From Distributions to Programs: A Benchmark for Algorithmic Inference

Eduardo Y. Sakabe[1,4] *eyujis@gmail.com*
Felipe S. Abrahão[2,3,4,5] *felipesabrahao@gmail.com*
Hector Zenil[3,4,6,7] *hector.zenil@algocyte.ai*
Ricardo R. Gudwin[1] *gudwin@unicamp.br*

[1]School of Electrical and Computer Engineering, University of Campinas (UNICAMP), Brazil
[2]Centre for Logic, Epistemology and the History of Science, University of Campinas (UNICAMP), Brazil
[3]Oxford Immune Algorithmics, Oxford University Innovation & London Institute for Healthcare Engineering, U.K.
[4]Algorithmic Dynamics Lab, Karolinska Institute & King's College London, U.K.
[5]DEXL, National Laboratory for Scientific Computing (LNCC), Brazil
[6]Research Departments of Biomedical Computing and Digital Twins, School of Biomedical Engineering and Imaging Sciences
[7]King's Institute for Artificial Intelligence, King's College London, U.K.

Abstract

We introduce the Algorithmic Inference Benchmark (AIB), a framework designed to distinguish whether learning systems capture the generative mechanisms underlying data or merely fit statistical regularities. AIB constructs synthetic datasets using known rules, allowing independent control over algorithmic and statistical difficulty. By manipulating the generative rule and sampling process, AIB enables controlled experiments that reveal whether models rely on algorithmic abduction or statistical prediction. This conceptual work outlines the design principles of AIB and motivates its use for developing learning systems with stronger algorithmic priors.

Keywords — Algorithmic Information Theory, Algorithmic Information Dynamics, Shannon Entropy, Machine Learning.

1. Introduction

A fundamental question in machine learning is whether models capture the *generative mechanisms* underlying data or merely approximate its *statistical regularities*. Consider a simple physical experiment in which a metal sphere is released from different heights on a ramp and its speed at the bottom is recorded. A model trained to predict the sphere's speed from its release height may simply interpolate the observed data, estimating the conditional distribution of outputs given inputs. Alternatively, it may internalize the physical law, such as conservation of mechanical energy or the equations of motion that govern the system. The interpolation-based approach remains tied to the observed distribution, while the mechanistic one generalizes beyond the samples by recovering the process that generates them. This example illustrates that learning from data involves not only predicting outcomes but also modeling the processes that generate them.

This perspective frames learning as an *inverse problem* [1, 2], where the goal is to infer a hidden generative process from the observations it produces. Supervised learning fits into this setting [3]: given input–output pairs, the learning system aims to model the rule that maps inputs to labels. However, it is essential to distinguish between (i) *statistical prediction*, where inference proceeds by approximating the probability distribution of the observations, and (ii) *algorithmic abduction*, which seeks the causal and mechanistic procedures that generate them.

This distinction is far from trivial. Because deep learning models achieve high predictive accuracy, they are often portrayed as exhibiting *reasoning* capabilities or functioning as *world models* [4, 5], suggesting that they capture the structure of the data on which they are trained. Nevertheless, their performance is brittle under distribution shifts [6], vulnerable to spurious correlations [7], and unstable in non-stationary environments [8]. Even architectures such as transformers show these limitations [9, 10, 11]. Such behaviors indicate that these systems often rely on statistical shortcuts instead of uncovering the generative structure of the data, motivating their characterization as performing nothing more than *statistical pattern matching*. At the same time, studies such as [12, 13] argue that phenomena like grokking and double descent in overparameterized models reflect a bias toward simplicity, and conjectures such as [14] propose that the internal representations learned by these models may function like short programs that capture the underlying generative structure of the data.

Practical approaches to this problem often analyze the internal parameters of specific models, for example by measuring compression [15] or examining simplicity bias [16]. However, these methods depend on particular architectures and training dynamics, and therefore do not provide a general methodology for evaluating what learning systems have actually acquired. To address this limitation, we introduce the *Algorithmic Inference Benchmark* (AIB), a model-agnostic suite of synthetic datasets in which each instance is generated by a known program. By independently varying the algorithmic and statistical properties of both the generative and sampling mechanisms, AIB enables controlled experiments that reveal whether learning systems recover the underlying programs that generate the data or remain confined to fitting surface-level patterns.

2. The Algorithmic Inference Benchmark

This work is at the conceptual stage and outlines the design principles of the Algorithmic Inference Benchmark (AIB). We define a dataset as a collection of input–output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where each pair is generated by a known rule r_j and a sampling procedure σ that determines the initial conditions $x_i \sim \sigma$. In our current setup, each output is obtained as $y_i = r_j(x_i)$, where r_j denotes one of the 256 possible rules of the one-dimensional elementary cellular automata (ECA) [17]. In this formulation, x_i and y_i are binary strings representing consecutive states of the automaton under rule r_j. The combined specification of the sampling process σ and the generative rule r_j defines the statistical and algorithmic characteristics of $\mathcal{D}$. One-dimensional CAs serve as a tractable initial implementation, while the same formulation generalizes to any computable transformation model, providing a scalable framework for constructing datasets with controllable algorithmic structure.

The difficulty of each dataset in AIB can be characterized along two complementary dimensions: its *statistical* and *algorithmic* complexity.

2.1. Statistical Difficulty

A natural method to quantify the statistical difficulty of a dataset is to ask how much uncertainty remains about the outputs y after observing the inputs x. In information theory, this notion is captured by the conditional Shannon entropy $H(Y \mid X)$, which measures the minimal expected number of bits required to describe Y given knowledge of X [18]. High conditional entropy corresponds to outputs that are statistically unpredictable from the inputs, indicating high statistical difficulty.

However, in the setting considered in AIB, each dataset is generated by a deterministic transformation $y = r_j(x)$. In this case, $H(Y \mid X)$ collapses to zero, regardless of how complex or irregular the mapping may appear from a statistical perspective. Thus, the standard conditional entropy fails to capture meaningful notions of statistical difficulty for deterministic generative processes.

To obtain a more informative measure, we introduce a partition operator ρ that decomposes each binary string into fixed-length segments:

$$\rho(x) = (x_1, x_2, \dots, x_{m_x}), \qquad \rho(y) = (y_1, y_2, \dots, y_{m_y}),$$

where each block x_i and y_j belongs to $\{0,1\}^k$.

The operator ρ specifies the partition scheme, such as non-overlapping windows or sliding windows, used to extract local patterns.

To quantify the statistical relationship between local patterns of x and y, we construct an empirical joint distribution over all block pairs that occur *within the same example*. Let

$$C(x_i, y_j) = \#\{(n, a, b) \ : \ \hat{x}_a^{(n)} = x_i, \ \hat{y}_b^{(n)} = y_j\}$$

count how many times the pattern pair (x_i, y_j) appears across the dataset, where $\hat{x}_a^{(n)}$ and $\hat{y}_b^{(n)}$ denote the a-th and b-th blocks of the n-th example. The empirical joint distribution is then

$$p(x_i, y_j) = \frac{C(x_i, y_j)}{N\, m_x m_y},$$

which corresponds to the frequency of co-occurrences of (x_i, y_j) relative to the total number of block comparisons $N m_x m_y$ (each block of x compared with each block of y within the same example).

The marginal distribution over input blocks follows naturally from the joint,

$$p(x_i) = \sum_{y_j} p(x_i, y_j) = \frac{\#\{(n, a) : \hat{x}_a^{(n)} = x_i\}}{N m_x},$$

ensuring internal consistency when defining conditional distributions.

We define the *partition-based conditional entropy* as

$$H_\rho(Y|X) = -\sum_{x_i, y_j} p(x_i, y_j) \, \log \frac{p(x_i, y_j)}{p(x_i)}.$$

This quantity measures the uncertainty of local patterns in y given those in x under the chosen partition scheme ρ. Low values of $H_\rho(Y|X)$ indicate that the local configurations of x strongly constrain those of y, corresponding to low statistical difficulty. Conversely, higher values reflect greater statistical irregularity, where correlations between local patterns weaken despite the deterministic generative rule.

Because the empirical distribution of blocks depends on the sampling process σ, the value of $H_\rho(Y|X)$ varies with σ. We hypothesize that changing σ provides a useful mechanism for modulating statistical difficulty, enabling AIB to probe how input sampling influences the apparent statistical structure of the data independently of the underlying rule.

2.2. *Algorithmic Difficulty*

While statistical difficulty captures how strongly the distributions of local patterns in x and y are correlated under the sampling process σ, it does not directly account for the algorithmic structure of the generative rule r_j itself. To quantify this intrinsic mechanistic complexity, we adopt a measure based on *conditional Kolmogorov complexity* [19]. The conditional complexity of an object y given x is defined as

$$K(y \mid x) = \min\{\, |p| : U(p, x) = y \,\},$$

which represents the length of the shortest program that outputs y when provided with x as input to a universal Turing machine U. As with ordinary Kolmogorov complexity, this quantity is uncomputable. However, in our controlled setting, each dataset is generated by a known transformation r_j, allowing us to approximate its algorithmic difficulty via the complexity of the rule itself.

We estimate $K(r_j)$ using the Coding Theorem Method (CTM) [20], which approximates Kolmogorov complexity from the empirical output distribution of small programs. By exhaustively enumerating all machines in a restricted computational model and applying Levin's coding theorem [21, 22], CTM approximates

$$K(x) \approx -\log_2 m_D(x),$$

where $m_D(x)$ denotes the empirical algorithmic probability of x. Let $\eta(r_j) \in \{0,1\}^8$ denote the canonical 8-bit representation of an elementary cellular automaton; we approximate

$$K(r_j) \approx K_{CTM}(\eta(r_j)).$$

By the invariance theorem, Kolmogorov complexity is stable under any computable recoding [23, 19, 24]. Thus, using the canonical binary encoding $\eta(r_j)$ changes the true complexity of the rule by at most a constant.

For example, the complexity of *Rule 90* can be approximated by $K_{CTM}(01011010)$, its canonical 8-bit transition-table encoding. This captures how compressible the underlying generative mechanism is, independently of the sampling distribution σ.

This approach is general and applies to any computable generative mechanism. When a rule requires an encoding larger than the CTM limit (12 bits), we estimate its complexity using the Block Decomposition Method (BDM) [25], which decomposes the encoding into CTM-sized pieces and aggregates their complexities with a penalization for repeated blocks. BDM preserves CTM's algorithmic grounding while behaving like entropy at larger scales.

2.3. *Evaluation Protocol: Algorithmic vs Statistical Learning*

AIB treats algorithmic and statistical difficulty as controllable parameters: the generative rule r_j sets the algorithmic difficulty of the dataset, while the sampling process σ sets its statistical difficulty.

In the main protocol, we fix r_j and vary σ. Changes in performance as σ varies indicate the extent to which a model depends on surface-level statistical correlations. Large performance drops suggest stronger reliance on statistical features of the observed data, whereas stability across sampling distributions indicates a greater degree of mechanism-based generalization.

At this stage, we restrict our analysis to this protocol and evaluate models using standard metrics such as accuracy or loss. Additional protocols and evaluation criteria are under development.

2.4. *An Ideal Algorithmic Learner*

An ideal learning system for the AIB setting would approximate the universal distribution induced by the Solomonoff prior [26], assigning higher probability to simpler generative explanations and selecting hypotheses according to their algorithmic probability. Such a system would implement an optimal form of Occam's razor [27]: it would identify the shortest program consistent with the data and generalize by preferring algorithmically minimal descriptions.

Although such a universal learner is uncomputable in principle, several research directions seek to approximate aspects of this behavior. These include models based on the minimum description length (MDL) principle [28], program-induction architectures [29], and approaches that explicitly incorporate algorithmic priors [30]. In Appendix A, we present a prototype model that captures part of this ideal within the constraints of finite computation.

3. Conclusion

The Algorithmic Inference Benchmark (AIB) provides a principled framework for determining whether learning systems rely on statistical correlations or recover the generative rules that underlie the data. We expect AIB to support a deeper understanding of whether a model captures distributions or infers programs, and to motivate approaches that incorporate stronger algorithmic inductive biases. Future extensions will explore broader classes of generative mechanisms, sampling schemes, and evaluation metrics, enabling more comprehensive studies of algorithmic abduction and offering a path toward more general solutions to inverse problems.

4. Acknowledgements

This project was supported by the Ministry of Science, Technology, and Innovation of Brazil, with resources granted by the Federal Law 8.248 of October 23, 1991, under the PPI-Softex. The project was coordinated by Softex and published as Intelligent agents for mobile platforms based on Cognitive Architecture technology [01245.003479/2024-10]. This study was partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001. Felipe S. Abrahão acknowledges support from the São Paulo Research Foundation (FAPESP), grants 2021/14501-8.

References

[1] A. Tarantola, *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Amsterdam ; New York : New York, NY, U.S.A: Elsevier Science, 1987.

[2] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, ser. Applied Mathematical Sciences, S. Antman, J. Marsden, and L. Sirovich, Eds. Springer, 2005, vol. 160. [Online]. Available: https://link.springer.com/10.1007/b138659

[3] E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, F. Odone, and P. Bartlett, "Learning from Examples as an Inverse Problem." *Journal of Machine Learning Research*, vol. 6, no. 5, 2005. [Online]. Available: http://www.jmlr.org/papers/volume6/devito05a/devito05a.pdf

[4] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, "Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task," Jun. 2024, arXiv:2210.13382 [cs]. [Online]. Available: http://arxiv.org/abs/2210.13382

[5] A. F. Spies, W. Edwards, M. I. Ivanitskiy, A. Skapars, T. Räuker, K. Inoue, A. Russo, and M. Shanahan, "Transformers Use Causal World Models in Maze-Solving Tasks," Mar. 2025, arXiv:2412.11867 [cs]. [Online]. Available: http://arxiv.org/abs/2412.11867

[6] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring Robustness to Natural Distribution Shifts in Image Classification," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 18 583–18 599. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/d8330f857a17c53d217014ee776bfd50-Abstract.html

[7] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, Nov. 2020, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s42256-020-00257-z

[8] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive Neural Networks," Oct. 2022. [Online]. Available: http://arxiv.org/abs/1606.04671

[9] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar, "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models," Oct. 2024, arXiv:2410.05229. [Online]. Available: http://arxiv.org/abs/2410.05229

[10] R. T. McCoy, S. Yao, D. Friedman, M. D. Hardy, and T. L. Griffiths, "Embers of autoregression show how large language models are shaped by the problem they are trained to solve," *Proceedings of the National Academy of Sciences*, vol. 121, no. 41, p. e2322420121, Oct. 2024. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.2322420121

[11] P. Shojaee, I. Mirzadeh, K. Alizadeh, M. Horton, S. Bengio, and M. Farajtabar, "The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity," Jul. 2025, arXiv:2506.06941 [cs]. [Online]. Available: http://arxiv.org/abs/2506.06941

[12] A. G. Wilson, "Deep Learning is Not So Mysterious or Different," Jul. 2025, arXiv:2503.02113 [cs]. [Online]. Available: http://arxiv.org/abs/2503.02113

[13] B. DeMoss, S. Sapora, J. Foerster, N. Hawes, and I. Posner, "The Complexity Dynamics of Grokking," Dec. 2024, arXiv:2412.09810 [cs]. [Online]. Available: http://arxiv.org/abs/2412.09810

[14] I. Sutskever, "Talk at the simons institute: Ilya sutskever (openai)," August 14 2023, relevant timestamp at 24:07. [Online]. Available: https://simons.berkeley.edu/talks/ilya-sutskever-openai-2023-08-14

[15] E. Y. Sakabe, F. S. Abrahão, A. Simões, E. Colombini, P. Costa, R. Gudwin, and H. Zenil, "Binarized Neural Networks Converge Toward Algorithmic Simplicity: Empirical Support for the Learning-as-Compression Hypothesis," Sep. 2025, arXiv:2505.20646 [cs]. [Online]. Available: http://arxiv.org/abs/2505.20646

[16] G. Valle-Pérez, C. Q. Camargo, and A. A. Louis, "Deep learning generalizes because the parameter-function map is biased towards simple functions," Apr. 2019, arXiv:1805.08522 [stat]. [Online]. Available: http://arxiv.org/abs/1805.08522

[17] S. Wolfram, "Cellular automata," *Los Alamos Science*, vol. 9, pp. 2–21, 1983. [Online]. Available: https://content.wolfram.com/sw-publications/2020/07/cellular-automata.pdf

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[19] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 4th ed., ser. Texts in Computer Science. Cham: Springer, 2019.

[20] J.-P. Delahaye and H. Zenil, "Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of randomness," *Applied Mathematics and Computation*, vol. 219, no. 1, pp. 63–77, Sep. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0096300311012367

[21] L. A. Levin, "Laws of information conservation (nongrowth) and aspects of the foundation of probability theory," *Problemy Peredachi Informatsii*, vol. 10, no. 3, pp. 30–35, 1974, publisher: Russian Academy of Sciences, Branch of Informatics, Computer Equipment and . . . . [Online]. Available: https://www.mathnet.ru/eng/ppi1039

[22] ——, "On a concrete method of assigning complexity measures," in *Doklady Akademii Nauk*, vol. 234. Russian Academy of Sciences, 1977, pp. 536–539, issue: 3. [Online]. Available: https://www.mathnet.ru/eng/dan40999

[23] C. S. Calude, *Information and Randomness: An algorithmic perspective*, 2nd ed. Springer-Verlag, 2002.

[24] G. Chaitin, *Algorithmic Information Theory*, 3rd ed. Cambridge University Press, 2004.

[25] H. Zenil, S. Hernández-Orozco, N. A. Kiani, F. Soler-Toscano, A. Rueda-Toicen, and J. Tegnér, "A Decomposition Method for Global Evaluation of Shannon Entropy and Local Estimations of Algorithmic Complexity," *Entropy*, vol. 20, no. 8, p. 605, Aug. 2018. [Online]. Available: https://www.mdpi.com/1099-4300/20/8/605

[26] R. J. Solomonoff, "Algorithmic Probability: Theory and Applications," in *Information Theory and Statistical Learning*, F. Emmert-Streib and M. Dehmer, Eds. Boston, MA: Springer US, 2009, pp. 1–23. [Online]. Available: https://doi.org/10.1007/978-0-387-84816-7_1

[27] G. Leuenberger, "General Mathematical Proof of Occam's Razor; Upgrading Theoretical Physicists' Methodology," Jun. 2025, arXiv:2506.23194 [cs]. [Online]. Available: http://arxiv.org/abs/2506.23194

[28] J. Rissanen, "Stochastic Complexity and Modeling," *The Annals of Statistics*, vol. 14, no. 3, pp. 1080–1100, 1986, publisher: Institute of Mathematical Statistics. [Online]. Available: https://www.jstor.org/stable/3035559

[29] K. Ellis, L. Wong, M. Nye, M. Sablé-Meyer, L. Cary, L. Anaya Pozo, L. Hewitt, A. Solar-Lezama, and J. B. Tenenbaum, "DreamCoder: growing generalizable, interpretable knowledge with wake–sleep Bayesian program learning," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 381, no. 2251, p. 20220050, Jun. 2023, publisher: Royal Society. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2022.0050

[30] P. Shaw, J. Cohan, J. Eisenstein, and K. Toutanova, "Bridging Kolmogorov Complexity and Deep Learning: Asymptotically Optimal Description Length Objectives for Transformers," Sep. 2025, arXiv:2509.22445 [cs]. [Online]. Available: http://arxiv.org/abs/2509.22445

A. Algorithmic Inference via Cellular Automata in Few-Shot Symbolic Tasks

Problem Setting

We consider tasks where the goal is to infer plausible output patterns y for previously unseen inputs x_{test}, given only a few example input-output pairs (x_i, y_i). Each x and y is a 4×4 binary matrix. The dataset is designed to emphasize low-data generalization based on algorithmic principles rather than statistical ones.

Method Overview

Our method leverages the generative capacity of cellular automata (CAs) to discover possible transformations from x to y that are algorithmically plausible and minimally complex. It proceeds in five main steps:

1. Rule Discovery

Simulate a large number (e.g., 1 million) of two-dimensional binary cellular automata rules. For each training pair (x, y), identify the subset of rules $\mathcal{R}$ that, when applied to x, generate y at some timestep before the system enters a repeating configuration, where the dynamics settle into a cycle.

2. Rule Complexity Ranking (Optional)

Each CA rule can be encoded in a binary string of length 512 bits (since a 9-bit neighborhood maps to a 1-bit output). We use the Block Decomposition Method (BDM) to estimate the algorithmic complexity of each rule. The rules in $\mathcal{R}$ are ranked by increasing complexity, under the assumption that simpler generative rules are more likely to reflect true underlying structure.

3. Rule Application to Test Inputs

For each x_{test}, apply the top-k rules (e.g., the 100 simplest rules from $\mathcal{R}$) to simulate their evolution. For each rule r, collect all candidate outputs y' that emerge during simulation.

4. Conditional Complexity Estimation

For each candidate output y', we estimate its conditional algorithmic complexity given x_{test} by computing its empirical frequency and applying the algorithmic coding theorem. The procedure is as follows:

- Let $\mathcal{R}_{y'} \subseteq \mathcal{R}$ be the subset of CA rules that produce y' from x_{test}.
- Let $|\mathcal{R}|$ be the total number of rules tried (e.g., the top-100 ranked by BDM), and let $|\mathcal{R}_{y'}|$ be the number of rules that generate y'.
- Define the empirical algorithmic probability as:

$$m_D(y' \mid x_{\text{test}}) = \frac{|\mathcal{R}_{y'}|}{|\mathcal{R}|} \tag{1}$$

- Applying the Coding Theorem Method, we approximate the monotonic conditional complexity as:

$$K_m(y' \mid x_{\text{test}}) = -\log_2 \left(\frac{|\mathcal{R}_{y'}|}{|\mathcal{R}|} \right) \tag{2}$$

- To obtain an upper bound on the conditional algorithmic complexity, we also record the minimal number of steps $t_{\min}$ required for any rule in $\mathcal{R}_{y'}$ to reach y'. This yields:

$$K(y' \mid x_{\text{test}}) \lessapprox K_{\text{CTM}}(y' \mid x_{\text{test}}) = K_m(y' \mid x_{\text{test}}) + \log_2(t_{\min}) \tag{3}$$

  where $K_{\text{CTM}}(y' \mid x_{\text{test}})$ denotes the upper bound derived from the Coding Theorem Method.

This estimate favors outputs generated by many simple rules and achieved quickly during simulation, aligning with the principles of algorithmic probability and Occam's razor.

5. Selection by Minimum Conditional Complexity

Finally, we select the output y' with the minimum estimated conditional complexity $K_{\mathrm{CTM}}(y' \mid x_{\mathrm{test}})$ as the best abduction/induction hypothesis for the given input.