SynthDAG a Synthetic data generation framework, Towards More Robust Iteration of Machine Learning Models

8. Semester, Oecon

Aalborg University Business School

AALBORG UNIVERSITET

Samuel Hvidager - 20195146

Supervisor: Thibault Laurentjoye

June 4, 2025

Abstract

Traditional machine learning algorithms, while powerful for prediction, are often agnostic to underlying causal structures, limiting their utility for policy analysis and robust decision-making. Existing benchmarks frequently lack the tools to systematically evaluate how ML models learn their causal relationships, particularly their sample efficiency and robustness. We introduce SynthDAG, a novel synthetic data generation framework designed to address this gap. SynthDAG utilizes Directed Acyclic Graphs (DAGs) and linear Structural Equation Models (SEMs) to hierarchically construct datasets with known, ground-truth mechanisms. A core contribution we add is our mechanism strategically masking a configurable subset of these nodes, this mechanism is intended to transform the fundamentally linear system, into one with observable, emergent non-linearities and to simulate real-world complexities like unobserved confounding or mediation. In practice, this approach in our current framework tends to generate only weak non-linearities, thereby creating a controlled testbed for evaluating how models discern underlying structure when faced with subtle, masking-induced complexities.

We demonstrated SynthDAG's capability to generate diverse datasets under deeply controlled and replicable conditions, mirroring real-world data artifacts such as varying kurtosis and heteroscedasticity. The framework offers fine-grained control, including the introduction of 'hub' nodes to concentrate causal influence, varied exogenous noise mechanisms (e.g., Gaussian, Laplace, mixture models) to modulate the strength of induced non-linearities, and a progressive sampling procedure to assess model learning curves. Crucially, SynthDAG facilitates a principled approach to benchmarking ML models by enabling evaluation against theoretical maximum R-squared values and their ability to recover true structural coefficients. We did evaluations utilizing SynthDAG to test various ML models; it was revealed that simpler linear models (e.g., OLS, Lasso) often demonstrated stronger predictive performance, consistently outperforming more complex non-linear models, particularly in recovering true structural coefficients, where Lasso excelled. This performance, coupled with the observation that the framework's current masking mechanism primarily induces weak non-linearities, underscores the challenge that complex models face in demonstrating superiority when underlying signals are subtly obscured rather than strongly non-linear. The discussion highlighted the fact that while our evaluations did confirm the framework's capacity to produce challenging yet tractable scenarios for assessing models, there are larger difficulties when it comes to consistently generating emergent non-linear behaviour, meaning we ultimately conclude that while SynthDAG is not robustly equipped to evaluate non-linear models yet, it provides a solid foundation paving the way for more rigorous development and deployment of causally-informed algorithms in the future.

We published our code as well as test as an R package available at https://github.com/Samuel99-2/SynthDAG

Contents

1 Preface **1**

2 Motivation and Outline of Research Focus **1**

3 Research Focus **1**

4 The DAG and Synthetic Data Generation **2**

5 Related Work **3**
5.1 Pitfalls of "off-the-shelf" Simulated DAGs 3
5.2 Benchmarks with Hidden Variables and Non-Linear Effects 4
5.3 End-to-end Causal Learners and Sample-Efficiency Reporting 4
5.4 Our Addition 4

6 Notation and Setup: The Synthetic Data Generation Framework **5**
6.1 Ground-Truth Causal Graph: Directed Acyclic Graph (DAG) 5
6.2 Structural Equation Model (SEM) 5
6.3 Masking Scheme: Introducing Latent Variables 6
6.4 Progressive Sampling Procedure 7

7 Implementation **8**
7.1 Core Data Generation in R 8
7.2 Hub Nodes: Concentrating Causal Influence 8
7.3 Noise Mechanisms for Robustness 8
7.4 Benchmarking using Theoretical Maximum R-squared 9
7.5 Initialization 10

8 Testing our Framework **11**
8.1 Robustness Test 11
8.2 Model Performance 16

9 Discussion **20**
9.1 Links Between Hubs, Number of Layers, and Increases in Model Accuracy 20
9.2 Why did Linear Models Consistently Perform Better? 21
9.3 Connection to Non-Linearity and Model Tests 21
9.4 Limits of our Framework and Possible Extensions 22

10 Conclusion **23**

11 Appendix **26**
11.1 Detailed Directed Acyclic Graph Generation Methodology 26
11.2 Additional plots and graphs 27

List of Figures

1 An example Directed Acyclic Graph (DAG). The acyclic nature allows for a topological ordering (e.g., A, B, C, F, D, E) essential for constructing Structural Equation Models (SEMs) 2
2 Residuals plotted with regime mixture enabled, showing weak nonlinearity compared to the base setup. 13
3 Graph illustrating bimodality of latent variables. 14
4 Standard set up, controlled seed generation, see 7.5 15
5 We see that the number of hubs has an outsized influence when skip layers is true 16
6 We see that increasing masking does lower the overall performance of models . . . 18
7 We see that the number of samples have clear influence on learning rate 19

List of Tables

1 Standard Parameters for Simulation Setup 10
2 Impact of Generator Settings on Synthetic Data Characteristics 12
3 Summary of Coefficient Recovery: OLS Estimated vs. True SEM Coefficients 16
4 Comparison of R-squared Values for Different Models and Sample Sizes (Target: N15) 17
5 Average Coefficiency recovery scores 17

1 Preface

This paper on synthetic data generation was written by Samuel Hvidager, with the supervision by Thibault Laurentjoye. Chicago citation style is used. The character count with spacing is 47349, equivalent to 19.72 pages of 2400 characters. Table of contents, citations, figures, images and appendix not accounted for.

2 Motivation and Outline of Research Focus

Machine learning methods that optimise out-of-sample prediction accuracy have become a routine part of the econometrician's toolkit (see Mullainathan and Spiess (2017)). Yet most of these algorithms are agnostic to causal structure, they fit $\mathbb{E}[Y \mid X]$ well, but they neither identify nor respect the structural relationships that drive policy. A model that excels at forecasting tax revenue may still deliver biased welfare estimates, if it conflates correlated covariates with true treatment effects. Without a principled identification strategy, researchers lack a solid measuring tool for deciding whether the additional predictive power of a given model justifies its adoption in policy analysis. To close this gap, we develop a synthetic-data framework whose ground-truth causal graph is known ex ante. By observing how quickly competing models learn the correct functional form as sample size grows, we obtain an objective measure of their causal learning ability, bringing ML evaluation back in line with the inference standards of applied econometrics.

3 Research Focus

We propose developing a synthetic-data framework whose full causal graph is known to us, but partially obscured from the models. This is achieved by constructing a ground-truth causal graph where all structural equations are fundamentally linear and additive. Non-linearities in the relationships between observed variables are then induced by strategically masking certain nodes.[1] The framework comprises:

(i) A directed acyclic graph (DAG) simulator in which all structural equations are linear, with coefficients and noise terms carefully randomized to prevent statistical shortcuts and ensure robust challenges for learners.

(ii) A masking scheme that hides a configurable subset of parent nodes from the learner

(iii) A progressive sampling procedure, which generates continuously larger nested training sets

Obscuring these latent parents converts the underlying linear relations into non-linear observable mappings, while preserving an analytically tractable "ground-truth" i.e. the causal mechanism behind the synthetic data.

[1] A key mechanism for generating emergent non-linearity involves designating latent common causes whose noise terms are drawn from distributions capable of representing such distinct underlying states, i.e. mixture models. For instance, a bimodal latent common cause can lead to sigmoidal conditional expectations between its observed dependent nodes. This process stimulates complex nonlinear patterns arising from simpler, but partially hidden, underlying linear dynamics.

For each DAG instance, we release a progressive sequence of training sets, $N_1 < N_2 < \cdots < N_K$, and evaluate every candidate model on every prefix. The resulting learning curves reveal the minimum sample size at which a model generalises to the true functional form.

Our Central Hypothesis can be stated as:

"Models that attain asymptotic generalisation with the fewest samples exhibit the most faithful inductive bias and the strongest bias-correction capability."

By translating the identification into a concrete, data-efficiency metric, our benchmark realigns ML model assessment with the inferential standards of applied econometrics, and creates an objective paradigm in which models can be evaluated. For further details about our theoretical setup, see section 6, notation and setup.

4 The DAG and Synthetic Data Generation

Our explanation of essential DAG theory in this section was in large part adapted from Scott Cunningham, 2021 book "Causal Inference: The Mixtape." as well as chapters 1 and 3 of Judea Pearl, 2009 book Causality: Models, Reasoning and Inference Series.

A DAG is a finite set of nodes, connected by directed edges such that you can never follow arrows and return to the same node, i.e. there are no directed cycles. In causal inference, the arrows encode assumptions about which variables can directly influence other downstream nodes.

Illustration of a Directed Acyclic Graph (DAG)

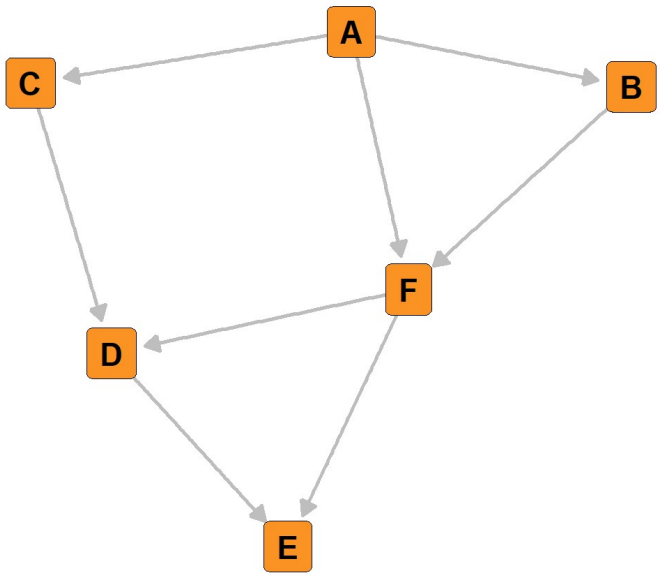

Figure 1: An example Directed Acyclic Graph (DAG). The acyclic nature allows for a topological ordering (e.g., A, B, C, F, D, E) essential for constructing Structural Equation Models (SEMs)

Because DAGs are acyclic, the nodes can be arranged in a topological order such that all arrows point forward. This ordering is key for constructing Structural Equation Models (SEMs). For example, consider the following DAG where one valid order is:

$$X_1 \rightarrow \{X_2, X_3\} \rightarrow Y.$$

This underpins the factorisation of the joint distribution via structural equations:

$$\begin{aligned} x_1 &= u_1, \\ x_2 &= f_2(x_1) + u_2, \\ x_3 &= f_3(x_1) + u_3, \\ y &= f_Y(x_2, x_3) + u_Y, \end{aligned}$$

Where each u_i (including u_Y) is an independent noise term. This functional form makes causal assumptions explicit and tractable, ideal for data generation process where traceable causality is desirable.

A DAG satisfies the Markov property: Each node is independent of its non- descendants given its parents. d-Separation is the graphical criterion that tells us which sets of variables block all back-door paths and hence yield valid conditional independencies. While methods like LiNGAM Shimizu et al. (2006) establish identifiability of fully observed linear SEMs under non-Gaussian noise, our framework introduces complexity by obscuring parent nodes. This transforms the underlying linear structural equations into non-linear observable relationships, presenting a different kind of identification challenge focused on learning these emergent functional forms.

5 Related Work

Synthetic data have become a staple for probing causal-learning algorithms because they let researchers train models on data with known ground-truth graphs and structural equations. Below we review literature that motivates, and delimit, the framework we propose in Section 6.

5.1 Pitfalls of "off-the-shelf" Simulated DAGs

Reisach, Seiler, and Weichwald (2021) et al. warned that generic additive-noise generators can leak statistical shortcuts. They identify *var-sortability*: a tendency for marginal variance to increase along the causal ordering. Continuous-data learners that implicitly rank variables by variance can therefore excel on such benchmarks without genuine causal reasoning. Their work spawned diagnostic tools and var-sortability baselines that routinely rival full-blown algorithms.

These findings compel benchmark designers to randomly choose coefficient scales, noise families, and measurement units. By incorporating these randomizations, our generator aims to produce benchmark datasets that are robust to such heuristic shortcuts, demanding more genuine causal reasoning from the evaluated models. We adopt this principle in our generator to avoid "gaming" of the system.

5.2 Benchmarks with Hidden Variables and Non-Linear Effects

A challenge of discovering causal structure in the case of unobserved latent variables, particularly when these introduce non-linear relationships. Kaltenpoth and Vreeken (2023) made significant strides in this domain by extending identifiability theory to non-linear causal models under latent confounding. They proposed a variational autoencoder approach capable of recovering both observed and hidden structure. Their synthetic experiments are interesting as they explicitly demonstrate how unobserved parent nodes can warp marginal dependencies into complex, non-monotone curves, precisely the phenomenon that a "hide-a-node" masking scheme is designed to replicate within a controlled, initially linear, environment. This allows us to study the emergence of non-linearity from a known, simpler underlying process. While their work shows the existence and learnability of such complex structures with latent variables, their evaluation methodology primarily focuses on the accuracy of structural recovery at a fixed dataset size. Consequently, the crucial question of how quickly different model classes adapt to these latent-induced non-linearities, and their relative sample efficiencies in doing so, remains largely unexplored in their work. Our framework aims to fill this specific gap by systematically assessing this rate of adaptation, providing a dynamic view of model performance beyond static accuracy metrics at a single sample size, and thereby addressing the efficiency aspect of learning complex causal dependencies.

5.3 End-to-end Causal Learners and Sample-Efficiency Reporting

Recent methods such as DECI bundle discovery and effectestimation into a single flow-based learner and demonstrate strong performance across hundreds of synthetic settings. Notably, DECI's ablation study records *learning curves* to justify architectural choices, but these curves are an internal diagnostic rather than the basis of a public benchmark. Our framework seeks to externalise that idea: we expose all candidate models to identical, progressively expanding datasets and score them by the area under the learning curve or the "first-to-generalise" threshold Geffner et al., 2022.

5.4 Our Addition

Existing suites give either (i) rich ground-truth graphs at a single sample size, or (ii) mixed real/synthetic data without verifiable causal structure. None deliver a *parametrically scalable* generator with (a) controllable linear and non-linear mechanisms, (b) explicit latent variables, and (c) built-in support for graduated sample-sizes so that researchers can compare sample-complexity across learners. Our contribution is to bridge this shortcoming while incorporating the anti-var-sortability safeguards highlighted above.

6 Notation and Setup: The Synthetic Data Generation Framework

Our synthetic data framework is designed to generate datasets with a known, but partially obscured, causal ground truth. This allows us to evaluate a model's ability to learn underlying causal relationships by observing its sample efficiency in approximating the true data generating process. The framework consists of three core components: a Directed Acyclic Graph (DAG) simulator with linear structural equations, a masking scheme to introduce latent variables, and a progressive sampling procedure.

6.1 Ground-Truth Causal Graph: Directed Acyclic Graph (DAG)

The foundation of our synthetic data is a Directed Acyclic Graph (DAG), denoted as $G = (V, E)$.

- $V = \{X_1, X_2, \ldots, X_D\}$ is a set of D nodes, representing random variables.
- E is a set of directed edges (X_i, X_j) between pairs of nodes, where $X_i \to X_j$ signifies that X_i is a direct causal parent of X_j.
- The graph G is acyclic, meaning there are no directed paths starting and ending at the same node. This ensures a valid topological ordering of the variables, $\pi = (X_{\pi(1)}, X_{\pi(2)}, \ldots, X_{\pi(D)})$, such that if $X_{\pi(i)} \to X_{\pi(j)}$, then $i < j$.
- For any node $X_j \in V$, we denote its set of direct parents in G as $PA_j = \{X_i \in V \mid (X_i, X_j) \in E\}$.

The DAG structure (i.e., the set E) is generated randomly for each instance, potentially controlled by parameters like graph density or maximum in-degree, to ensure a variety of causal structures.

6.2 Structural Equation Model (SEM)

Given the DAG G, we define a Structural Equation Model (SEM) where each variable X_j is determined by a linear function of its parents, plus an independent noise term: For $j = 1, \ldots, D$, following a topological order:

$$x_j = \sum_{X_k \in PA_j} \beta_{kj} x_k + u_j$$

where:

- β_{kj} are the structural (causal) coefficients representing the direct effect of X_k on X_j. To mitigate issues like "var-sortability" Reisach, Seiler, and Weichwald, 2021 and prevent trivial solutions, these coefficients are drawn randomly.
- u_j are mutually independent exogenous noise terms (or disturbances), i.e., $u_j \perp u_l$ for $j \neq l$.
- Each noise term u_j is independent of the parents of X_j, i.e., $u_j \perp PA_j$.
- The noise terms are drawn from distributions with zero mean, $\mathbb{E}[u_j] = 0$, and finite variance, $\mathrm{Var}(u_j) = \sigma_j^2$. To further counter statistical shortcuts, the noise distributions can be chosen from a family of non-Gaussian distributions (e.g., Uniform, Laplace, or mixture models) and their variances σ_j^2 can also be randomized (e.g., $\sigma_j^2 \sim \mathcal{U}(\sigma^2_{\min}, \sigma^2_{\max})$).

This specification ensures that the full joint distribution for $P(V)$ (over $V = \{X_1, \ldots, X_D\}$) is uniquely determined by the DAG structure G, the coefficient values $\{\beta_{kj}\}$, and the distributions of the noise terms $\{u_j\}$.

6.3 Masking Scheme: Introducing Latent Variables

A key feature of our framework is the introduction of latent variables by obscuring a subset of nodes from the learner. Let V be the set of all variables in the ground-truth DAG. We define:

- $V_L \subset V$: A configurable subset of nodes designated as *latent* (hidden). These variables are part of the true data generating process but are not observed by the learning algorithm. The choice of V_L can be based on specific criteria, such as node centrality, position in the topological sort, or random selection based on a masking probability p_m.
- $V_O = V \setminus V_L$: The set of *observed* variables, which are provided to the learner.

The learner's task is to model relationships between variables in V_O. For instance, the learner might be tasked to predict a specific target variable $Y \in V_O$ using a set of covariates $\boldsymbol{X}_{\text{obs}} \subseteq V_O \setminus \{Y\}$. The ground-truth conditional expectation to be learned is $g(\boldsymbol{x}_{\text{obs}}) = \mathbb{E}[Y|\boldsymbol{X}_{\text{obs}} = \boldsymbol{x}_{\text{obs}}]$. Due to the marginalization of variables in V_L that may be common causes, mediators, or ancestors of Y and $\boldsymbol{X}_{\text{obs}}$, the function $g(\boldsymbol{x}_{\text{obs}})$ can become a complex, non-linear function of $\boldsymbol{x}_{\text{obs}}$, even though the underlying structural equations of the full DAG are linear. This emergent non-linearity is the primary challenge posed to the learning algorithms.

For example, consider a simple case where Y has an observed parent $X_1 \in V_O$ and a latent parent $X_L \in V_L$. Further, X_L itself has an observed parent $X_2 \in V_O$. The structural equations might be:

$$\begin{aligned} x_2 &= u_2 \\ x_L &= \beta_{2L} x_2 + u_L \\ x_1 &= u_1 \\ y &= \beta_{1Y} x_1 + \beta_{LY} x_L + u_Y \end{aligned}$$

The learner observes X_1, X_2, Y. The target function is $\mathbb{E}[Y|X_1 = x_1, X_2 = x_2]$.

$$\begin{aligned} \mathbb{E}[Y|X_1 = x_1, X_2 = x_2] &= \mathbb{E}[\beta_{1Y} X_1 + \beta_{LY} X_L + u_Y | X_1 = x_1, X_2 = x_2] \\ &= \beta_{1Y} x_1 + \beta_{LY} \mathbb{E}[X_L | X_1 = x_1, X_2 = x_2] \\ &= \beta_{1Y} x_1 + \beta_{LY} \mathbb{E}[\beta_{2L} X_2 + u_L | X_1 = x_1, X_2 = x_2] \\ &= \beta_{1Y} x_1 + \beta_{LY} \beta_{2L} x_2 \end{aligned}$$

In this specific simple example, the resulting function remains linear. However, non-linearity in $g(\boldsymbol{x}_{\text{obs}})$ can arise if X_L influences multiple observed variables in $\boldsymbol{X}_{\text{obs}}$ and Y (acting as a confounder), or through more complex interaction pathways when several latent variables are involved. We operate under the premise that the combination of linear SEMs, specific DAG structures with appropriately chosen V_L, and non-Gaussian noise terms for the latent variables, which affects the form of $\mathbb{E}[X_L|\boldsymbol{X}_{\text{obs}} = \boldsymbol{x}_{\text{obs}}]$ when X_L is not simply a mediator as above, resulting in $g(\boldsymbol{x}_{\text{obs}})$ being sufficiently non-linear to robustly test model flexibility and causal learning.

Our proposed masking scheme for introducing latent variables is similar to the one proposed by Louizos et al., 2017 in *"Causal Effect Inference with Deep Latent-Variable Models"* though ours go beyond their set up with single latent confounder scenario, and expands on it significantly through our incorporation of other contributions.

6.4 Progressive Sampling Procedure

We use a progressive sampler similar to learning curve benchmarks such as AutoML Hestness et al. (2017). Where for each generated DAG instance and masking scheme, we generate a sequence of nested training datasets. Let S be a large dataset of $N_{\max}$ samples, $S = \{\boldsymbol{v}^{(i)}\}_{i=1}^{N_{\max}}$, where each $\boldsymbol{v}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})$ is an independent draw from the joint distribution $P(V)$ defined by the SEM.

From S, we create K nested training sets: $D_1 \subset D_2 \subset \cdots \subset D_K$. Each training set D_k consists of the first N_k samples from S, but only revealing the observed variables: $D_k = \{(\boldsymbol{x}_{\text{obs}}^{(i)}, y^{(i)})\}_{i=1}^{N_k}$, where $\boldsymbol{x}_{\text{obs}}^{(i)}$ are the realizations of variables in $\boldsymbol{X}_{\text{obs}} \subseteq V_O$ for sample i, and $y^{(i)}$ is the realization of a designated target variable $Y \in V_O$. The sample sizes are strictly increasing: $N_1 < N_2 < \cdots < N_K \leq N_{\max}$.

This progressive sampling allows us to evaluate models on continuously larger training sets, tracking their learning curves and identifying the minimum sample size N_k at which a model successfully generalizes to the true functional form $g(\boldsymbol{x}_{\text{obs}})$. Generalization is assessed on a separate, large, held-out test set drawn from the same data generating process, again only revealing observed variables.

7 Implementation

The following section details how we bridge the gap between the idealized theoretical framework and the practical realities or additions in the actual implementation.

The Full framework implementation as well as test scripts, are available via GitHub as an R package at https://github.com/Samuel99-2/SynthDAG

7.1 Core Data Generation in R

For the framework's core DAG generation capabilities, we rely on dagitty's library for the foundational representation and manipulation of DAGs. Textor et al., 2016

We begin by using a layered approach, hierarchically constructing a directed DAG. Nodes are distributed across a predefined number of layers, and edge formation is predominantly directed from nodes in earlier layers to those in subsequent ones. This inherently promotes acyclicity and provides control over the graph's hierarchical structure and causal flow. Parameters govern intra-layer and inter-layer connectivity probabilities, the potential designation and influence of "hub" nodes, and, crucially, we introduce a mechanism to ensure that nodes in the final layer, the outcomes, achieve a minimum number of parent connections. This latter feature guards against isolated outcome nodes, ensuring a baseline level of causal input.

Once a DAG structure is established, our DAG graph is populated by our SEM function when synthesizing data. This function orchestrates the creation of a linear Structural Equation Model (SEM) across the graph by assigning randomized structural coefficients as detailed in Sections 6.2 and 6.3. Our progressive sampler is currently implemented on an individual basis, with plans for implementing it in the overall tools available for our package.

7.2 Hub Nodes: Concentrating Causal Influence

To better reflect real-world systems, our framework incorporates "hub" nodes for the DAG generation phase. In most complex systems, certain factors act as strong predictors, in our case, these are termed "hub nodes". Hubs have extensive influence on downstream variables. For our framework, the number of hubs (specified by the parameter `num_hubs`) and their distribution across layers are controlled parameters. Hub placement is influenced by a hub diversity score, D_{hub}, which dictates the probability distribution of Hubs. A D_{hub} value of 0 restricts hubs to the first layer, while a value of 1 allows uniform selection across all layers. Intermediate values create a decaying probability for hub placement in later layers.

As we define hubs, it is an increased propensity for nodes to form outgoing connections. This factor can be controlled by a hub multiplier, which scales up the base edge formation probabilities, effectively increasing node reach and strength.

7.3 Noise Mechanisms for Robustness

To ensure that models are evaluated against varied and challenging conditions, our framework incorporates a flexible system for generating exogenous noise terms (u_j). The variance of each noise term, $\text{Var}(u_j)$, is randomly drawn from a pre-defined range $[\sigma^2_{\text{min}}, \sigma^2_{\text{max}}]$.

Beyond just using Gaussian noise, our implementation supports several distributions for u_j, including Uniform, Laplace, Gaussian Mixture Models and Discrete Mixture Models. This allows for the simulation of data where noise terms deviate significantly from normal Gaussian variance, reflecting a commonality of real-world datasets. The specific noise type for each node can be globally configured, individually specified using node-specific configs, or automatically diversified across a subset of nodes via an auto noise mechanism. This latter feature systematically introduces varied noise distributions.Titterington, Smith, and Makov (1985)McLachlan and Peel (2000)

For mixture models, the components are defined by their respective parameters [2]. This implementation ensures that the final generated noise term u_j, regardless of its underlying distribution type, is scaled to match the randomly selected target variance $\text{Var}(u_j)$ and a target mean[3]. This careful parameterization and diversification of noise structures helps ensure diversity of a learning models ability to differentiate between underlying data-generating processes beyond simple assumptions.

7.4 Benchmarking using Theoretical Maximum R-squared

To better evaluate models, we establish an asymptotic performance benchmark. This benchmark is the theoretical maximum R-squared (R^2_{max}), which signifies the proportion of variance in a designated target variable, $Y \in V_O$, that is inherently explainable by its true, direct causal parents as defined in the ground-truth SEM.

The theoretical maximum R-squared is defined as the ratio of the systematically explained variance to the total variance:

$$R^2_{\text{max}} = \frac{\text{Var}\left(\sum_{X_k \in PA_Y} \beta_{kY} X_k\right)}{\text{Var}(Y)} = \frac{\text{Var}(Y) - \text{Var}(u_Y)}{\text{Var}(Y)} = 1 - \frac{\text{Var}(u_Y)}{\text{Var}(Y)}$$

Within our framework, $\text{Var}(u_Y)$ is a pre-specified parameter of the data generating process, determined during the SEM setup. The total variance, $\text{Var}(Y)$, is practically estimated using the empirical variance of Y calculated from the complete dataset S (containing N_{max} samples) generated by the framework. This empirical estimate from a large N_{max} serves as a robust proxy for the true population variance of Y.

This R^2_{max} value is "theoretical" because it presupposes perfect knowledge of the true causal parents PA_Y and the true linear functional form of their influence on Y. It is "maximum" because no predictive model, regardless of its sophistication or the amount of data it is trained on, can account for the variance introduced by the irreducible noise term u_Y. [4]

Our definition of R^2_{max} is not fully original, and follows the variance decomposition logic of Gelman et al. (2019) and the “oracle” benchmarks common in synthetic Monte-Carlo studies (Parikh et al. (2022)). We see our implementation as natural extension of their idea, by embedding the metric in a DAG-based generator with latent masking that outputs R^2_{max} automatically for every dataset instance.

[2] i.e. means and standard deviations for Gaussian components, values and probabilities for discrete components
[3] Typically zero
[4] It could be noted that R^2_{max} is thus a fixed characteristic of the specific data-generating process.

7.5 Initialization

To isolate the marginal effect of a single graph-generation parameter on out-of-sample predictive accuracy, we employ a standard parameter design. For every replication, we first draw one fully specified data-generating graph using standardized structural coefficients, noise variances, latent-node mask, and a fixed train/test split, using a single master seed. Within that test we generate a series of treatment conditions by varying only a single parameter, while re-using the same underlying parameters for all other elements. Thus, across the treatment set the DAGs are identical except for the controlled modification and sampling variability. Thirty independent replications of this procedure yield unbiased estimates of the treatment effect and its standard error, while keeping variance to a minimum.

Table 1: Standard Parameters for Simulation Setup

Parameter	Value / Setting(s)	Role in Simulation Framework
Number of Replications	30	Number of independent simulation runs per configuration to ensure statistical robustness.
Learning Sizes	400	Training sample sizes used to construct learning curves and assess performance
Hub Levels	3	Varied number of hub nodes in DAGs to test model sensitivity to graph structure see figure 5 and 4.
Number of Nodes	20	Total number of variables (nodes) in each generated Directed Acyclic Graph.
Number of Layers	3	Number of hierarchical layers used to structure the DAGs, influencing causal depth.
Percent Mask	20%	Proportion of nodes designated as latent (unobserved), introducing confounding.
Fixed Test Size	150	Constant size of the hold-out test set used for model evaluation across all runs.
DAG Edge Probability	Intra: 0.15, Inter: 0.15	Base probabilities governing random edge formation within and across DAG layers.
Allow Skip Layers	Varied (typical False)	Boolean that controls if edges can span non-adjacent layers (e.g., TRUE for learning curves, FALSE for hub analysis).
Hub Diversity	0.5	Parameter (0 to 1) influencing the distribution of hub nodes across DAG layers.

Aditional parameters were set up, such as Master seed, hub multiplier score, etc. But they were omitted here for brevity. For the full setup, refer to our R repository available at https://github.com/Samuel99-2/SynthDAG

8 Testing our Framework

To validate the implemented framework and demonstrate its capabilities in generating diverse and challenging synthetic datasets, a series of robustness test were conducted. Our tests explore the impact of key framework parameters on data characteristics and the subsequent performance of common machine learning models. The following sub-sections present results from these evaluations, highlighting the framework's utility for systematic model benchmarking.

8.1 Robustness Test

Our framework capacity to produce varied marginal distributions beyond standard Gaussian forms was a primary concern. As seen in table 2 under "Marginal Kurtosis" baseline configurations with Gaussian noise yielded average kurtosis near zero, as expected. When employing heavy-tailed distributions such as Laplace, increased the average kurtosis to 2.170, demonstrating high degree of control over generating tail distribution. Our auto noise parameter setting successfully generated complex marginal distributions, as can be seen by figures 3 exhibiting clear bimodality.

The biggest challenge was inducing robustly non-linear relationships by latent variables of complex interactions. The framework's ability to create such non-linearities was assessed by comparing the fit of linear models against Generalized Additive Models (GAM). Baseline configurations with linear structural equations and no masking showed negligible GAM R^2 improvement as expected. Incorporating latent variables with mixture noise induced observable non-linearity, with an average GAM R^2 improvement of approximately 0.004. Similarly, "Regime Mixture Active (No Masking)" also showed potential for generating non-linear effects.

Table 2: Impact of Generator Settings on Synthetic Data Characteristics

Feature test	Value	Interpretation
Marginal Kurtosis (Avg. Kurtosis)		
Baseline Gaussian (No Masking)	0.000	Confirmed expected Gaussian-like distribution (low kurtosis) for baseline.
Baseline Laplace (No Masking)	2.170	Successfully generated heavy-tailed data, increasing realism by mimicking outliers.
Auto-Profile Diverse Noise	0.040	Auto-profile introduced some distributional diversity, though effect on avg. kurtosis is mild.
Non-Linearity (Avg. GAM R^2 Impr.)		
Baseline Gaussian (No Masking)	−0.001	Baseline shows minimal non-linearity in randomly selected observed pairs.
Latent Mixture Noise (1 Random Masked)	0.004	Latent variable with mixture noise induced observable non-linearity, enhancing complexity.
Regime Mixture Active (No Masking)	0.001	Regime mixture contributes a slight non-linear effect to observed relationships.
Heteroscedasticity (Prop. with BP p<0.05)		
Baseline Gaussian (No Masking)	0.000	Baseline relationships are predominantly homoscedastic as expected.
Low Connectivity DAG	0.330	Simpler DAG structures revealed underlying heteroscedastic patterns.
Regime Mixture Active (No Masking)	0.330	Regime mixture successfully induced heteroscedasticity in a notable proportion of pairs.
Correlation Strength (Median Abs. Corr.)		
Low Connectivity DAG	0.040	Confirmed ability to generate data with generally sparse/weak inter-variable correlations.
High Connectivity DAG (Many Hubs)	0.210	Demonstrated generation of more densely correlated data with stronger average relationships.
Outlier Generation (Avg. Outlier %)		
Baseline Gaussian (No Masking)	0.700	Baseline confirmed few outliers, consistent with Gaussian noise assumptions.
Baseline Laplace (No Masking)	4.300	Clearly generated a significantly higher proportion of outliers, enhancing data realism.

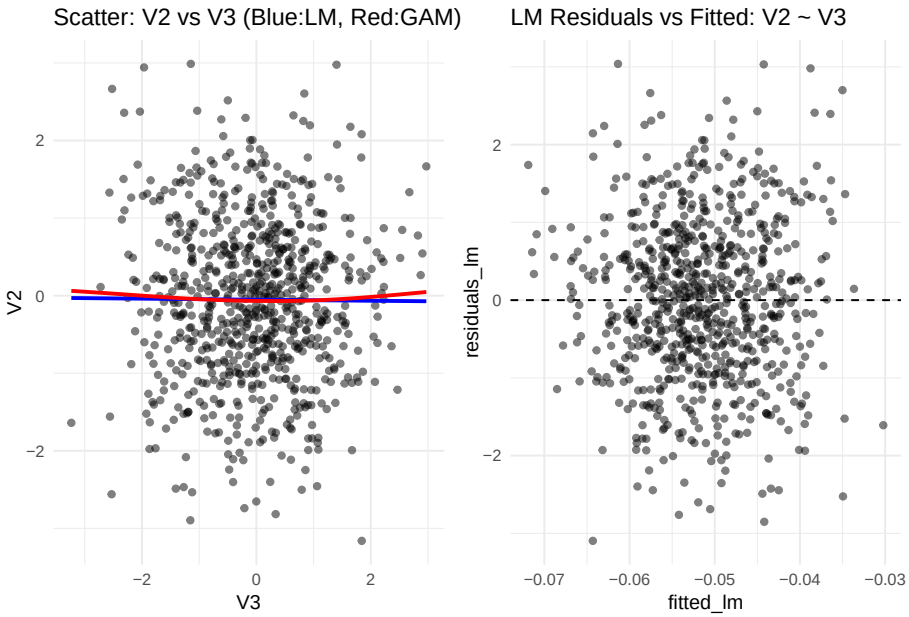

Figure 2: Residuals plotted with regime mixture enabled, showing weak nonlinearity compared to the base setup.

We confirmed that we have control over inter-variable correlation and heteroscedasticity generation, low connectivity DAGs produced data with generally sparse correlation, (Cor. ≈ 0.040), whereas high connectivity DAGs with hubs generated more densely correlated data (Cor. ≈ 0.210). Both "Low Connectivity DAG" and "Regime Mixture Active (No Masking)" configurations indicated a higher propensity for heteroscedasticity (Prop. with BP $p < 0.05 \approx 0.330$), showing the framework can model changes in variance.

Our outlier generation, as detailed under "Outlier Generation", is shown to be effectively managed by noise distribution choices. The "Baseline Laplace (No Masking)" setting produced a markedly higher average outlier percentage ($\approx 4.3\%$) compared to Gaussian baselines ($\approx 0.7\%$), enhancing data realism.

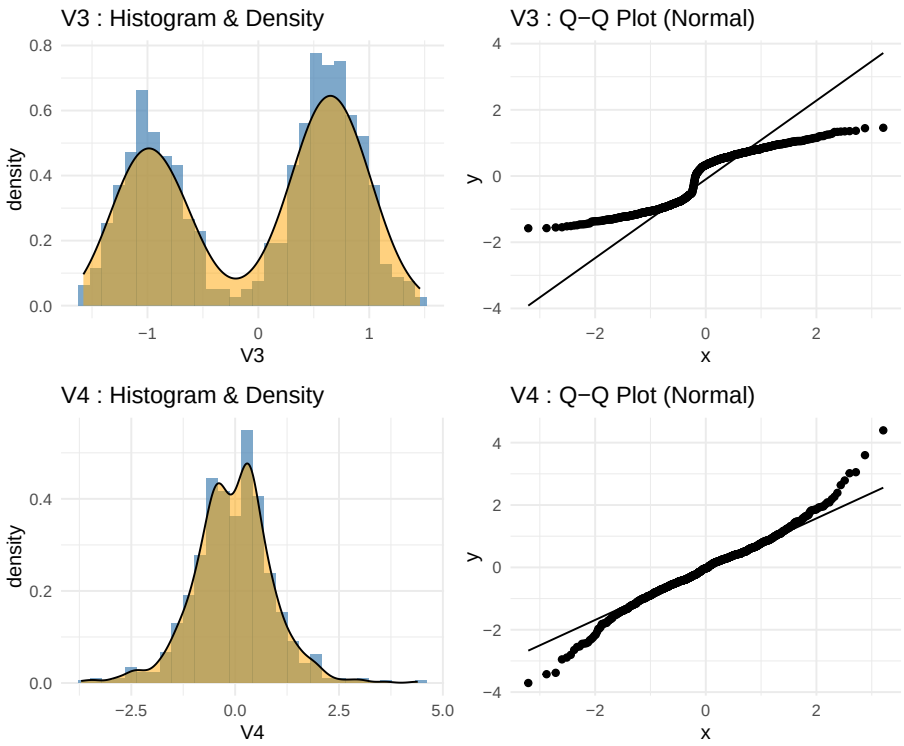

Figure 3: Graph illustrating bimodality of latent variables.

Hub influence on out-of-sample R2 values when skip layers is disabled.

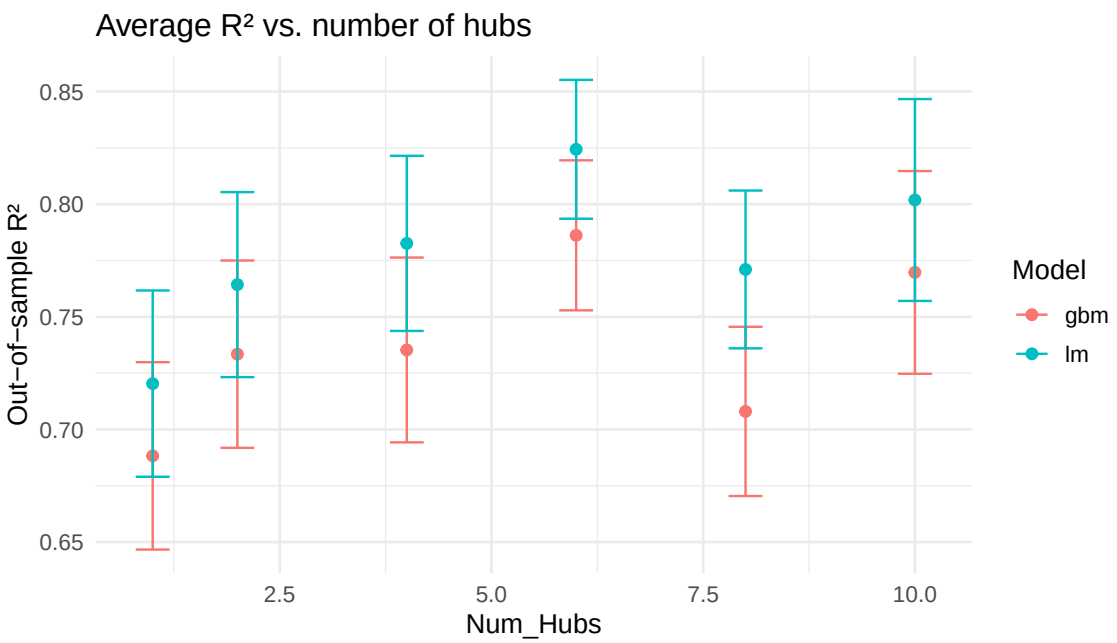

Figure 4: Standard set up, controlled seed generation, see 7.5

Hub influence on out-of-sample R2 values when skip layers is enabled.

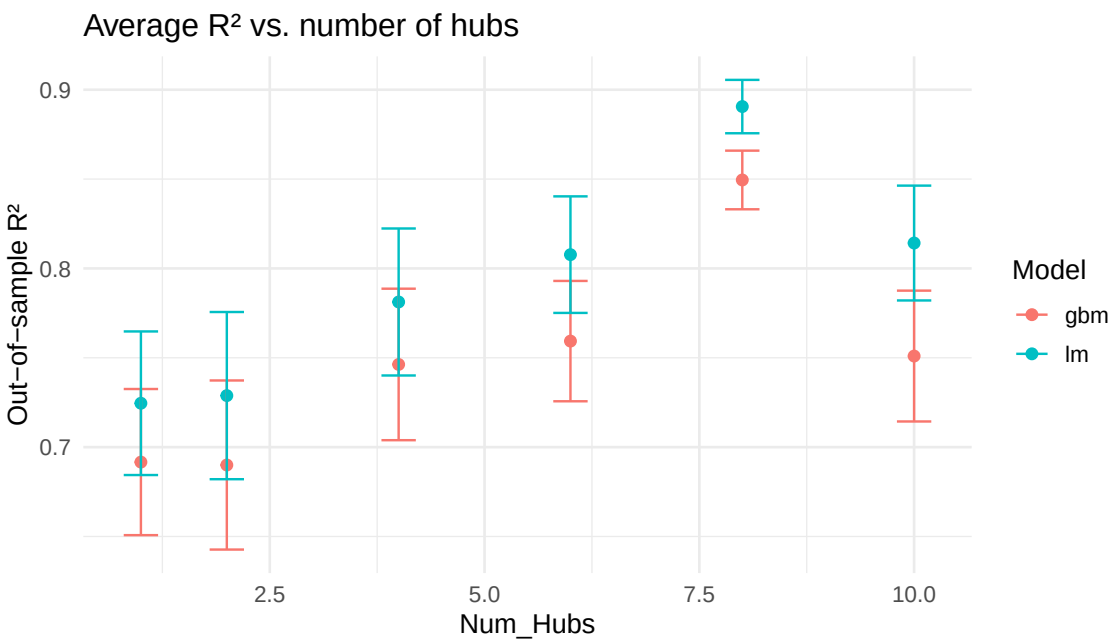

Figure 5: We see that the number of hubs has an outsized influence when skip layers is true

8.2 Model Performance

The performance of various machine learning models was evaluated using our synthetic data generation framework. This section assesses not only predictive accuracy (R-squared) but also how models learn under increasing data availability and complexity, their robustness to structural variations in the causal graph, and their ability to recover true underlying parameters. These evaluations are framed by our Central Hypothesis: "Models that attain asymptotic generalisation with the fewest samples exhibit the most faithful inductive bias and the strongest bias-correction capability."

Table 3: Summary of Coefficient Recovery: OLS Estimated vs. True SEM Coefficients

Target Variable	Parent Variable	True SEM Coeff.	LM Estimated Coeff.	Percent Diff. (%)
N20	N3	0.3677	0.3930	6.87
N11	N9	−0.7329	−0.7566	3.24
N19	N5	−0.5015	−0.5239	4.47
N19	N9	−0.5302	−0.7476	41.02
N15	N3	1.0480	1.1803	12.62

To test the first part of our hypothesis concerning sample efficiency and asymptotic generalisation, we tracked model performance across progressively larger nested training sets (Table 4 and

Figure 7). As shown in Table 4, linear models (ols, glmnet) and svmLinear rapidly approached a significant portion of the theoretical maximum R-squared (0.7045) even with smaller sample sizes (e.g., N=25 to N=40). For instance, ols achieved an R-squared of 0.5681 at N=40. Tree-based models, like gradient boosting, showed a more gradual learning curve, starting with lower (R-squared of 0.2180 for gbm at N=25, Target: N15) but demonstrating continued improvements, nearing the linear models often. Our support vector machine model consistently underperformed, indicating its inductive bias might be less suited for this particular setup.

Table 4: Comparison of R-squared Values for Different Models and Sample Sizes (Target: N15)

	Sample Size				
Model	10	25	40	55	70
Maximum theoretical R-squared			0.7045		
lm	−3.3079	0.4227	0.5681	0.5762	0.4919
glmnet	−1.3953	0.4443	0.5510	0.5258	0.4982
rf	0.0613	0.1382	0.2567	0.3511	0.3022
gbm	–	0.2180	0.4376	0.5139	0.4519
svmLinear	−0.9175	0.4399	0.5127	0.5607	0.4419
svmRadial	−0.0890	0.1646	0.2244	0.2485	0.1426

Another direct test of a model's "faithful inductive bias" towards the true data generating process is its ability to recover the true SEM coefficients. Table 3 (OLS Estimated vs. True SEM Coefficients) reveals that OLS, while capturing the sign and general magnitude, can exhibit notable percentage differences for specific coefficients (e.g., a 41.02% difference for N19 ← N9).

Table 5 provides a summary of coefficient recovery performance using Mean Absolute Error (MAE) for true parent coefficients. Lasso Showed the best performance with an Avg. MAE of 0.173 (Std. Err. 0.0567), outperforming OLS (Avg. MAE 0.192) and Ridge (Avg. MAE 0.241). This superior performance of Lasso in coefficient recovery, particularly in a system with underlying linear relationships and potential sparsity[5]. Strongly supports the idea that its inductive bias is more "faithful" to the true structural parameters, as such its ability to accurately estimate direct causal effects is a cornerstone of building causally-informed models.

Table 5: Average Coefficiency recovery scores

Model	Avg. MAE (True Parents)	Std. Err. MAE (True Parents)
Lasso	0.173	0.0567
OLS	0.192	0.0617
Ridge	0.241	0.0682

[5]which Lasso's penalty promotes

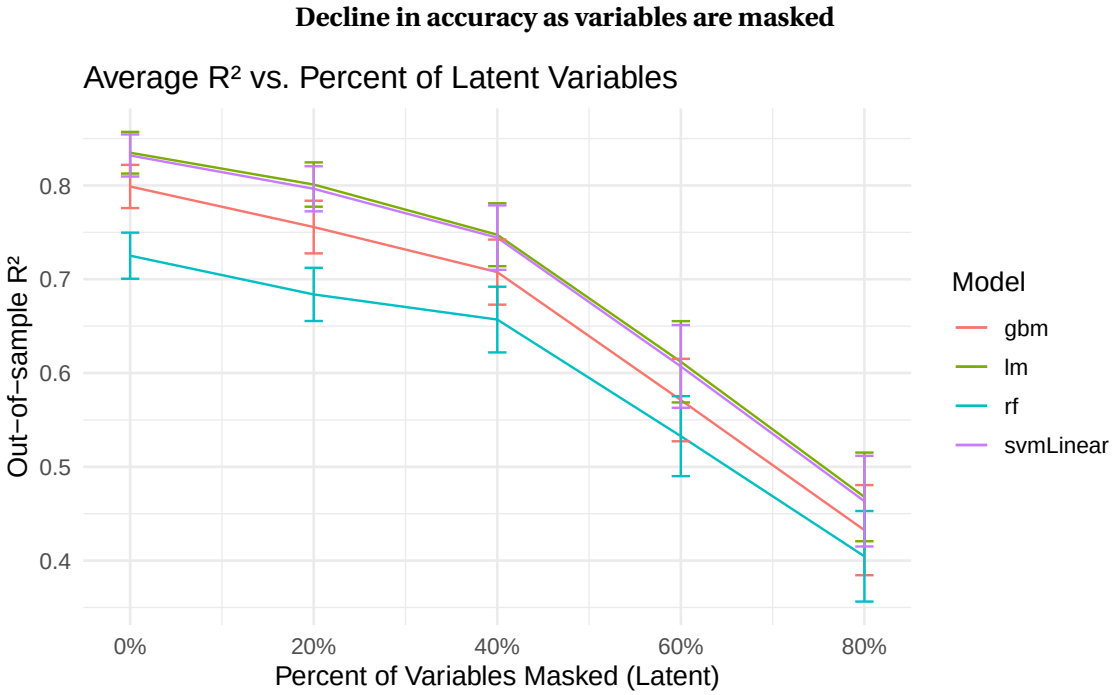

Figure 6: We see that increasing masking does lower the overall performance of models

Robustness to Structural Complexity and Latent Confounding

Our Masking allows for systematic variation of graph structure and the introduction of latent variables to test model robustness and, by extension, bias correction capability.

The influence of hub nodes varied significantly depending on whether "skip layers"[6] were permitted (Figures 4 and 5). In the case where skipping layers were disabled (Figure 4), the number of hubs did not significantly impact the accuracy of our models (Estimate: 0.0061, p-value: 0.0876). However, when skip layers were enabled (Figure 5), Number of hubs became a highly significant predictor, with an number of increasing hubs leading to higher accuracy of model trained on the dataset (Estimate: 0.0134, p-value: <0.0001). This suggests that when direct long range connections are possible, hubs become critical information conduits that even simpler models can exploit. The ability of a model to leverage such structural nuances reflects its capacity to adapt its learning to the underlying causal topology. For more on this see the discussion 9.1

As expected, increasing the percentage of masked, variables led to a quick decline in R-squared for all models (Figure 6).

[6]Skip layers allow for edges to connect across layers

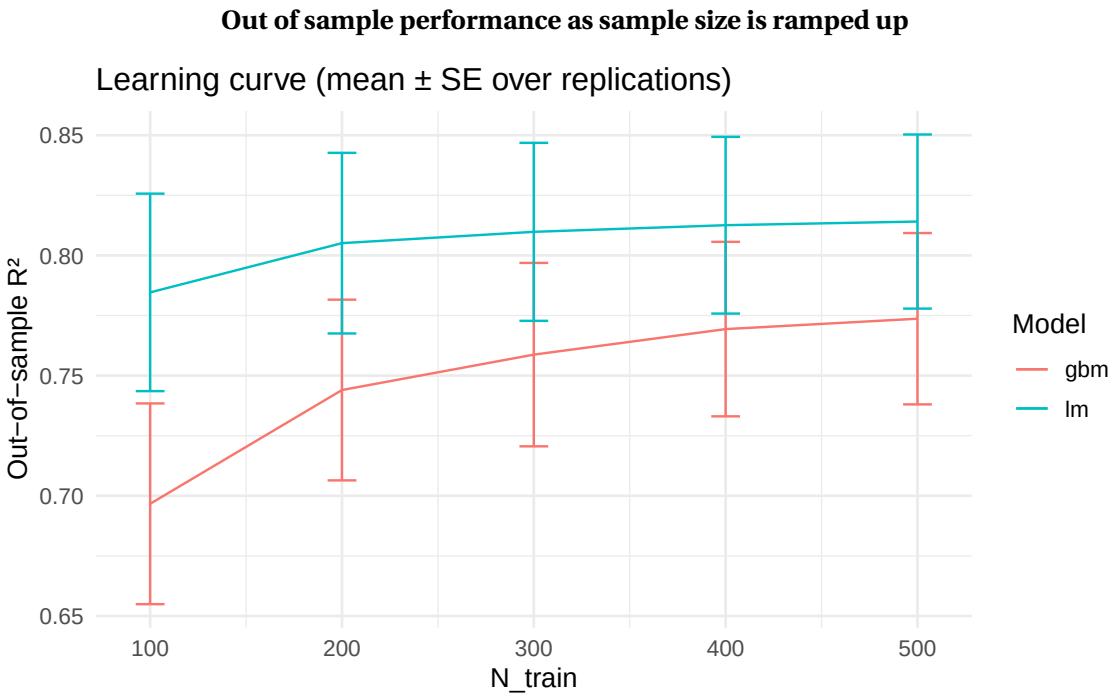

Figure 7: We see that the number of samples have clear influence on learning rate

9 Discussion

Having presented the empirical performance and robustness of our framework in the preceding section, this discussion covers primarily a deeper analysis of the short comings of our framework.

9.1 Links Between Hubs, Number of Layers, and Increases in Model Accuracy

The design of our synthetic data framework includes parameters that control the structure of the generated causal graphs. We observed that changing these parameters, specifically by decreasing the number of hierarchical layers or the number of hub nodes, can affect how well models can predict a target variable. See figures 4 and 5 Even if we keep the number of hidden latent nodes the same, a lower predictive accuracy, such as a lower R-squared value, is often seen with these simpler graph structures. This might seem unexpected if one thinks that larger graphs are always more complex. However, it makes sense when we consider how causal effects travel through a network and how information is spread or concentrated.

First, regarding layers, consider the fact that layers in a DAG help organize the direction of causal effects. When a graph has many layers, especially if edges can skip layers, there are many possible paths for an effect to travel from one node to another, including to the target node. These paths can be direct or go through several intermediate nodes. If we reduce the number of layers, for example, to just one, there are fewer of these long, indirect paths, skipping over nodes. In such a flatter graph, the effect of a parent node is more direct. At the same time, if a parent node of the target is hidden, there are fewer alternative paths through other observed nodes that could help predict the target. The observed nodes have fewer ways to act as substitutes for the information lost from the hidden parent. So, hiding a few key nodes in a graph with few layers can break important predictive connections more easily than in a graph with many layers. The system has thus a sort of lesser backup, or redundancy, in its causal pathways.

As also explained in the section 7 implementation, Hub nodes are designed to have many connections, mirroring how some variables in the real world affect many downstream factors. They either influence many other nodes or are influenced by many other nodes. As such, they can be said to gather or propagate causal links.

In a graph with many hubs, it is more likely that the target node itself, or its main parents, are hubs. If these important hubs are part of the observed nodes, they can provide strong information for prediction. Even if some parent hubs of the target are hidden, other observed nodes connected to these hidden hubs can act as good stand-ins. The many connections around hubs mean their influence often reaches several observable nodes.

When we use fewer hubs, the graph's connections become more evenly spread out. The chance that the target node or its important parent nodes are highly connected decreases. If a regular parent node (not a hub) becomes hidden, the remaining observed nodes might have weaker connections to this hidden cause. This makes them less effective as stand-ins. Less information "leaks" from the hidden variables to the observed ones through shared hub connections. The highest possible R squared for the target might also be lower, because the target's value depends on fewer, and possibly less strong, influencing parent nodes.

These effect are only strongly present when skipping of layers is visible, see figure 5. When disabled there is only a weak link $p = 0.08$, indicating that hubs lead to short cuts, figure 4.

9.2 Why did Linear Models Consistently Perform Better?

One thing we saw consistently throughout this report is that linear models such as support vector machines, OLS and Lasso were consistently out performing, more complex models. Our verdict is that because our SEM is fundamentally linear, there is a strong inductive bias towards linear relationships. If the true underlying relationships are linear, and thus our induced non-linearities in the observed data are weak or not highly complex. The outcome inevitably becomes that the linear models are more well-suite. We might have created a scenario where the non-linear models perform well for our in sample testing, but poorly on the out of sample, i.e as linear models are less prone to overfitting in such scenarios they in effect become the less biased variant.

Another factor we would argue, is that linear models, such as OLS, is often difficult to beat even when working with partially non linear dataMullainathan and Spiess, 2017. Linear models have high bias, they make strong assumptions about the data's structure, but low variance. If a given true relationship isn't wildly non-linear, the linear approximation, might still capture a significant portion of the signal. Meanwhile the low variance means they are less likely to overfit to the noise present in the data, especially with smaller sample sizes. Complex models, with their low bias and high variance, can easily fit the noise if our non-linear signal isn't strong enough or the dataset isn't large enough. As such, most real world phenomena, even if non-linear, can be reasonably well approximated by a linear relationship, especially over a limited range of the data.

Lastly, our induced apparent non-linearity, as shown in table 2, is at best only weakly present. A weak non linearity makes the general difficulty of beating linear models more pronounced. Models such as LASSO are likely capturing the dominant, still largely linear structure effectively. The minor subtle non linearities introduced by masking aren't yet strong enough for the complex models to gain a clear advantage. We expand on this in the following section below.

9.3 Connection to Non-Linearity and Model Tests

It is important to separate two ideas: the complexity of the relationship between observed variables (which in theory becomes non-linear when we hide nodes), and how high the R squared can actually get. A graph with fewer layers and hubs might result in a lower maximum R squared. This is not because the non-linear relationship is simpler. It is because less total predictive information about the target remains in the observed variables after some nodes are hidden.

This situation creates a harder test for learning models. A very high R squared (like over 0.95) might be easy to get in a graph with many connections and hubs, even if many nodes are hidden, because strong signals are still easy to find. But a graph with fewer connections and structural "shortcuts" might limit the best possible model to a lower R squared (like 0.7). In these more difficult cases, the differences in R squared between different models can tell us more about how well they can learn the non-linear relationships caused by hidden variables. They are not just finding strong, obvious signals. The task changes from fitting a strong signal to finding a weaker, more hidden one.

By changing parameters like the number of layers and hubs, our framework can create datasets where the challenge to the learner varies. It changes not just the non-linearity, but also the basic amount of signal versus noise that the model can see in the observed data. This helps us to better check how robust models are and how well they can learn causal relationships under different conditions.

9.4 Limits of our Framework and Possible Extensions

Perhaps one of the biggest weaknesses for our framework, is that we only succeed very weakly in establishing non linearity. It took significant feature engineering during DAG construction, introducing complex non-gaussian noise distributions and complex in node mask generation.

The emergence of significant non-linearity, hinges on a specific configuration within our synthetic data generation, one that only emerges with the current implementation sporadically. If we wish to extend the framework such that strong non-linearity is guaranteed, we stipulate that we need to modify our latent variable generation such that a latent variable, L, is guaranteed to simultaneously fulfill two conditions. First, any latent variable must be a common cause of a downstream predictor variable, that is visible X_{obs}, and it must also be linked to the outcome variable, Y_{obs}. Second the latent variable must possess a multi modal distribution, such as mixture models [7]. When these conditions are met, the process of marginalizing out the multi-modal latent common cause L can transform an underlying linear system into one where the conditional expectation $E[Y_{obs}|X_{obs}]$ exhibits non-linear behavior. Currently, this is possible within our current implementation, but only arises sporadically with base parameters.[8] We posit that a strong candidate for extending and improving our work is modifying the SEM such that these conditions are more consistently meet. Specifically our hub node, due to their designed characteristic of having extensive downstream influence, are strong candidates for this position. If a hub acts as such a multi-modal latent common cause, the resulting non-linearity in the observed variables is likely to be more pronounced and robust compared to a its non hub node counterpart. However, the principle applies to any node satisfying these criteria, regardless of its hub status.

[7] Non-gausian

[8] When we started off on this project, we knew that including noise terms with diverse draws, such as mixture models for the common cause, was necessary, as we also highlighted in the footnote of our initial proposal section. Though we significantly overestimated the likelihood that nonlinearity would appear as an emergent nature of the setup, we now see that a much more careful calibration was necessary to foster such conditions. Successfully implementing it to a satisfactory degree would go beyond the possible scope of a student project.

10 Conclusion

We have successfully implemented our synthetic DAG generation framework, designed to address the need for a robust, interpretable methods to evaluate how well machine learning models can learn underlying causal structures and generalize to true underlying causal effects in the data. Our SynthDAG framework offers a controlled environment for generating complex datasets with partially obscured, ground truths, thereby providing a more rigorous baseline for benchmarking models than typically available.

Throughout this project, we have demonstrated our SynthDAG framework's capabilities to:

- Systematically generate Directed Acyclic Graphs and corresponding data population via a linear Structural Equation Models.
- Introduces complexity by strategically creating latent nodes through masking, thereby mirroring artifacts of real world data
- We showed the strength of various ml learning curves through progressive sampling, allowing for the assessment of sample efficiency.
- We enabled the evaluation of a model's ability to recover true structural coefficients and performance against a theoretical maximum R-squared.

Our experimental evaluations yielded several findings. Consistently, simpler linear models (OLS, Lasso, svmLinear) demonstrated strong predictive performance, often outperforming more complex, non-linear models. This suggests that in our current setup, the induced non-linearity is not sufficiently pronounced to give complex models a decisive advantage, and linear models benefited from their inherent bias towards linearity and lower susceptibility to overfitting with limited data. We also observed the significant impact of structural graph parameters: the presence of "hub" nodes, especially when "skip layers" were enabled, substantially influenced model accuracy, highlighting how information concentration and direct pathways can be leveraged, potentially bypassing more complex structural learning. We also showed that masking predictably degraded performance across all models. From our model evaluation section we saw that Lasso was best performing when it came to recovering true SEM coefficients.

A pivotal insight from our investigation was the challenge in robustly and consistently inducing strong non-linear effects through the masking mechanisms. While the framework is designed to create non-linearity via masking, our tests indicated this effect, though present, were very subtle. As we detailed in Section 9.4, significant emergent non-linearity hinges on very specific configurations, such as a latent variable being a multi-modal common cause of both an observed predictor and the outcome, a condition our current implementation allows for sporadically, but does not guarantee.

We single out this weakness as an opening for our future work. A primary focus could be put on enhancing the framework's capacity to reliably generate stronger, more complex non-linearities. As proposed, modifying the SEM generation to ensure that designated latent variables, particularly hub nodes, are reinforced as multi-modal common causes would be a significant step towards creating more challenging and realistic testbeds for evaluating the non-linear learning capabilities of advanced ML models. Further extensions should involve incorporating a wider array of noise distributions, more sophisticated masking strategies, and testing a broader suite of causal discovery and prediction algorithms.

Our framework represents a valuable contribution towards more principled and causally-informed evaluation of machine learning models. We provide a flexible and transparent tool for generating synthetic data that mimics key aspects of real world causal systems, including latent confounding and emergent non-linearity. While we acknowledge the current limitations to the strength of our induced non-linearity, our framework offers a solid foundation, and a good path for future enhancements. We think this is a good first contribution, helping push for the development of ML models that are not merely predictively accurate but also more structurally sound and causally robust.

References

Cunningham, Scott (2021). *Causal Inference: The Mixtape*. Yale University Press. ISBN: 9780300251685. URL: https://mixtape.scunning.com/.

Geffner, Tomas et al. (2022). *Deep End-to-end Causal Inference*. arXiv preprint arXiv:2202.02195. URL: https://arxiv.org/abs/2202.02195.

Gelman, Andrew et al. (2019). „R-squared for Bayesian Regression Models“. In: *The American Statistician* 73.3, pp. 307–309. DOI: 10.1080/00031305.2018.1549100. URL: https://doi.org/10.1080/00031305.2018.1549100.

Hestness, Joel et al. (2017). *Deep Learning Scaling Is Predictable, Empirically*. arXiv preprint arXiv:1712.00409. URL: https://arxiv.org/abs/1712.00409.

Kaltenpoth, David and Jilles Vreeken (2023). „Nonlinear Causal Discovery with Latent Confounders“. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research, pp. 15639–15654.

Louizos, Christos et al. (2017). „Causal Effect Inference with Deep Latent-Variable Models“. In: *Advances in Neural Information Processing Systems (NIPS 2017)*, pp. 6446–6456. URL: https://arxiv.org/abs/1705.08821.

McLachlan, Geoffrey J. and David Peel (2000). *Finite Mixture Models*. New York: John Wiley & Sons. ISBN: 9780471006268.

Mullainathan, Sendhil and Jann Spiess (2017). „Machine Learning: An Applied Econometric Approach“. In: *Journal of Economic Perspectives* 31.2, pp. 87–106. DOI: 10.1257/jep.31.2.87.

Parikh, Anjali et al. (2022). *Oracle Benchmarks for Sample-Efficient Causal Structure Learning*. arXiv preprint arXiv:2203.14941. URL: https://arxiv.org/abs/2203.14941.

Pearl, Judea (2009). *Causality: Models, Reasoning, and Inference*. 2nd. Cambridge University Press.

Reisach, Alexander G., Christof Seiler, and Sebastian Weichwald (2021). „Beware of the Simulated DAGCausal Discovery Benchmarks May Be Easy to Game“. In: *Advances in Neural Information Processing Systems (NeurIPS 2021)*. URL: https://arxiv.org/abs/2102.13647.

Shimizu, Shohei et al. (2006). „A linear non-Gaussian acyclic model for causal discovery“. In: *Journal of Machine Learning Research* 7.Oct, pp. 2003–2030.

Textor, Johannes et al. (2016). „Robust Causal Inference Using Directed Acyclic Graphs: The R Package `dagitty`“. In: *International Journal of Epidemiology* 45.6, pp. 1887–1894. DOI: 10.1093/ije/dyw341. URL: https://academic.oup.com/ije/article/45/6/1887/2572602.

Titterington, D. M., A. F. M. Smith, and U. E. Makov (1985). *Statistical Analysis of Finite Mixture Distributions*. Chichester, UK: John Wiley & Sons. ISBN: 9780471912552.

11 Appendix

11.1 Detailed Directed Acyclic Graph Generation Methodology

The foundation of our framework, the Directed Acyclic Graphs (DAGs), $G = (V, E)$, are generated via a parametric algorithm, trying to maximize the diversity of the range of generated structures. Our Framework employs a layered construction process, which inherently ensures acyclicity by predominantly directing causal influences from nodes in earlier-assigned layers to those in subsequent layers. Such an approach gives deeper control over the graphs hierarchical organization and overall connectivity patterns.

The generation of each DAG instance is governed by a list key parameters[9]. We start of by defining the total number of nodes, D, and the number of hierarchical layers, L, across which these nodes are distributed, typically with an effort towards even distribution per layer. Edge formation (E) is then managed by simple probabilistic rules. The probability of an edge forming between two nodes situated within the same layer is denoted p_{intra}. To maintain acyclicity, these intra-layer connections are permitted only from a node to another that appears later in that layer's specific internal ordering. Similarly, a probability p_{inter} dictates the formation of edges between nodes in different layers, specifically from a node in an earlier layer L_i to one in a later layer L_j (where $j > i$). The nature of these inter-layer connections can be further refined by a boolean parameter, S_{skip}. If S_{skip} is true, edges are restricted to only immediately adjacent layers ($L_i \to L_{i+1}$), thereby creating explicit mediation stages. Alternatively, if false, edges are permitted to span multiple layers ($L_i \to L_{i+k}$ for $k > 1$), allowing for more direct, long-range causal effects.

To emulate the varying degrees of influence variables exhibit in real complex systems, our framework incorporates "hub" nodes. We start of by specifying a number of nodes, N_{hubs}, as designated hubs, intended to possess a broader connectivity, or "causal reach". The distribution of these N_{hubs} across the L available layers can be adjusted by the hub diversity parameter, $D_{hub} \in [0, 1]$. A D_{hub} value of 0 limits hubs within the first layer, promoting strictly early sources of wide influence. $D_{hub} = 1$ results in hubs being selected with a uniform probability from nodes across layers. Intermediate values of D_{hub} induce a probabilistically decaying propensity for hub placement in subsequent layers, implemented via a weighted sampling based on the layer-specific propensities. The influence of these designated hub nodes is subsequently amplified by an outward degree multiplier, M_{hub}. This parameter scales the base probabilities of edge formation (p_{intra} and p_{inter}) for any outgoing edge originating from a hub node, although this amplified probability is typically capped (i.e at 0.99) to maintain stability. Optionally, a parameter $O_{max,hub}$ can impose an absolute upper limit on the total number of outgoing edges from any single hub, constraining its maximum potential influence.

Last, to ensure that nodes representing potential final outcomes in the causal process are not isolated as dead nodes, our framework includes a mechanism to guarantee a minimum level of connectivity. For nodes situated in the final layer of the DAG, if their number of direct parents falls below the specified minimum, $P_{min,outcome}$, additional parent edges are selectively introduced from eligible nodes located in strictly earlier layers.

[9] See Cran package for specific documentation

11.2 Additional plots and graphs

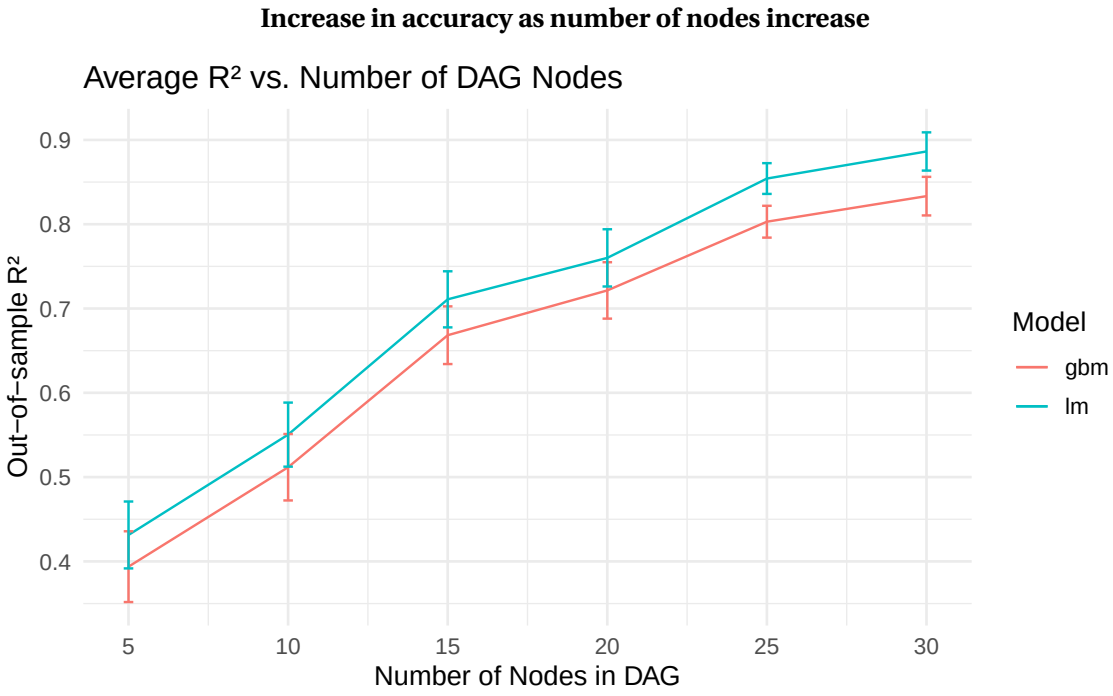

Figure 8: Perhaps counterintuitively, increasing the number of nodes increases accuracy. We observed that if a number of nodes is scaled up without also scaling the number of layers, the models get continuously better at prediction.

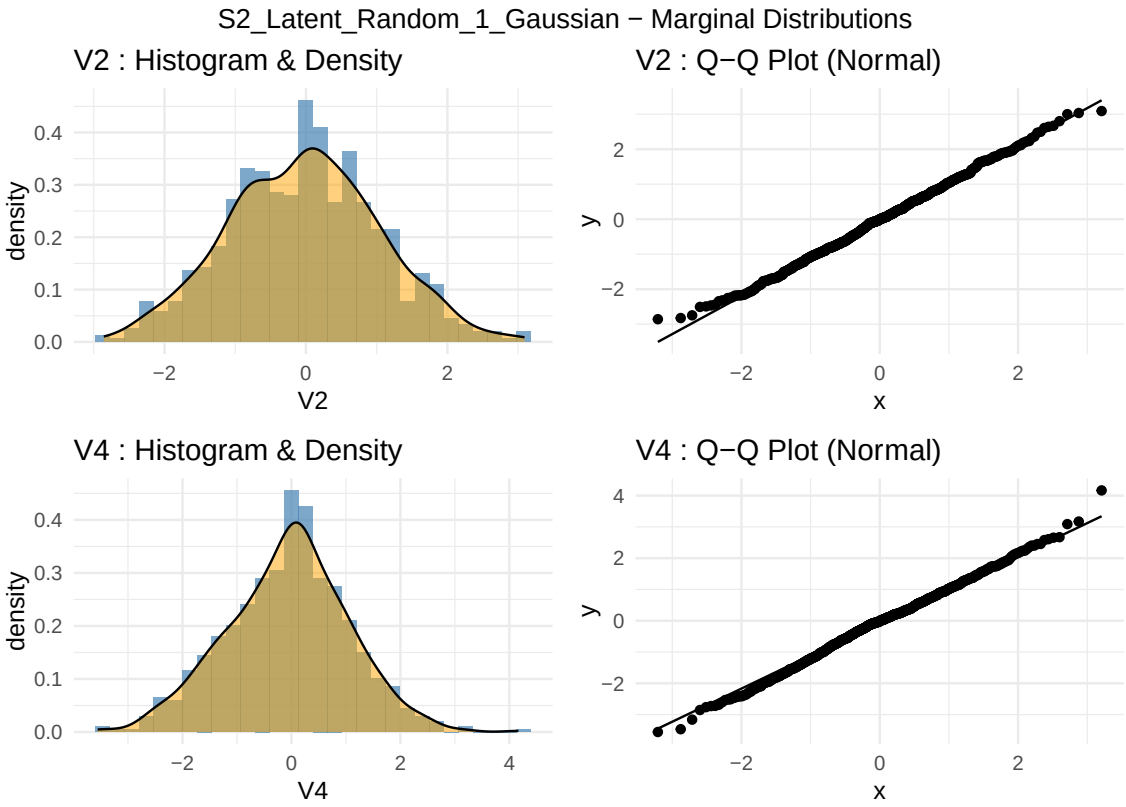
S2_Latent_Random_1_Gaussian − Marginal Distributions
V2 : Histogram & Density
V2 : Q−Q Plot (Normal)
V4 : Histogram & Density
V4 : Q−Q Plot (Normal)
density
V2
V4
x
y

Figure 9

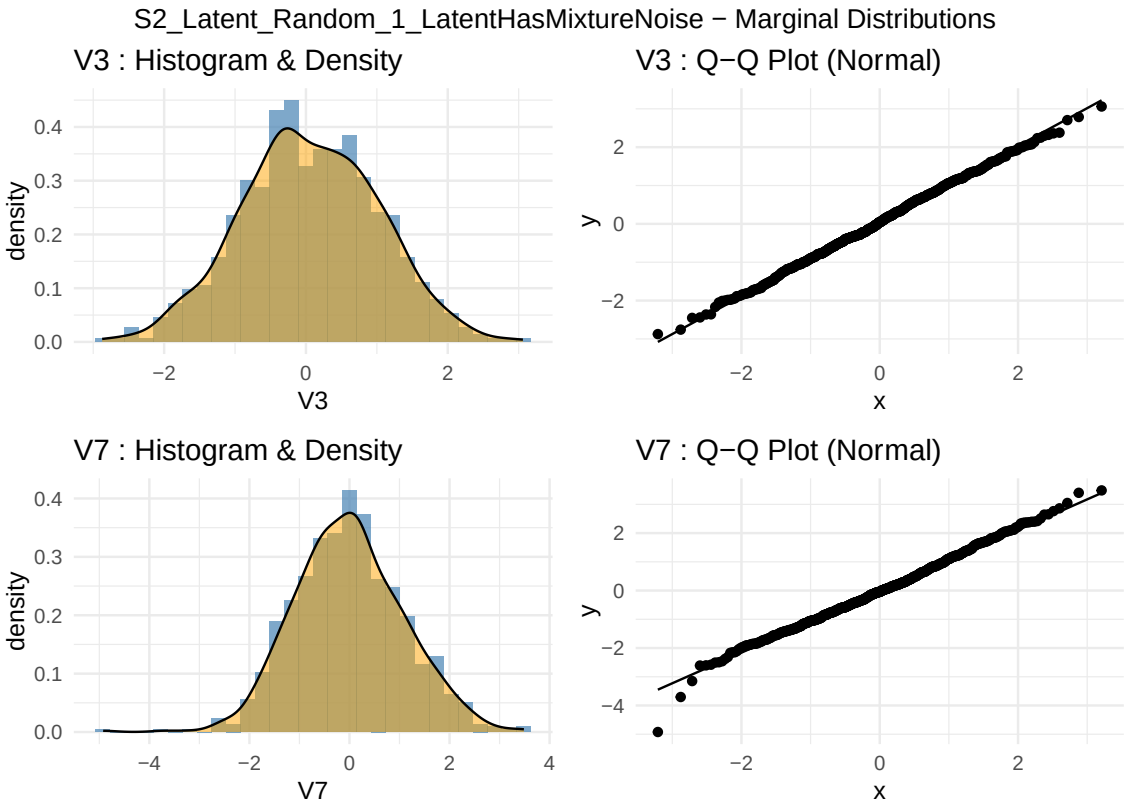
S2_Latent_Random_1_LatentHasMixtureNoise − Marginal Distributions
V3 : Histogram & Density
V3 : Q−Q Plot (Normal)
density
V3
y
x
V7 : Histogram & Density
V7 : Q−Q Plot (Normal)
density
V7
y
x

Figure 10

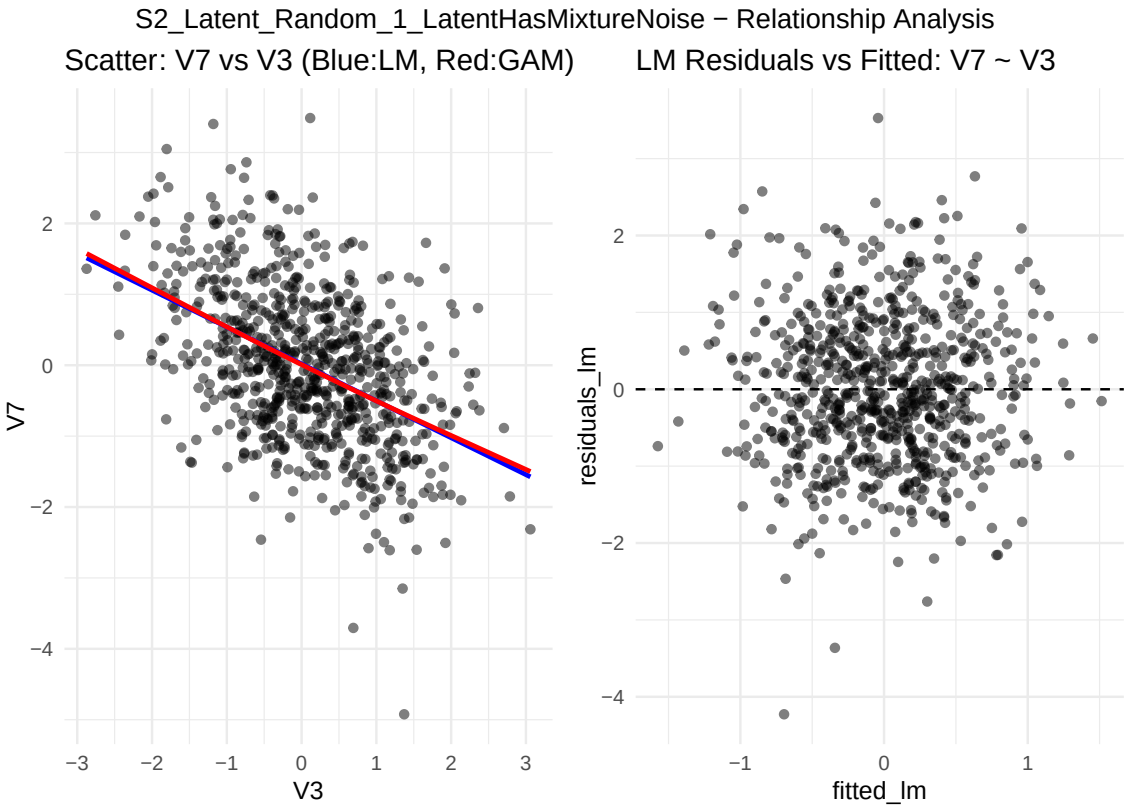
S2_Latent_Random_1_LatentHasMixtureNoise − Relationship Analysis
Scatter: V7 vs V3 (Blue:LM, Red:GAM)
LM Residuals vs Fitted: V7 ~ V3
V7
V3
residuals_lm
fitted_lm

Figure 11

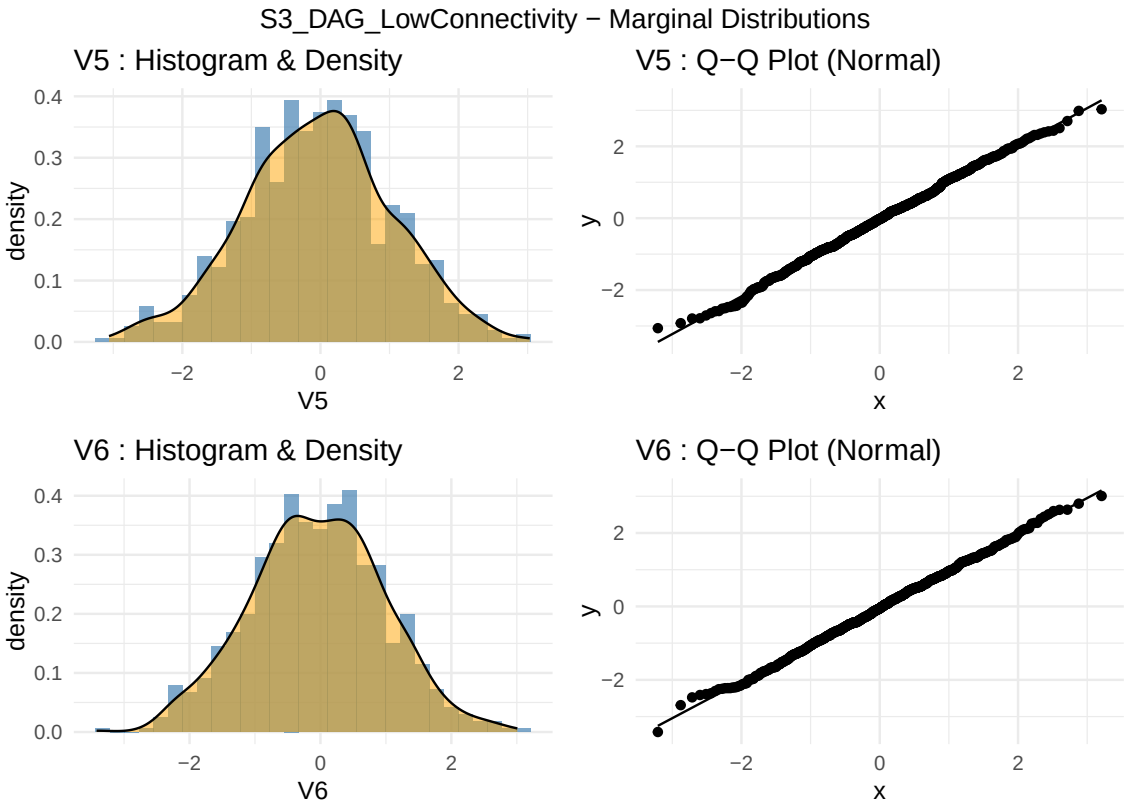
S3_DAG_LowConnectivity − Marginal Distributions
V5 : Histogram & Density
V5 : Q−Q Plot (Normal)
V6 : Histogram & Density
V6 : Q−Q Plot (Normal)
density
V5
V6
x
y
0.0
0.1
0.2
0.3
0.4
−2
0
2

Figure 12

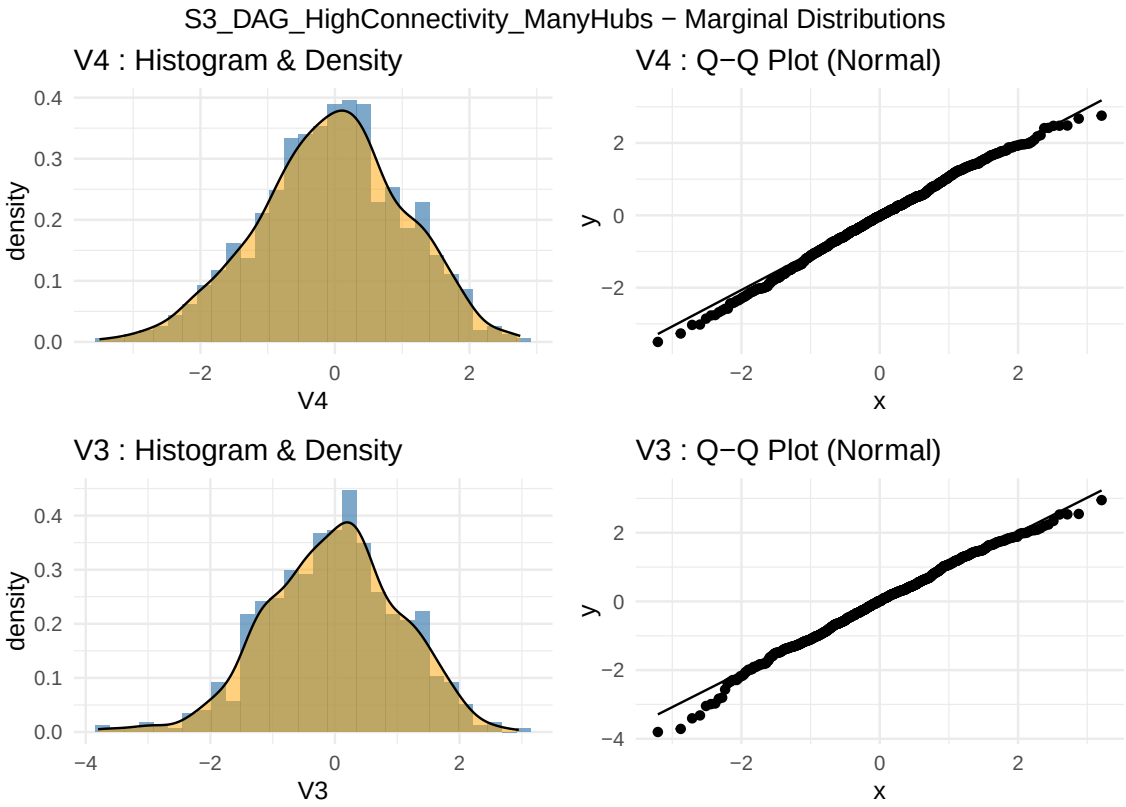
S3_DAG_HighConnectivity_ManyHubs − Marginal Distributions
V4 : Histogram & Density
V4 : Q−Q Plot (Normal)
V3 : Histogram & Density
V3 : Q−Q Plot (Normal)
density
V4
V3
x
y

Figure 13

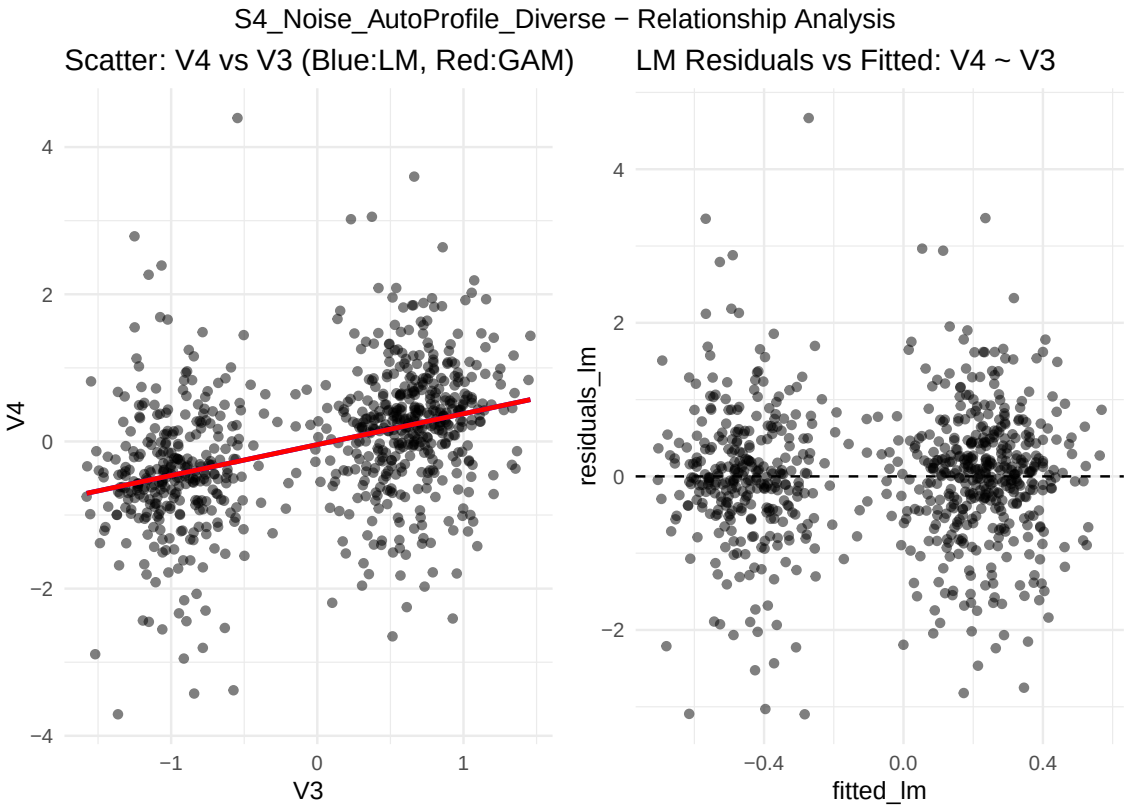
S4_Noise_AutoProfile_Diverse − Relationship Analysis
Scatter: V4 vs V3 (Blue:LM, Red:GAM)
LM Residuals vs Fitted: V4 ~ V3
V4
V3
residuals_lm
fitted_lm

Figure 14

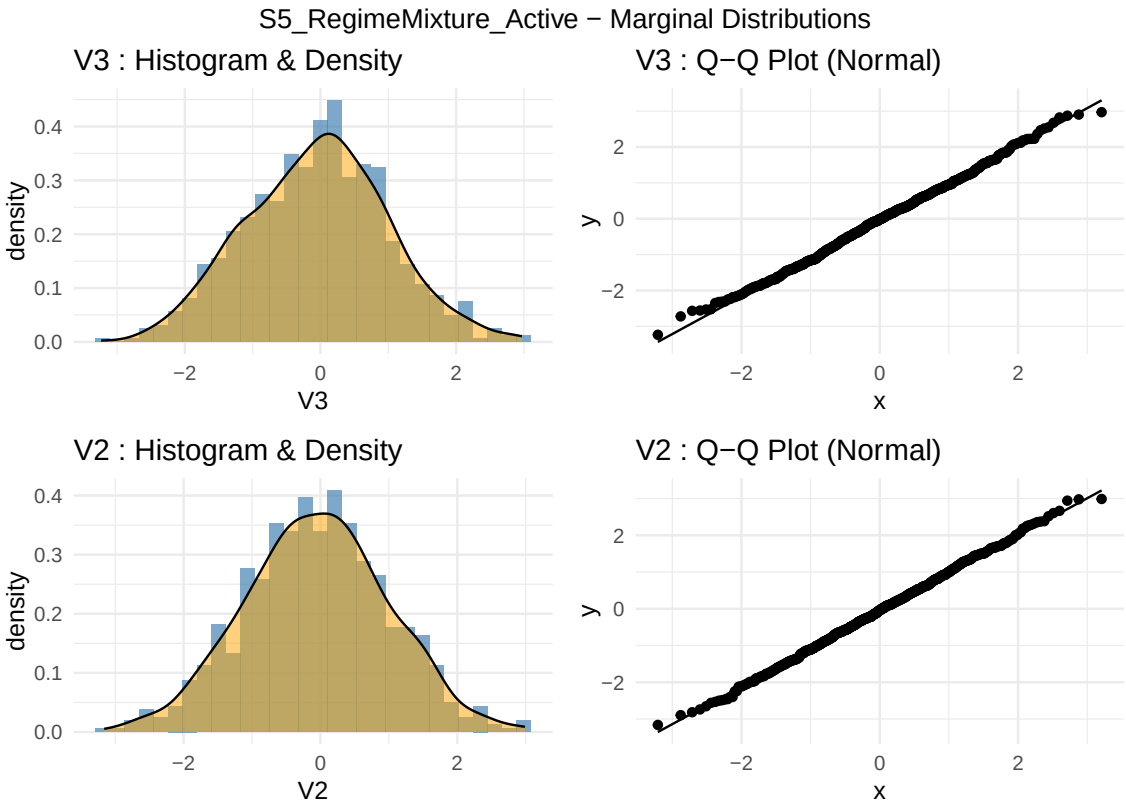
S5_RegimeMixture_Active − Marginal Distributions
V3 : Histogram & Density
V3 : Q−Q Plot (Normal)
density
V3
y
x
V2 : Histogram & Density
V2 : Q−Q Plot (Normal)
density
V2
y
x

Figure 15

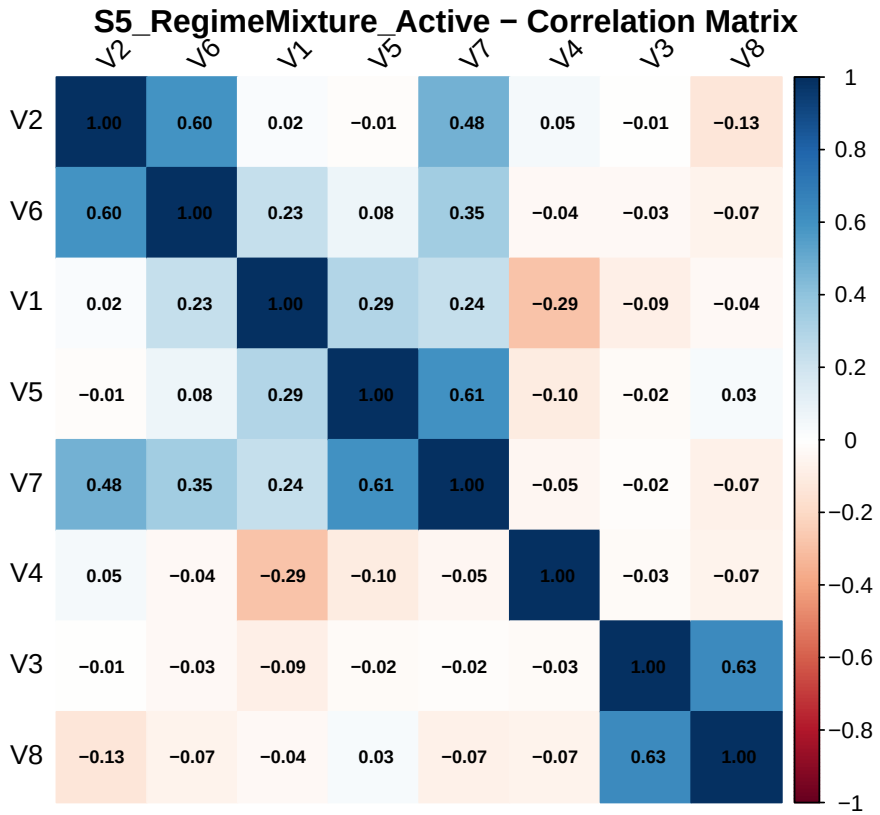
S5_RegimeMixture_Active − Correlation Matrix
V2 V6 V1 V5 V7 V4 V3 V8
V2 1.00 0.60 0.02 −0.01 0.48 0.05 −0.01 −0.13
V6 0.60 1.00 0.23 0.08 0.35 −0.04 −0.03 −0.07
V1 0.02 0.23 1.00 0.29 0.24 −0.29 −0.09 −0.04
V5 −0.01 0.08 0.29 1.00 0.61 −0.10 −0.02 0.03
V7 0.48 0.35 0.24 0.61 1.00 −0.05 −0.02 −0.07
V4 0.05 −0.04 −0.29 −0.10 −0.05 1.00 −0.03 −0.07
V3 −0.01 −0.03 −0.09 −0.02 −0.02 −0.03 1.00 0.63
V8 −0.13 −0.07 −0.04 0.03 −0.07 −0.07 0.63 1.00
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Figure 16

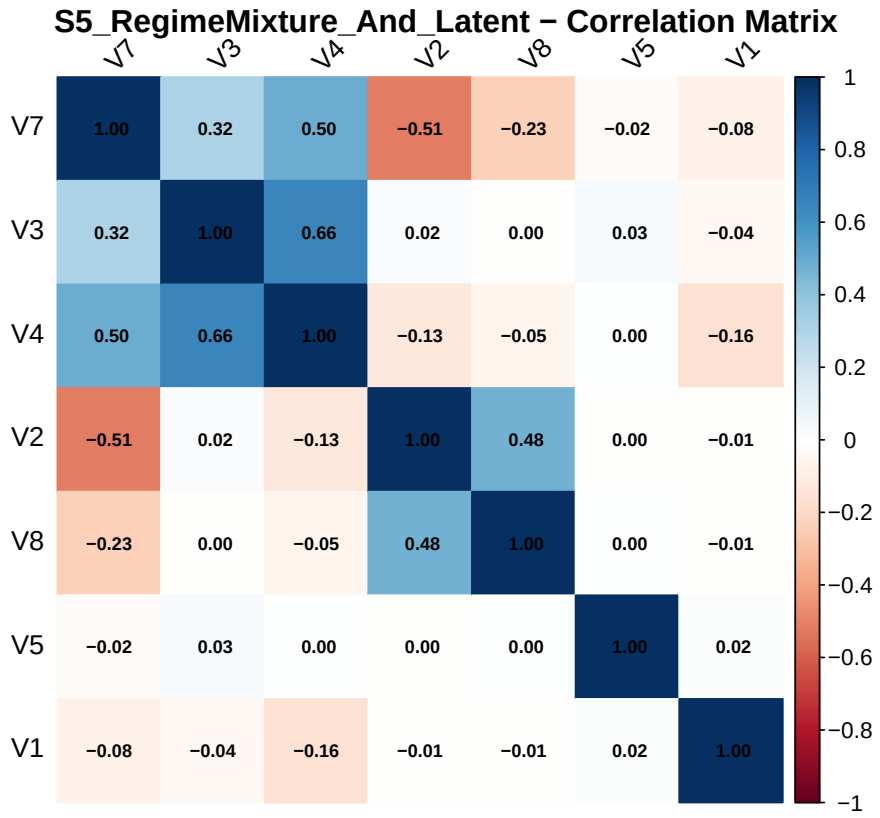
S5_RegimeMixture_And_Latent − Correlation Matrix
V7 V3 V4 V2 V8 V5 V1
V7 1.00 0.32 0.50 −0.51 −0.23 −0.02 −0.08
V3 0.32 1.00 0.66 0.02 0.00 0.03 −0.04
V4 0.50 0.66 1.00 −0.13 −0.05 0.00 −0.16
V2 −0.51 0.02 −0.13 1.00 0.48 0.00 −0.01
V8 −0.23 0.00 −0.05 0.48 1.00 0.00 −0.01
V5 −0.02 0.03 0.00 0.00 0.00 1.00 0.02
V1 −0.08 −0.04 −0.16 −0.01 −0.01 0.02 1.00
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Figure 17