MIIC-SR: From Complex Data to Structural Causal Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimating the set of mathematical equations from observational data for complex systems with nonlinear relationships presents a significant challenge, particularly when the model specification is not straightforward and the causal graph is not known. We propose a comprehensive framework that estimates a Structural Causal Model (SCM) from data, requiring no prior information on the underlying causal graph. Our framework incorporates MIIC (Multivariate Information-Based Inductive Causation), a well-established causal discovery algorithm, with Symbolic Regression (SR). Our results demonstrate that the association of MIIC and Symbolic Regression shows at least comparable results in SCM estimation on the studied benchmarks with the advantage of providing an interpretable causal model.

1 Introduction

To gain a deeper understanding of complex systems, such as those found in industries and healthcare, a causally oriented approach can be employed to identify and characterize the causal relationships among the components of these systems. These relations can be represented using Structural Causal Models (SCMs), which facilitate modeling and counterfactual reasoning, enabling the evaluation of the impact of interventions. SCMs have the potential to generate completely synthetic data, which can be of critical importance across various fields and in multiple scenarios.

However, estimating SCMs from data is challenging due to non-linear relationships and collinearity among variables. This necessitates robust performance in two tasks: causal discovery, which involves representing causal relationships through graphs, and structural causal modeling, which mathematically defines these relationships.

Traditional regression methods often ignore causal directions, predicting upstream features from downstream variables that do not influence them. To find reasonable SCMs and minimize this risk, we introduce the MIIC-SR framework, combining the MIIC causal discovery algorithm with Genetic Programming-Based Symbolic Regression. It estimates both linear and non-linear equations to build an effective, interpretable data-driven simulator where each feature is a function of its causal parents.

2 Material and Methods

2.1 Causal discovery algorithms

The causal discovery task consists of inferring graphical networks from observational data and identifying direct and potentially causal relations between variables. We compared multiple state-of-the-art and recent algorithms: DirectLingam [1], ICALiNGAM [2], PC algorithm [3], NotearsNonlinear [4], GOLEM [5], GraNDAG [6] and MIIC [7, 8]. In particular, MIIC has proven to be effective across diverse data distributions without requiring prior assumptions. It uses a constraint-based,

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

information-theoretic approach to efficiently select conditioning sets, addressing limitations of the PC algorithm. MIIC is robust to sampling noise, needs no hyperparameter tuning, and is available as an open-source web server or R package.

2.2 Symbolic Regression

Symbolic Regression (SR) is a data-driven method for discovering interpretable mathematical expressions linking input variables to a target output, without assuming predefined functional forms [9]. Unlike traditional techniques relying on fixed forms (e.g., linear or polynomial), SR uses evolutionary algorithms to explore a broad space of models. The most effective SR methods are based on Genetic Programming (GP), a metaheuristic inspired by biological evolution that enables efficient convergence to optimal solutions [10, 11, 12]. SR estimation via genetic programming was performed using the PySR package [13].

2.3 MIIC-SR method

We introduce a robust framework that uses MIIC and SR for full structural causal modeling with very few, if any, assumptions or particular knowledge of the dataset. The framework pipeline is applied on numerical datasets and can be divided into several steps:

1. Causal discovery: from a given dataset, we reconstruct the causal network using the MIIC algorithm, with consistent separating set search enabled [14], which outputs a CPDAG (Completed Partially Directed Acycle Graph).
2. Transform the MIIC reconstructed network to a Directed Acyclic Graph (DAG). MIIC reconstructed graph can contain a mixture of directed and non directed edges. For this step, we used the *pdag2dag* algorithm, present in the *pcalg* R package [15].
3. Generate data for nodes that do not have parents (generators or exogenous nodes). This step is performed by sampling from the observed marginal distribution of generator nodes.
4. For each remaining child node X_i in the graph, we apply SR to learn the regression function f_i linking $Pa(X_i)$ and X_i, with $Pa(X_i)$ being the set of predictors (or *parents*) of the target node X_i. To avoid overly convoluted expressions and to enhance the interpretability of our model, we introduce a set of structural constraints (see the supplementary section for more insights on pipeline complexity and SR parameters).

2.4 Comparative approach

To assess the accuracy of MIIC-SR in estimating the SCM, we compared it to other causal discovery algorithms paired with SR. We also used classical Generalized Linear Models (GLM) with interaction terms as a baseline, allowing interactions between predictors. Our benchmark included seven causal discovery algorithms, with only PC and MIIC (the top performers) used alongside SR or GLM. The protocol involved: i) generating training data from a specific SCM; ii) learning graphs (causal discovery) and regression formulas (SR); iii) generating synthetic data from the estimated SCMs; iv) comparing synthetic data to an unseen test set built from the same SCM. We analyzed various sample sizes to evaluate their impact on performance. The full pipeline is shown in Figure S1. The synthetic data generated through our pipeline is also benchmarked against leading methods for synthetic data generation, such as MIIC-SDG [16] and Synthpop [17] along with a random marginal feature generation as reference.

2.4.1 Synthetic data

We first analyzed a graph with 2 colliders (Figure 1a), highlighting the critical role of causal discovery in accurately deriving a valid SCM, as discussed in the Results section.
For comparison against other methods, we took into account different tasks and models:

1. We generated a training and test dataset, based on a defined SCM:
   - **Symprod Simpson Graph**: a graph inspired by CSuite (Figure 2a) with 7 nodes and 7 edges. The SCMs, reported in Table S1 is non-linear.

- **Steel toughness**: This study uses Weibull-distributed data to predict steel toughness [18], with parameter dependencies from a Gaussian copula. The dataset includes 10 variables, detailed in section C.

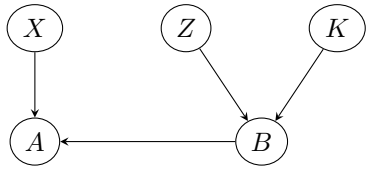
2. We analyzed the **Fault Detection Dataset in Photovoltaic Farms** from Kaggle [19, 20], representing a 250-kW Photovoltaic power plant.

2.4.2 Performances evaluation

The comparison between the generated synthetic data and the test set is made through the multivariate **Wasserstein distance**. We used the Sinkhorn regularized version [21].

3 Results

To show the importance of causal discovery before regression, we built a network (Figure 1a) and ran our pipeline. MIIC and PC correctly identified the parent nodes for A and B, allowing SR to retrieve the accurate SCM (Figure 1b). In contrast, without causal discovery, SR mistakenly identified X and A as parents of B, violating the SCM causal direction.



(a)

Equation	Method	SCM
$B = 2.5Z + 2.2K$	MIIC-SR	$B = 2.5Z + 2.2K$
	PC-SR	$B = 2.5Z + 2.2K$
	SR	$B = 0.322A - 0.713X$
$A = 2.21X + 3.11B$	MIIC-SR	$A = 2.21X + 3.11B$
	PC-SR	$A = 2.21X + 3.11B$
	SR	$A = 2.21X + 3.11B$
$X \sim \mathcal{N}(0, 1)$ $Z \sim \mathcal{N}(0, 1)$ $K \sim \mathcal{N}(0, 1)$		

(b)

Figure 1: (a) Network with 2 colliders. (b) Comparison between true SCM (first column) and estimated SCM (last column) for the compared methods. Small variations of the constants can sometimes be observed over multiple executions.

We considered the "Symprod Simpson" network (Figure 2a), with a more complex and non-linear SCM. Networks reconstructed by MIIC and PC are shown in Figure 2b and 2c, using 10k samples. MIIC correctly retrieves all directed edges, while PC fails to capture the causal association $X_0 \to X_2$ and $X_1 \to X_2$. For node X_1, the PC algorithm mistakenly identifies X_3 as a parent (Figure 2c). For the PC network, the GLM equation for X_1 includes all 3 identified parents, while instead SR correctly excludes X_3 and derives the correct regression formula (Table S1). Even with the correct network (MIIC case), the linear nature of GLM provides only a linear approximation of the right formula, whereas SR captures the exact equation. For node X_2, the PC algorithm identifies only one association with node Z_2, proposing an undirected edge (Figure 2c). This allows PC-GLM and PC-SR methods to include only Z_2 in the estimation of the causal equation (Table S1). In contrast, the MIIC algorithm accurately identifies all parent nodes for X_2, allowing both MIIC-SR and MIIC-GLM models to find the correct mathematical equation, that is this time linear.

To assess the stability of our method and the impact of sample size, we conducted the Symprod benchmark with sizes 100, 200, 500, 1000, 5000 and 10000, each with 10 iterations. Causal discovery performances, measured by precision, recall, and F1 scores, are shown in Figure S2, demonstrating the reliability of the MIIC algorithm. We evaluated the Mean Squared Error (MSE) for MIIC-SR's regression tasks using known exogenous variables for each endogenous node in the Symprod Simpson Graph. Results are shown in Figure S3. While generalized linear models (GLMs) reach a plateau for X_1 and X_3, indicating their struggle with nonlinear associations, symbolic regression (SR) achieves an MSE of 0 at $n = 5000$. MIIC-SR performs comparably or better than PC-SR for X_1 and X_3 across most sample sizes. However, for node X_2, even with 10,000 samples, PC-SR fails to obtain low MSE due to the PC algorithm's limitations in identifying the predictors of X_2.

Figure 3 reports the multivariate Wasserstein distance between the generated data and the test set. The MIIC-SR algorithm (in black) demonstrates performance comparable to Synthpop and MIIC-SDG,

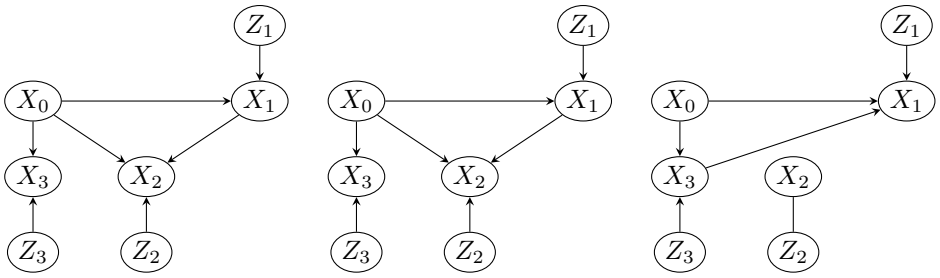


Figure 2: Symprod Simpson Graph (a) and the graph identified by MIIC (b) and PC (c) algorithms.

while outperforming PC-SR. Figure S4 also reports the distance with the training set and shows the random method performances as a baseline. It can be noticed that Synthpop exhibits low distances from the training set but higher distances from the test set, suggesting a tendency to overfit the training data. Similar results have been obtained for the steel toughness dataset, Figure S5, while the Photovoltaic Faults dataset, Figure S6, shows some limitations of our model. In these datasets, methods show similar performances, with PC capable of generating good-quality data. It is important to note that Synthpop and MIIC-SDG do not estimate SCMs, thus lacking the capacity to provide fully explainable models capable of performing interventions.

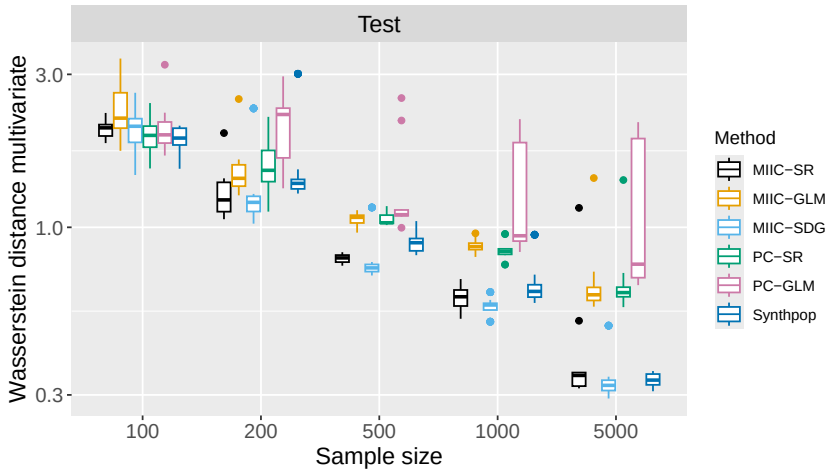


Figure 3: Multivariate Wasserstein distance of synthetic data from test data.

4 Discussion

In this paper, we demonstrate that integrating a state-of-the-art causal discovery algorithm with symbolic regression enables accurate estimation of SCMs without making prior assumptions about data distribution or model structure. Our results emphasize the importance of causal discovery in accurately identifying parent nodes, which is essential to understand causal mechanisms and specify predictive models correctly. Moreover, our flexible pipeline allows to incorporate domain-expert knowledge, when already estimated feature equations or physical-chemical laws are known.

From a theoretical perspective, some challenges remain. One of them arises from the class of networks found by constraint-based causal discovery methods that do not guarantee the output of a completely oriented graph (DAG), but rather a combination of directed and undirected edges, representing a class of Markov equivalences [22, 23]. For this reason, multiple DAGs could be derived from the same graph, possibly impacting the outcome. Moreover, it is crucial to assess the accuracy of the estimated SCMs without relying solely on the evaluation of regression errors or distribution distances. In this context, a relevant approach would involve computing distances between estimated and true mathematical equations using tree representation-based equations [24]. This would provide a better understanding of the accuracy and correctness of the estimated SCMs. Lastly, to further evaluate the methodology, it would be useful to test it on more complex systems and real-world datasets.

References

[1] Shohei Shimizu et al. "DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model". In: *Journal of Machine Learning Research-JMLR* 12.Apr (2011), pp. 1225–1248.

[2] Shohei Shimizu et al. "A Linear Non-Gaussian Acyclic Model for Causal Discovery". In: *Journal of Machine Learning Research* 7 (2006), pp. 2003–2030. URL: http://www.jmlr.org/papers/v7/shimizu06a.html.

[3] Peter Spirtes and Clark Glymour. "An algorithm for fast recovery of sparse causal graphs". In: *Social science computer review* 9.1 (1991), pp. 62–72.

[4] Xun Zheng et al. *Learning Sparse Nonparametric DAGs*. 2020. arXiv: 1909.13189 [stat.ML]. URL: https://arxiv.org/abs/1909.13189.

[5] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. "On the role of sparsity and dag constraints for learning linear dags". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17943–17954.

[6] Sébastien Lachapelle et al. "Gradient-based neural dag learning". In: *arXiv preprint arXiv:1906.02226* (2019).

[7] Louis Verny et al. "Learning causal networks with latent variables from multivariate information in genomic data". In: *PLoS computational biology* 13.10 (2017), e1005662.

[8] Vincent Cabeli et al. "Learning clinical networks from medical records based on information estimates in mixed-type data". In: *PLoS computational biology* 16.5 (2020), e1007866.

[9] Gabriel Kronberger et al. *Symbolic Regression*. CRC Press, 2024.

[10] John R Koza et al. "Hierarchical genetic algorithms operating on populations of computer programs." In: *IJCAI*. Vol. 89. 1989, pp. 768–774.

[11] John R Koza. *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*. Vol. 34. Stanford University, Department of Computer Science Stanford, CA, 1990.

[12] John R Koza. "Genetic programming as a means for programming computers by natural selection". In: *Statistics and computing* 4 (1994), pp. 87–112.

[13] Miles Cranmer. *Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl*. 2023. arXiv: 2305.01582 [astro-ph.IM]. URL: https://arxiv.org/abs/2305.01582.

[14] Honghao Li et al. "Constraint-based causal structure learning with consistent separating sets". In: *Advances in neural information processing systems* 32 (2019).

[15] Dorit Dor and Michael Tarsi. "A simple algorithm to construct a consistent extension of a partially oriented graph". In: *Technicial Report R-185, Cognitive Systems Laboratory, UCLA* (1992), p. 45.

[16] Nadir Sella et al. "Preserving information while respecting privacy through an information theoretic framework for synthetic health data generation". In: *npj Digital Medicine* 8.1 (2025), p. 49.

[17] Beata Nowok, Gillian M Raab, and Chris Dibben. "synthpop: Bespoke creation of synthetic data in R". In: *Journal of statistical software* 74 (2016), pp. 1–26.

[18] Nadia Pérot and Nicolas Bousquet. "Functional Weibull-based models of steel fracture toughness for structural risk analysis: estimation and selection". In: *Reliability Engineering & System Safety* 165 (2017), pp. 355–367.

[19] SS Ghoneim, Amr E Rashed, and Nagy I Elkalashy. "Fault detection algorithms for achieving service continuity in photovoltaic farms". In: *Intelligent Automation & Soft Computing* 30.2 (2021), pp. 467–479.

[20] Amr Ezz El-Din Rashed. *Fault Detection Dataset in Photovoltaic Farms*. https://www.kaggle.com/datasets/amrezzeldinrashed/fault-detection-dataset-in-photovoltaic-farms.

[21] Jean Feydy et al. "Interpolating between optimal transport and mmd using sinkhorn divergences". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2681–2690.

[22] Markus Kalisch and Peter Bühlmann. "Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm". In: *Journal of Machine Learning Research* 8 (2007), pp. 613–636. URL: http://www.jmlr.org/papers/v8/kalisch07a.html.

[23] Steven B. Gillispie and Michael D. Perlman. *Enumerating Markov Equivalence Classes of Acyclic Digraph Models*. 2013. arXiv: 1301.2272 [cs.AI]. URL: https://arxiv.org/abs/1301.2272.

[24] Tatsuya Akutsu et al. *Tree Edit Distance with Variables. Measuring the Similarity between Mathematical Formulas*. 2021. arXiv: 2105.04802 [cs.DS]. URL: https://arxiv.org/abs/2105.04802.

5 Supplementary section

A Algorithmic pipeline

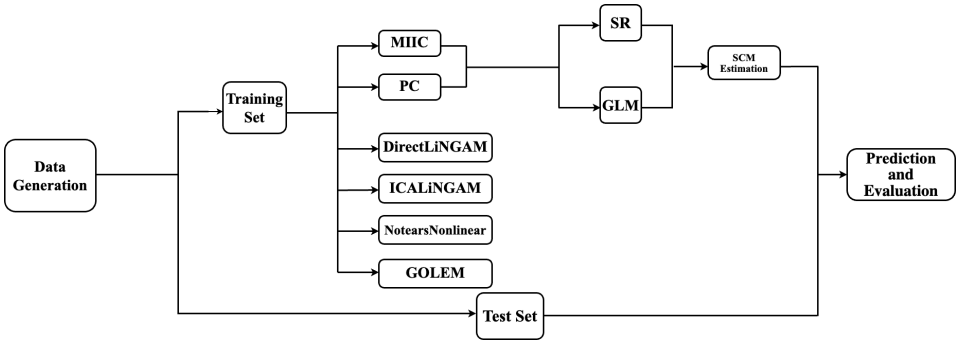


Figure S1: Pipeline for the execution and evaluation of the different causal learning and regression methods.

B Symprod Sympson

B.1 Symprod Sympson equations

True SCM	Method	Estimated SCM
$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$	MIIC-GLM	$X_1 = 1.458X_0 + 0.323Z_1 + 0.001X_0Z_1$
	MIIC-SR	$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$
	PC-GLM	$X_1 = 1.146X_0 + 0.453X_3 + 0.325Z_1 + 0.004X_0Z_1$ $-0.005Z_1X_3 - 0.006X_0Z_1X_3 + 0.001$
	PC-SR	$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$
$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$	MIIC-GLM	$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$
	MIIC-SR	$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$
	PC-GLM	$X_2 = 0.709Z_2 + 0.729$
	PC-SR	$X_2 = Z_2 + 0.726$
$X_3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$	MIIC-GLM	$X_3 = 0.689X_0 + \sqrt{\frac{3}{10}}Z_3$
	MIIC-SR	$X_3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$
	PC-GLM	$X3 = 0.689X0 + \sqrt{\frac{3}{10}}Z3$
	PC-SR	$X3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$
$Z_1 \sim t_3$, $Z_2 \sim Laplace(1)$, $Z_3 \sim \mathcal{N}(0,1)$, $X_0 \sim \mathcal{N}(0,1)$		

Table S1: Symprod Sympson equations. In general, the algorithm yields approximations that are remarkably close to the true constants of the problem, although it does not recover the exact values themselves.

B.2 Causal Discovery Algorithms Performances

To assess the performance of the causal discovery algorithms, we evaluate their accuracy using key metrics: Skeleton Precision (or Positive Predictive Value) $Prec = TP/(TP + FP)$, Recall (or Sensitivity) $Rec = TP/(TP + FN)$, and $F\text{-}score = (2 \times Prec \times Rec)/(Prec + Rec)$, the harmonic mean between $Prec$ and Rec. True Positives (TP) refer to correctly identified causal

relationships, False Positives (FP) are incorrectly identified relationships, and False Negatives (FN) are missed relationships.

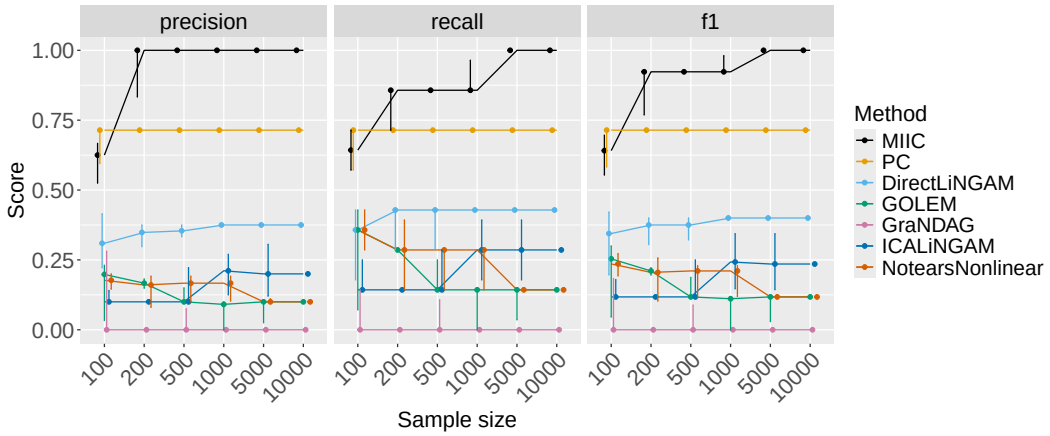


Figure S2: Evaluation of Precision, Recall, and F-score for MIIC (black line), PC (second best algorithm, in yellow), DirectLINGAM, GOLEM, GraNDAG, ICALiNGAM and NoterarsNonlinear algorithms in multiple sample sizes (100, 200, 500, 1000, 5000, and 10000) in the Symprod Simpson Graph. Median values with first and third quartiles as error bars are reported.

B.3 Predicting endogenous variables from exogenous ones

In this part we report MSE estimations for endogenous nodes. Estimated formulas for SR and GLM are used from the proposed pipeline.

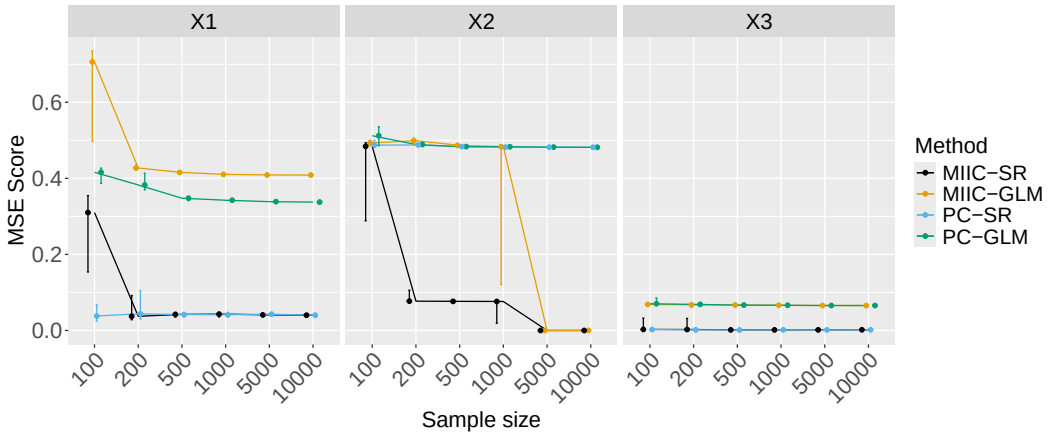


Figure S3: Evaluation of MSE in predicting X_1, X_2, and X_3 for the different methods and multiple sample sizes (100, 200, 500, 1000, 5000, and 10000) in the Symprod Simpson Graph. Median values with first and third quartiles as error bars are reported.

B.4 Distance of synthetic data from training and test data

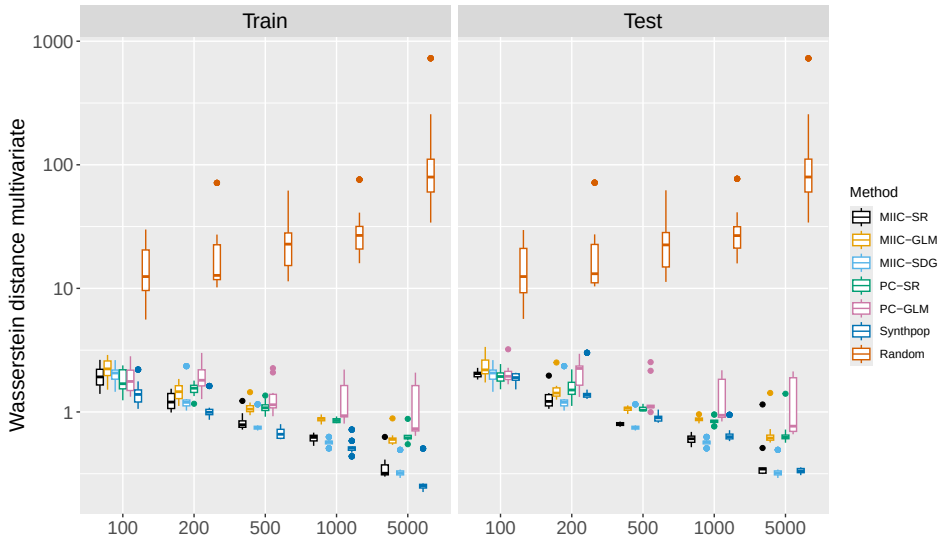


Figure S4: Multivariate Wasserstein distance of synthetic data from training and test data on the Symprod dataset for sample size 100, 200, 500, 1000 and 5000. Random method corresponds to generating each feature using uniform marginal distribution over the empirical range of the variable.

C Steel toughness

The simulation models material toughness as a function of temperature, incorporating parameter dependencies through a Gaussian copula.

- **Temperature distribution:** simulated from a uniform distribution over $[-200, 50]$ °C.
- **Parameter generation:** $(\alpha_0, \alpha_1, \lambda_1, \lambda_3)$ follow fitted normal distributions based on empirical data.
- **Dependency modeling:** A Gaussian copula ensures realistic correlations between these parameters.
- **Key computations:**
  - K_0 (baseline toughness) is modeled as a linear function of temperature.
  - K_u (ultimate toughness) follows an exponential temperature-dependent model.
  - failure probabilities are estimated using the Weibull model.
  - The expected toughness Y is computed using the gamma function.

Table S2 provides an overview of the input parameters (features) and their associated marginal distributions, complemented by a histogram to visualize the simulated values.

Features	**Description**	**Distribution**
T	Simulated temperature range	Uniform distribution : $\mathcal{U}(-200, 50)$
K_0	Initial toughness parameter	Linear relationship: $K_0 = \alpha_0 + \alpha_1 \cdot T$
K_u	Ultimate toughness parameter	Exp. relationship: $K_u = \lambda_0 + \lambda_1 \cdot \exp(\lambda_3 \cdot T)$
m	Shape parameter for Weibull failure model	Log-normal distribution
α_0	Intercept of linear toughness model	Normal: $\mathcal{N}(25, 3)$
α_1	Slope of linear toughness model	Normal: $\mathcal{N}(0.05, 0.01)$
λ_1	Parameter for exponential toughness model	Normal: $\mathcal{N}(10, 2)$
λ_3	Parameter for exponential toughness model	Normal: $\mathcal{N}(0.01, 0.002)$
Y	Simulated toughness values	Derived from K_0 and K_u

Table S2: Description of features and their distributions

C.1 Distance of synthetic data from training and test data

Multivariate Wasserstein distance of synthetic data from training and test data on the steel toughness dataset.

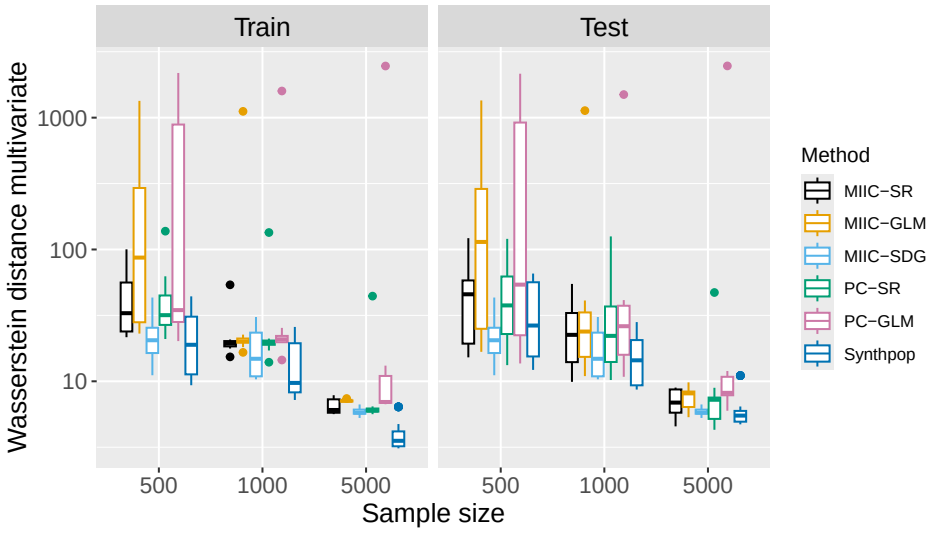


Figure S5: Multivariate Wasserstein distance of synthetic data from training and test data on the steel toughness dataset for sample size 500, 1000 and 5000.

D Photovoltaic faults

D.1 Distance of synthetic data from training and test data

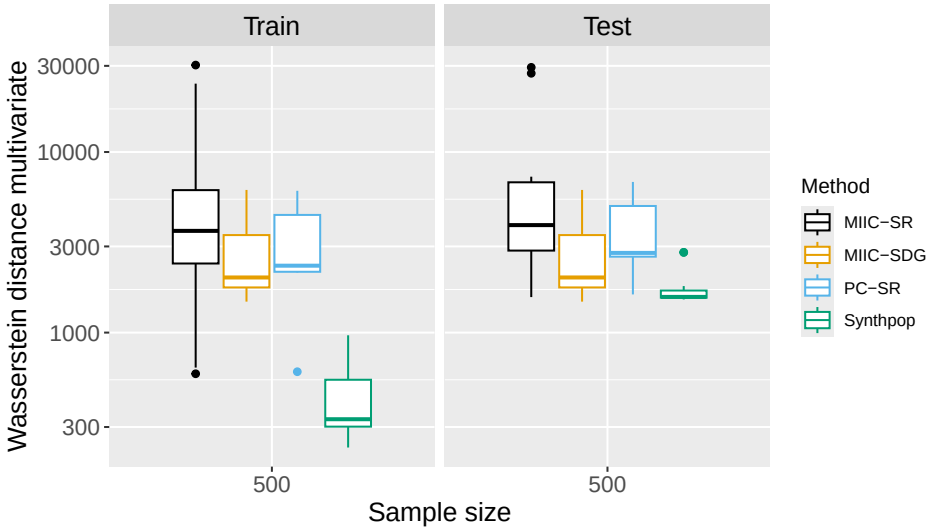


Figure S6: Multivariate Wasserstein distance of synthetic data from training and test data on the Photovoltaic Faults dataset for sample size 500. The estimation of larger samples sizes is not possible due to the size of the data. To be able to compare the best performing algorithms, MIIC-GLM and PC-GLM are not shown in the picture due to very large Wasserstein distance estimation.

E Pipeline complexity

The complexity of the SCM estimation lies primarily within the SR phase. The MIIC algorithm has already been applied in relatively large real scenarios (100 variables, 10,000 samples) and runs efficiently in a few minutes. The complexity of SR estimation is associated with the connectivity of the resulting causal network and to the choice of the regression parameters.

F Implementation Details of PySR

In the Symbolic Regression experiments, we employed the PySR package with the parameters reported in Table S3.

Certain functions are penalized by assigning them a high complexity score, and we impose *nested constraints* to limit the depth of function compositions. For instance, an expression such as $\exp(\tanh(x))^{\sin^2(x)}$ is highly unlikely to be selected, while repeated compositions like $\cos \circ \cos \circ \cos$ or $\sin \circ \cos \circ \cos$ are explicitly prohibited. This ultimately leads to more interpretable functions, more likely to have physical meaning, while avoiding overly complex expressions. Furthermore, for numerical stability, we extend all partially defined functions – such as square root, logarithm, or the exponentiation $x, y \mapsto x^y$ – by zero outside their domain of definition.

Here, in the `extra_sympy_mappings` category, s_safe_sqrt, s_safe_log, and s_safe_pow are custom SymPy-defined functions for the square root, logarithm, and power operations, respectively. These functions extend the domain of their standard counterparts to ensure numerical stability.

Moreover, to ensure that the magnitudes of the different loss values are comparable and to prevent issues with early stopping triggered by the `early_stop_condition`, we normalize the data prior to the regression phases.

Parameter	Description
`random_state`	42
`niterations`	200
`populations`	15
`population_size`	100
`maxsize`	10
`binary_operators`	`["+", "-", "*", "/", "SafePow(x, y) = (x < zero(x) && y % one(y) != 0) ? zero(x) : x^y"]`
`unary_operators`	`["sin", "cos", "tan", "sinh", "cosh", "tanh", "exp", "neg", "inv", "square", "abs", "floor", "ceil", "round", "SafeLog(x) = log(x < convert(typeof(x), 1e-10) ? convert(typeof(x), 1e-10) : x)", "SafeSqrt(x) = x < zero(x) ? zero(x) : sqrt(x)"]`
`extra_sympy_mappings`	`{"sin": sin, "cos": cos, "tan": tan, "sinh": sinh, "cosh": cosh, "tanh": tanh, "exp": exp, "square": lambda x: x**2, "abs": abs, "floor": sympy.floor, "ceil": sympy.ceiling, "round": lambda x: sympy.Function("round")(x), "inv": lambda x: 1/x, "neg": lambda x: -x, "SafeSqrt": s_safe_sqrt, "SafeLog": s_safe_log, "SafePow": s_safe_pow }`
`complexity_of_operators`	`{"+": 1, "-": 1, "*": 1, "/": 1, "neg": 1, "inv": 1, "SafeSqrt": 1.5, "square": 1.5, "abs": 2, "exp": 2, "SafeLog": 2, "sin": 2, "cos": 2, "tan": 2, "SafePow": 2, "sinh": 2.5, "cosh": 2.5, "tanh": 2.5, "floor": 3, "ceil": 3, "round": 3}`
`nested_constraints`	`{ "sin": {"sin": 1, "cos": 1, "tan": 1}, "cos": {"sin": 1, "cos": 1, "tan": 1}, "tan": {"sin": 1, "cos": 1, "tan": 1}, "sinh": {"sinh": 1, "cosh": 1, "tanh": 1}, "cosh": {"sinh": 1, "cosh": 1, "tanh": 1}, "tanh": {"sinh": 1, "cosh": 1, "tanh": 1}, "SafeLog": {"SafeLog": 1} }`
`elementwise_loss`	`loss(prediction, target) = (prediction - target)^2`
`early_stop_condition`	`stop_if(loss, complexity) = loss < 1e-10 && complexity < 10`

Table S3: PySR Parameters