coincides with the positive fixed point of the ordinary Lorenz system, and every lobe shares the common radius r_L. Since $\theta \mapsto \phi$ is a diffeomorphism, the anchored system is diffeomorphic to the ordinary Lorenz system for every N, and the Lyapunov exponent is invariant under a diffeomorphism. N is therefore varied to change complexity at fixed chaos, while ρ can instead be varied at fixed N to change chaos independently of complexity. All datasets consisted of bivariate time series, with only (x, y) values provided at each time step, as the transformation was applied only to x and y, with z integrated but excluded from the input.

A.2 Task-Specific Dataset Composition

The trajectories were integrated using a time step of $\Delta t = 0.1$, with $\sigma = 10$, $\beta = \frac{8}{3}$, and $\rho = 28$, unless mentioned otherwise. N denotes the number of lobes.

Task 1: Ranking

(a) Complexity ranking. The dataset contained 20 time series with 2-21 lobes. The time series were randomly ordered, and each LLM was asked to rank them from least to most complex.

(b) Chaos ranking. The dataset contained 20 time series with one series generated for each value of $\rho \in \{25, 25.5, 26.8, 28, 29, 30.4, 32.05, 33.5, 34.8, 36.7, 38.1, 39.7, 42, 44, 45.7, 49, 50.5, 53.5, 56, 59\}$, exhibiting different degrees of chaotic behavior. All time series were generated with N=15, and were randomly ordered. Each LLM was asked to rank these time series from least to most chaotic. Figure 2 shows the ρ values used in this study and their corresponding Lyapunov exponents.

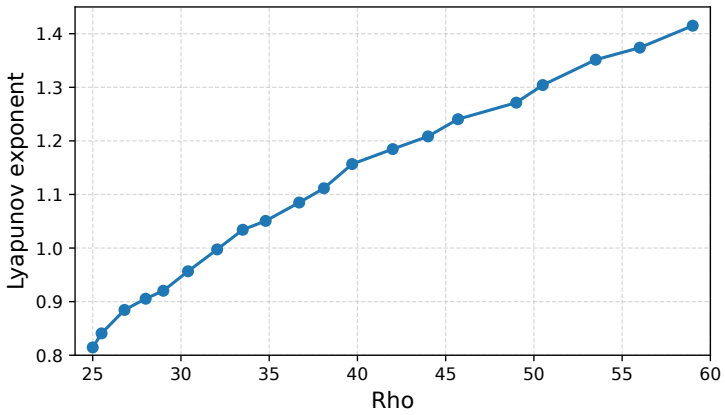

Figure 2: ρ versus Lyapunov exponent for the dataset used to evaluate Task 1b.

Task 2: Outlier detection

(a) Single-outlier detection. The dataset contained 20 time series, with 19 having the same number of lobes $(N = 5)$ but generated using different initial conditions, and one having a different number of lobes $(N = 10)$. The time series were randomly ordered, and each LLM was asked to identify the outlier time series based on its relative complexity.

(b) Multiple-outlier detection. Fourteen series (70% of the dataset) had the same number of lobes $(N = 10)$ but were generated using different initial conditions. The six remaining time series were generated with different number of lobes $N \in \{3, 7, 12, 14, 18, 21\}$. Each LLM was asked to identify the time series that differed from the majority in terms of complexity.

Task 3: Complexity-based clustering

(a) Fixed group structure. The dataset contained 20 time series divided into four equal groups of five. The groups corresponded to $N \in \{4, 11, 16, 20\}$. Within each group, the time series shared the same number of lobes N but were generated using distinct initial conditions. These time series were randomly ordered, and their corresponding lobe counts were not disclosed to the LLMs; however, the number of groups and their sizes were provided. Each LLM was asked to cluster and order the time series according to their inferred complexity levels.

(b) Unknown group structure. The dataset contained 20 time series divided into seven groups of unequal sizes. The groups corresponded to $N \in \{2, 6, 8, 11, 14, 16, 19\}$ and contained 2, 2, 7, 3, 1, 1, and 4 time series, respectively. Repeated instances of the same N were generated using distinct initial conditions. These time series were randomly ordered, and neither the lobe counts nor the number of groups or group sizes were disclosed to the LLMs. Each LLM was asked to cluster and order the time series according to their inferred complexity levels and also report its estimates of the number of the groups and group sizes.

Task 4: Pairwise evaluation

Each instance contained two time series that were randomly assigned the labels A and B. Each LLM was instructed to act as a scientific evaluator to assess the following six candidate claims, and to select all claims that it considered supported by the data: (1) series A is more complex than series B; (2) series A is more chaotic than series B; (3) series B is more complex than series A; (4) series B is more chaotic than series A; (5) series A and series B are equally complex; and (6) series A and series B are equally chaotic.

Three variants were evaluated to determine whether and how the LLMs changed their judgments in response to different relationships between the input time series:

(a) Similar complexity levels but different levels of chaotic behavior. Both time series were generated with $N = 8$, one with $\rho = 28$ and the other with $\rho = 50$. Figures 3, and 4 show these time series and their $x - y$ projections.

(b) Similar levels of chaotic behavior but different complexity levels. Both time series were generated with $\rho = 50$, one with $N = 8$ and the other with $N = 17$. Figures 4, and 6 show these time series and their $x - y$ projections.

(c) Different levels of both complexity and chaotic behavior. One of the time series was generated with $N = 8$ and $\rho = 50$, while the other was generated with $N = 17$ and $\rho = 28$. Figures 4, and 5 show these time series and their $x - y$ projections.

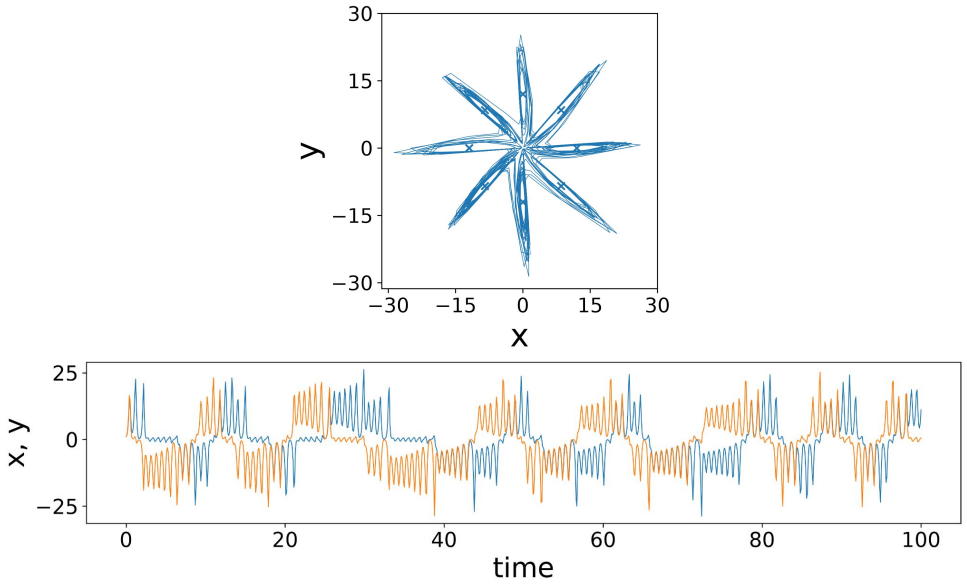

Figure 3: Dataset with lower complexity ($N = 8$) and lower chaos (Lyapunov exponent = 0.9054). The top figure shows the x-y projection, and the bottom figure shows the x and y time series.

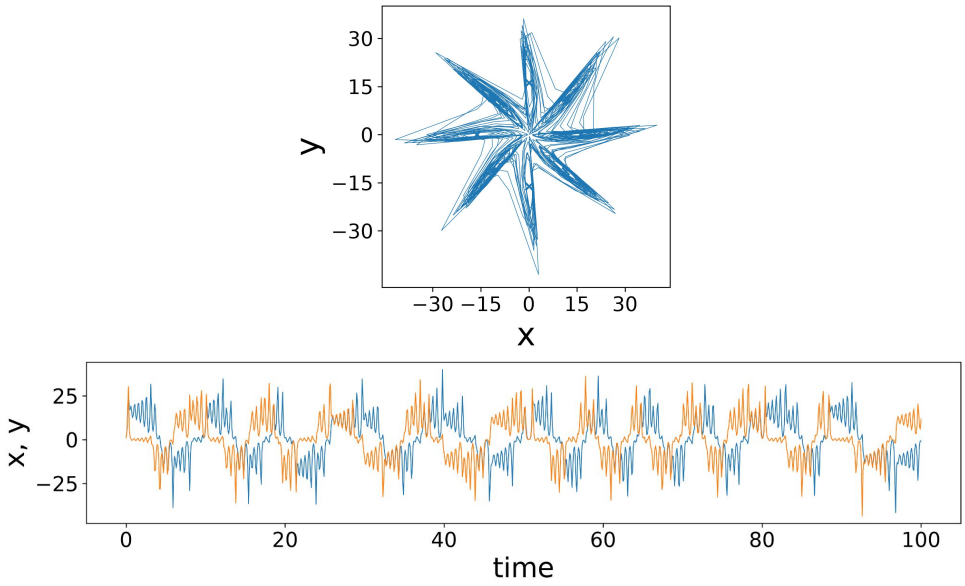

Figure 4: Dataset with lower complexity ($N = 8$) and higher chaos (Lyapunov exponent = 1.2764). The top figure shows the x-y projection, and the bottom figure shows the x and y time series.

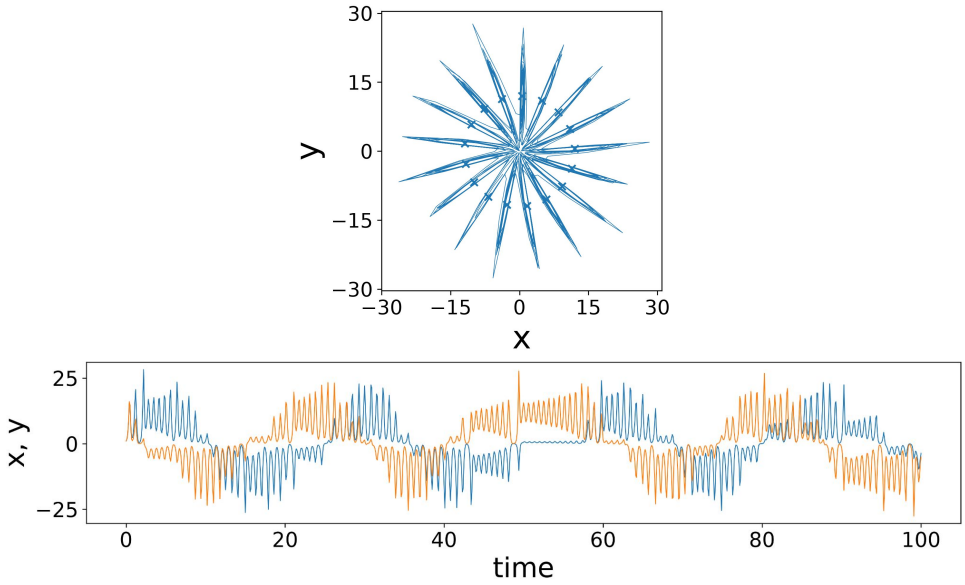

Figure 5: Dataset with higher complexity ($N = 17$) and lower chaos (Lyapunov exponent = 0.9054). The top figure shows the x-y projection, and the bottom figure shows the x and y time series.

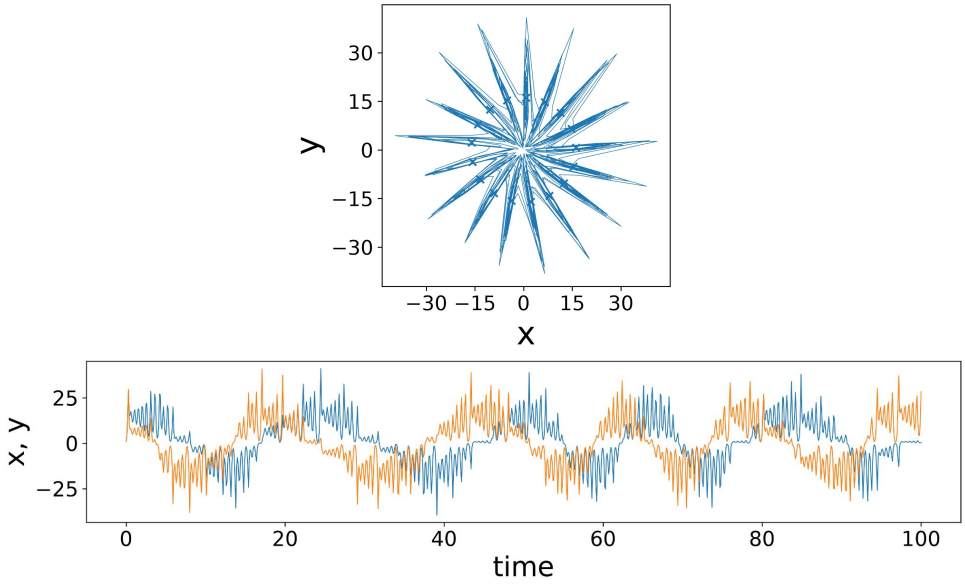

Figure 6: Dataset with higher complexity ($N = 17$) and higher chaos (Lyapunov exponent = 1.2764). The top figure shows the x-y projection, and the bottom figure shows the x and y time series.

A.3 Prompts

TASK1A PROMPT

```
You are analyzing trajectories from a dynamical system. Each trajectory orbits N distinct
"lobes" around a central axis; N is an unknown positive integer that differs between
trajectories. You are given only the raw (t, x, y) data and must infer N for each
trajectory from its shape and statistics.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. Order the labels by ascending N (fewest lobes first, most last).

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer with all {n} labels, each exactly once. Each block must appear exactly once, with
the json block last. Do not restate the raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"order": ["<lowest N>", "...", "<highest N>"]}}
```
```

TASK1B PROMPT

```
You are analyzing trajectories from a dynamical system. Each trajectory exhibits a
different degree of chaotic behavior. You are given only the raw (t, x, y) data and must
infer the level of chaos for each trajectory.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. Order the labels in ascending order of chaos level (least chaotic first, most last).
```

````
Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer with all {n} labels, each exactly once. Each block must appear exactly once, with
the json block last. Do not restate the raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"order": ["<least chaotic>", "...", "<most chaotic>"]}}
```
````

TASK2A PROMPT

````
You are analyzing trajectories from a dynamical system. Each trajectory orbits N distinct
"lobes" around a central axis. You are given only the raw (t, x, y) data and must infer N
for each trajectory from its shape and statistics.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. {n_minus_1} of them share the same N (each started from a different initial
condition, so their paths differ in phase and shape, but the lobe count is the same);
exactly one trajectory has a different N. Find the label of that one outlier.

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer. Each block must appear exactly once, with the json block last. Do not restate the
raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"outlier": "<label>"}}
```
````

TASK2B PROMPT

````
You are analyzing trajectories from a dynamical system. Each trajectory orbits N distinct
"lobes" around a central axis. You are given only the raw (t, x, y) data and must infer N
for each trajectory from its shape and statistics.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. Most of them share the same majority N (each started from a different initial
condition, so their paths differ in phase and shape, but the lobe count is the same); the
rest have an N that differs from that majority (and possibly from each other). Determine
how many trajectories differ from the majority, and find the label of each one.

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer. List every outlier label exactly once, in any order; if you believe there are none,
 use an empty list. Each block must appear exactly once, with the json block last. Do not
restate the raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"outliers": ["<label>", "..."]}}
```
````

TASK3A PROMPT

````
You are analyzing trajectories from a dynamical system. Each trajectory orbits N distinct
"lobes" around a central axis; N is an unknown positive integer that differs between
trajectories. You are given only the raw (t, x, y) data and must infer N for each
trajectory from its shape and statistics.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. They form exactly {num_groups} groups of {group_size} trajectories each: every
trajectory in a group shares the same N (each started from a different initial condition,
so their paths differ in phase and shape), and every group has a different N from every
other group. Cluster the labels into their {num_groups} groups, and order the groups from
least to most complex (lowest-N group first).

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer with all {n} labels, each exactly once, grouped as a list of {num_groups} lists
ordered from least to most complex. Each block must appear exactly once, with the json
block last. Do not restate the raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"groups": [["<label>", "...", "<label>"], ..., ["<label>", "...", "<label>"]]}}
```
````

TASK3B PROMPT

````
You are analyzing trajectories from a dynamical system. Each trajectory orbits N distinct
"lobes" around a central axis; N is an unknown positive integer that differs between
trajectories. You are given only the raw (t, x, y) data and must infer N for each
trajectory from its shape and statistics.

You will be given {n} trajectories, each labeled with a single letter, in no particular
order. They form some unknown number of groups, each containing some unknown number of
trajectories: every trajectory in a group shares the same N (each started from a different
 initial condition, so their paths differ in phase and shape), and every group has a
different N from every other group. Cluster the labels into groups, order the groups from
least to most complex (lowest-N group first).

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer with all {n} labels, each exactly once, grouped as a list of lists ordered from
least to most complex. Each block must appear exactly once, with the json block last. Do
not restate the raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"groups": [["<label>", "...", "<label>"], ..., ["<label>", "...", "<label>"]]}}
```
````

TASK4 PROMPT

````
You are analyzing two trajectories, labeled A and B, from a dynamical system. Each
trajectory orbits some number of distinct "lobes" around a central axis and may exhibit
chaotic behavior to some degree; both properties are unknown and must be inferred from the
 raw (t, x, y) data.
````

````
Complexity means the number of lobes a trajectory visits; chaos means its dynamical
instability (sensitivity to initial conditions). The two properties are distinct and need
not covary.

Act as a scientific evaluator. Consider the following six candidate claims about the two
trajectories:

1. Series A is more complex than Series B.
2. Series A is more chaotic than Series B.
3. Series B is more complex than Series A.
4. Series B is more chaotic than Series A.
5. Series A and Series B are equally complex.
6. Series A and Series B are equally chaotic.

Select every claim that you consider supported by the data. Claims 1/3/5 are mutually
exclusive with each other (exactly one should hold for complexity), and claims 2/4/6 are
mutually exclusive with each other (exactly one should hold for chaos).

Structure your response as two fenced blocks, in this order: a ```reasoning block
containing your reasoning process, followed by a ```json block containing your final
answer. Each block must appear exactly once, with the json block last. Do not restate the
raw data in your response.

```reasoning
<your reasoning process>
```

```json
{{"claims": [<claim number>, "..."]}}
```
````

A.4 Detailed Evaluation Metrics

The largest Lyapunov exponent, λ_1 [23], was estimated following the standard renormalization procedure of Wolf et al. [24]. Two trajectories separated by δ_0 are integrated in parallel, renormalized to δ_0 every $\Delta s = 0.2$, and averaged over M steps, where T is the total integration time:

$$\lambda_1 = \frac{1}{T} \sum_{i=1}^{M} \ln \frac{\delta_i}{\delta_0}. \tag{9}$$

Task 1. Kendall's tau [25] was used as an indicator of complexity, using normalized Kendall concordance $C = (\tau + 1)/2$. It ranks correlation between the predicted label order and the true ascending-N order:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}, \tag{10}$$

where n is the number of time series to be ranked, and n_c and n_d are the numbers of concordant and discordant pairs:

$$n_c = \#\{(i,j) : i < j,\ \mathrm{rank}_i < \mathrm{rank}_j\}, \qquad n_d = \#\{(i,j) : i < j,\ \mathrm{rank}_i > \mathrm{rank}_j\}, \tag{11}$$

Here i, j index positions in the model's predicted order ($i < j$ means the label at position i was predicted before the label at position j), and rank_i is the true N of the label placed at position i.

A perfect order has $C = 1$, while $C = 0$ is fully reversed, and $C = 0.5$ is no better than random.

Task 2 & 4. Outlier detection (Task 2) and pairwise claim evaluation (Task 4) both reduce to the same underlying problem: comparing a predicted label/claim set against the true one. Both are therefore scored with the same F1 score:

$$P = \frac{|S_{\mathrm{pred}} \cap S_{\mathrm{true}}|}{|S_{\mathrm{pred}}|}, \qquad R = \frac{|S_{\mathrm{pred}} \cap S_{\mathrm{true}}|}{|S_{\mathrm{true}}|}, \qquad F_1 = \frac{2PR}{P+R}, \tag{12}$$

with $F_1 \in [0, 1]$. For Task 2, S_{pred} is the set of labels the model reported as outliers and S_{true} is the true outlier set. For Task 2a, $|S_{\mathrm{true}}| = 1$, so F_1 reduces to a simple correct/incorrect indicator. For

Task 4, S_{pred} and S_{true} are instead the model's selected and true claim-number sets out of the six candidate claims per instance ($|S_{\text{true}}|$ is always 2).

$F_1 = 1$ represents an exact match between the predicted and true sets, $F_1 = 0$ is complete disagreement, and intermediate values give proportional credit for partial overlap.

Task 3. Clustering was scored using pairwise accuracy, in the spirit of the Rand index [28], covering both the fixed (3a) and unknown (3b) group-structure variants. For every pair of time series, we check whether the model and the ground truth agree on whether that pair belongs in the same group; pairwise accuracy is simply the share of pairs where they agree:

$$\text{PA} = \frac{\text{\# of pairs where model and ground truth agree}}{\text{total \# of pairs}}, \tag{13}$$

with $\text{PA} \in [0, 1]$. Labels the model omits from its predicted grouping are each treated as their own singleton group, so incomplete predictions are penalized rather than ignored.

$\text{PA} = 1$ is a perfect partition match (up to relabeling of the groups), $\text{PA} = 0$ means the model disagreed with the truth on every single pair.

A.5 Table of results

Table 1: Task results. Best results for each task are shown in blue.

Score	GPT-5.6-sol	Claude-opus-5	Gemini-3.7-flash
Task 1a	0.899	1.000	1.000
Task 1b	1.000	0.997	1.000
Task 2a	0.800	1.000	1.000
Task 2b	0.725	1.000	0.784
Task 3a	0.287	0.902	0.661
Task 3b	0.131	1.000	0.772
Task 4a	1.000	1.000	0.700
Task 4b	0.900	1.000	0.400
Task 4c	0.700	1.000	0.300
Overall Score	0.716	0.989	0.735