Exploring Neural Granger Causality with xLSTMs: Unveiling Temporal Dependencies in Complex Data

Harsh Poonia[1,*] **Felix Divo**[2,*] **Kristian Kersting**[2,3,4,5] **Devendra Singh Dhami**[6]

[1]Carnegie Mellon University [2]AI & ML Group, TU Darmstadt [3]hessian.AI
[4]DFKI [5]Centre for Cognitive Science, TU Darmstadt [6]TU Eindhoven

hpoonia@cs.cmu.edu d.s.dhami@tue.nl
{felix.divo,kersting}@cs.tu-darmstadt.de

Abstract

Causality in time series can be challenging to determine, especially in the presence of non-linear dependencies. Granger causality helps analyze potential relationships between variables, thereby offering a method to determine whether one time series can predict—Granger cause—future values of another. Although successful, Granger causal methods still struggle with capturing long-range relations between variables. To this end, we leverage the recently successful Extended Long Short-Term Memory (xLSTM) architecture and propose Granger causal xLSTMs (GC-xLSTM). It first enforces sparsity between the time series components by using a novel dynamic loss penalty on the initial projection. Specifically, we adaptively improve the model and identify sparsity candidates. Our joint optimization procedure then ensures that the Granger causal relations are recovered robustly. Our experimental evaluation on six diverse datasets demonstrates the overall efficacy of GC-xLSTM.

1 Introduction

Finding cause and effect among and within a group of multivariate time series can lead to a better understanding of the dynamics of the involved time series. For instance, in computational neuroscience and medicine, discovering brain connectivity assists in better understanding natural cognition [Smith et al., 2011]. Discovering inter-dependencies between time series also has a critical impact on many other research areas, such as finance [Masini et al., 2023], climate science [Mudelsee, 2019], and industrial applications [Strem et al., 2025]. Although efforts have recently been made to improve the interpretability of time series models [Ismail et al., 2020, Turbé et al., 2023], most methods are restricted to finding post-hoc interpretations and only focus on short-term dependencies.

The framework of Granger causality (GC) [Granger, 1969] was introduced to address the challenge of determining whether one variable's past values can help forecast another's future values, without implying direct causality. GC can be established using statistical hypothesis tests, determining whether one time series can predict another. The test traditionally involves estimating a vector autoregressive model and examining whether lagged values of a time series improve or degrade the prediction of the other, while controlling the past behavior of both series. Although GC does not imply a direct cause-and-effect relationship between the involved time series [Heckman, 2008], recognizing these interdependencies can lead to a better understanding of the dynamic relationships between variables over time [Marcinkevičs and Vogt, 2021, Shojaie and Fox, 2022].

[*]Authors contributed equally.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Many families of deep learning architectures have been explored for time series analysis over the years, such as multilayer perceptrons [Zeng et al., 2023, Das et al., 2023], recurrent neural networks (RNNs) [Hochreiter and Schmidhuber, 1997, Cho et al., 2014], convolutional neural networks [Wu et al., 2022, Wang et al., 2022], Transformers [Vaswani et al., 2017, Nie et al., 2023], state-space models (SSMs) [Wang et al., 2025], or mixing architectures [Wang et al., 2024]. Throughout this, recurrent models remained a natural choice for time series data since their direction of computation aligns well with the forward flow of time. This aligns particularly well with the goals of neural Granger causality. Although SSMs are comparable in that regard, RNNs tend to offer more powerful forecasting capabilities since they deteriorate less when modeling long-term dependencies. Furthermore, their inference runtime is typically linear in the sequence length at constant memory cost, making them much more efficient than, for instance, Transformers with quadratic runtimes and memory requirements, while remaining highly expressive. Recently, Beck et al. [2024] revisited recurrent models by borrowing insights gained from Transformers in many domains, specifically natural language processing. Their proposed Extended Long Short-Term Memory (xLSTM) model sparked a resurgence of interest in recurrent architectures for sequence modeling and has already proven highly suitable for time series forecasting [Kraus et al., 2025, Alharthi and Mahmood, 2024].

Although most Granger causal machine learning methods assume linearity in time series as a fundamental prerequisite [Siggiridou and Kugiumtzis, 2015, Zhang et al., 2020], recent efforts capture non-linear dynamics in time series by using neural networks as the modeling choice instead of VARs [Tank et al., 2022, Löwe et al., 2022, Cheng et al., 2024]. Although successful, these non-linear methods require careful feature engineering to include time-based patterns. Thus, they may not capture interactions between time series and external factors as effectively as xLSTMs, which can learn non-linear patterns and adapt to the non-stationary nature of time series data.

We introduce *GC-xLSTM*, a novel method that leverages xLSTMs to uncover the GC relations in the presence of complex data, which inherently can have long-range dependencies.[1] GC-xLSTM first enforces sparsity between the time series components by using a novel lasso penalty on the initial projection layer of the xLSTM. We learn a weight per time series and then adapt them to find the relevant variates for that step. Then, each time series component is modeled using a separate xLSTM model, enabling us to discover interpretable GC relationships between the time series variables. After the forecast results by the individual xLSTM models, the important features are made more prominent, whereas the less important ones are diminished by a joint optimization technique, which includes using a novel reduction coefficient. Thus, the overall GC-xLSTM model can be trained end-to-end to uncover long-range Granger causal relations.

Our main research **contributions** can be summarized as follows:

(i) We propose GC-xLSTM, a novel model that can uncover Granger causal relations in non-linear time series.

(ii) Our novel algorithm jointly improves the forecasting model while adaptively enforcing strict sparsity.

(iii) Our empirical evaluations demonstrate that GC-xLSTM can robustly discover Granger causal relations in the presence of complex simulated and real-world data.

Outline. We start by recalling preliminaries and reviewing related research to contextualize this work in the broader body of research on neural Granger causality in Section 2. This allows us to introduce GC-xLSTM in Section 3 and empirically evaluate in relation to other methods in Section 4. Finally, we conclude with an outlook to future work in Section 5.

2 Preliminaries and Related Work

We are interested in datasets of strictly stationary time series $\boldsymbol{S} \in \mathbb{R}^{V \times T}$ of V variates with length T. Let $\boldsymbol{S}_t \in \mathbb{R}^V$ denote the value of $\boldsymbol{S}$ at time t. A variate v (sometimes called a channel) can be any scalar measurement, such as the chlorophyll content of a plant or the spatial location of some object being tracked. Its value at time t is $S_{v,t} \in \mathbb{R}$. The measurements are assumed to be carried out jointly at T regularly spaced time steps. In forecasting, a model is presented with a time series of C context steps $\boldsymbol{S}_{<t}$ before t, from which it shall predict the next value $\boldsymbol{S}_t \in \mathbb{R}^V$.

[1] Code available at github.com/harpoonix/GC-xLSTM.

2.1 Granger Causality

If the observed time series were generated by some underlying process g, which we can formalize as a structural equation model for all time steps t as $S_{v,t} = g_v(\boldsymbol{S}_{1,<t}, \ldots, \boldsymbol{S}_{V,<t}) + \epsilon_{v,t}$ for all $v \in \mathcal{V}$, where $\epsilon_{v,t}$ is some additive zero mean noise independent from all variates $\mathbf{S}_{v,<t}$ and $\mathcal{V} := \{1, \ldots, V\}$ is the set of all variates. In Granger causality [Granger, 1969], we aim to determine whether past values $\boldsymbol{S}_{v,<t}$ of a variate v are predictive for future values $\boldsymbol{S}_{w,\geq t}$ of another variate w. Following the notation of Shojaie and Fox [2022], we formally define:

Definition 1 (Granger Causality)**.** *Variate v is* Granger non-causal *for w if and only if g_v is invariant to $\boldsymbol{S}_{w,<t}$ for all $t \in \{1, \ldots, T\}$, i.e., if and only if*

$$g_v(\boldsymbol{S}_{1,<t}, \ldots, \boldsymbol{S}_{V,<t}) = g_v(\boldsymbol{S}_{1,<t}, \ldots, \boldsymbol{S}_{V,<t} \setminus \boldsymbol{S}_{w,<t}).$$

Else, we call v Granger causal *for w.*

The set of all such relationships are the directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of the *Granger causal graph* $(\mathcal{V}, \mathcal{E})$ of the variates, which we eventually aim to uncover.

2.2 Neural Granger Causality

Unfortunately, however, we cannot explicitly access g in most realistic settings. Using machine learning methods, we can nonetheless estimate each of the V process components g_v by an autoregressive time series forecasting model $\mathcal{M}_{\boldsymbol{\theta},v}(\boldsymbol{S}_{<t}) \approx g_v(\boldsymbol{S}_{<t}) = \boldsymbol{S}_{v,t} - \epsilon_{v,t}$. We can do so by estimating the parameters $\boldsymbol{\theta}$ of the model based on the dataset of time series, minimizing the predictive mean squared error (MSE) loss

$$\mathcal{L}_{\text{pred}}(\boldsymbol{\theta}) = \sum_{v=1}^{V} \sum_{t=1}^{T} \left(S_{v,t} - \mathcal{M}_{\boldsymbol{\theta},v}(\boldsymbol{S}_{<t})\right)^2. \tag{1}$$

In the case of using neural networks for $\mathcal{M}_{\boldsymbol{\theta},v}$, this approach is called *Neural Granger Causality* [Tank et al., 2022].

It would be very costly to train a total of V^2 models to test if each variate Granger causes any other variate and thus construct the entire Granger causal graph. To avoid this, we can train merely a single *component-wise* model $\mathcal{M}_{\boldsymbol{\theta},v}$ for each variate v and inspect what inputs w it is sensitive to. While this does not ablate models as in classical GC, it reduces the number of models to be trained from quadratic to linear in V. This can, for instance, be achieved by optimizing the predictive loss $\mathcal{L}_{\text{pred}}(\boldsymbol{\theta})$ based on all model parameters $\boldsymbol{\theta}$ with a regularizer $\Omega(\widehat{\boldsymbol{\theta}}_v)$ enforcing sparsity in the input features:

$$\min_{\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}} \; \mathcal{L}_{\text{pred}}(\boldsymbol{\theta}) + \lambda \sum_{v=1}^{V} \Omega\left(\widehat{\boldsymbol{\theta}}_v\right), \tag{2}$$

where $\widehat{\boldsymbol{\theta}}_v$ are tunable parameters of the regularizer for variate v and $\lambda \in \mathbb{R}_+$ is a hyperparameter to adjust the degree of sparsity. One such approach are cMLPs [Tank et al., 2022]; multilayer perceptrons where the first weight matrix $\boldsymbol{W} \in \mathbb{R}^{D \times V}$ projecting from V features at each time step to D hidden dimensions is regularized to encode a sparse selection of input features. Ω is instantiated as an L^2 norm of its columns: $\sum_{v=1}^{V} \|\boldsymbol{W}_v\|_2$. Note that $\boldsymbol{\theta}$ and $\widehat{\boldsymbol{\theta}}$ can overlap.

Sparsity can then be extracted by binarizing the entries of $\boldsymbol{W}$ using a user-defined threshold τ. This is necessary as the L^2 penalty tends only to shrink parameters to small values near zero, yet not clamp them sharply to it. This, however, allows subsequent layers to amplify the dampened signal again and still use it for forecasting. We avoid this disadvantage in GC-xLSTM by explicitly optimizing the feature extractor for strict sparsity. This more principled approach works without determining a sparsity threshold τ.

Previous work has explored both more regularizers [Tank et al., 2022] and different means to extract Granger causal relationships, including using feature attribution via explainability [Atashgahi et al., 2024] and interpretability [Marcinkevičs and Vogt, 2021]. Furthermore, several works have gone towards learning relevant representations that respect the underlying Granger causality [Xu et al., 2016, Varando et al., 2021, Dmochowski, 2023]. Zoroddu et al. [2024] present another approach where prior knowledge is encoded in the form of a noisy undirected graph, which aids the learning of

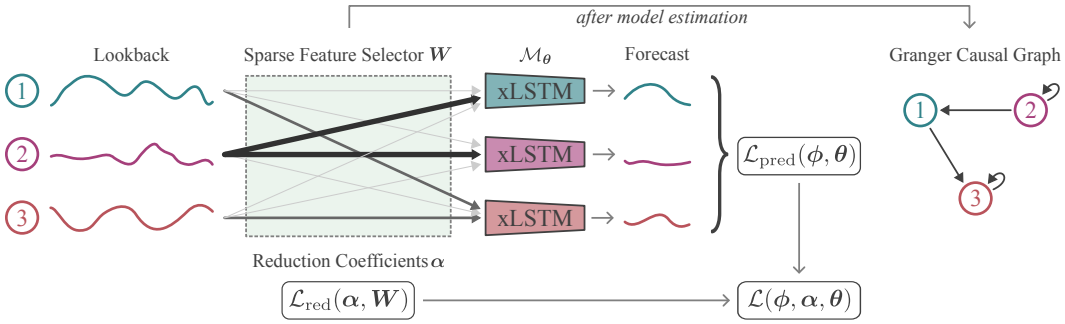


Figure 1: **GC-xLSTM performs three key steps to determine the Granger causal links:** Firstly, for each time series component, all variates are embedded with a sparse feature encoder $\boldsymbol{W}$ that is regularized through a novel sparsity loss with learned reduction coefficients $\boldsymbol{\alpha}$. xLSTM models then learn to autoregressively predict future steps from that embedding. Finally, once model estimation is complete, Granger causal dependencies can be extracted from $\boldsymbol{W}$.

Granger causality graphs. A more recent approach [Lin et al., 2024] employs Kolmogorov-Arnold networks [Liu et al., 2024] to learn the Granger causal relations between time series. For a discussion of when Granger causality implies (Pearlian) causality, we refer to Pettenuzzo and White [2011] and Das and Babadi [2023], allowing to connect to works such as the one of Rubenstein et al. [2017].

2.3 Extended Long Short-Term Memory (xLSTM)

Beck et al. [2024] propose two building blocks to build up xLSTM architectures: the sLSTM and mLSTM modules for vector-valued (i.e., multivariate) sequences. sLSTM cells improve upon classic LSTMs by exponential gating. For parallelizable training, mLSTM cells replace memory mixing between hidden states with an associative matrix memory. We will continue by recalling how sLSTM cells function since we found their memory mixing more effective in time series forecasting.

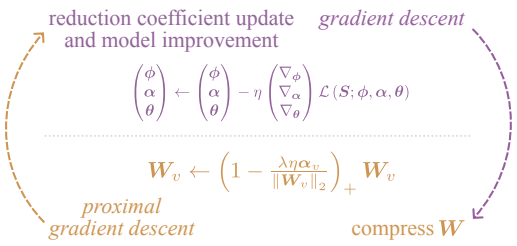
The standard LSTM architecture of Hochreiter and Schmidhuber [1997] updates the cell state $\mathbf{c}_t$ through a combination of input, forget, and output gates, which regulate the flow of information across tokens. sLSTM blocks, owing to the contained sLSTM cells, enhance this by incorporating exponential gating and memory mixing [Greff et al., 2017] to better handle complex temporal and cross-variate dependencies. Additional normalization states are introduced to stabilize training under the new exponential activation function. As Beck et al. have shown, it is sufficient and computationally beneficial to constrain the memory mixing performed by the recurrent weight matrices $\boldsymbol{R}_z, \boldsymbol{R}_i, \boldsymbol{R}_f$, and $\boldsymbol{R}_o$ to individual *heads*. This is inspired by the multi-head setup of Transformers [Vaswani et al., 2017], yet more restricted and efficient. In particular, each token gets broken up into groups of features, where the input weights $\boldsymbol{W}_{z,i,f,o}$ act across all of them, but the recurrence matrices $\boldsymbol{R}_{z,i,f,o}$ are implemented as block-diagonal. This permits specialization of the individual heads to patterns specific to the respective section of the tokens and empirically does not sacrifice expressivity.

3 GC-xLSTM

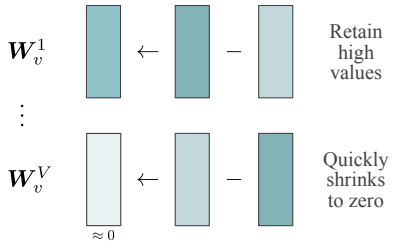
We will now introduce the GC-xLSTM architecture and detail the optimization for strict sparsity jointly with the model parameters. At the end, we will additionally discuss theoretical properties of the proposed system.

3.1 Overall Architecture

As Figure 1 shows, we estimate a pipeline of sparse feature selectors and xLSTM models to predict the multivariate time series. Eventually, this allows us to derive Granger causal dependencies from the selected features. Specifically, we learn for each variate v a separate sparse projection and compute $\boldsymbol{x}_v = \boldsymbol{W}_v \boldsymbol{S} + \boldsymbol{b}_v$ shared across time. The matrix $\boldsymbol{W}_v \in \mathbb{R}^{D \times V}$ is shared across all lags of the time series for simplicity. We will lift this restriction in Section 4.2. Note that $\boldsymbol{b}$ does not affect the sparse use of inputs. We write $\boldsymbol{\phi}$ for the set of parameters $\boldsymbol{W}_v$ and $\boldsymbol{b}_v$ for all $v \in \mathcal{V}$. In addition to selecting dependencies, the sparse projection embeds the data into D-dimensional hidden space.



(a) **The optimization procedure for GC-xLSTM alternates between GD and proximal GD.** A high value of α_v^w entails a strong compression of column $\boldsymbol{W}_v$, i.e. more sparsity in depending on variate w.



(b) **Visual representation of the compression step in Algorithm 1, line 16.** The matrix preserves columns with high norm that are important for Granger causality detection (dark) while suppressing others (light).

Figure 2: **Optimization procedure and compression intuition for GC-xLSTM.**

For Neural Granger Causality to successfully and faithfully extract the proper underlying dependencies, it is essential to employ models that can capture the complete set of dependencies. We, therefore, employ powerful deep-learning models with significantly higher capacity than the cMLPs and cLSTMs in prior work [Tank et al., 2022]. In particular, we instantiate the individual time series forecasters $\mathcal{M}_{\boldsymbol{\theta}_v, v}$ with sLSTM blocks as introduced in Section 2.3. They can capture long-range dependencies in time series data, substantially enhancing the capabilities of traditional LSTMs in handling extended contexts. GC-xLSTM consists of V sLSTM models, each modeling a different time series component. They are trained using established forecasting losses, such as the MSE loss $\mathcal{L}_{\text{pred}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ from Eq. (1).

3.2 Optimizing for Strict Input Sparsity

The purpose of the feature selector $\boldsymbol{W}$ is to only "pay attention" to as many variates as necessary for successful forecasting. A common approach to achieve this sparsity on the Granger causal relationships is via the lasso regularization explained in Section 2.2. We use a variation of the group lasso penalty [Yuan and Lin, 2006, Simon and Tibshirani, 2012] on the initial projection layer of GC-xLSTM as a structured sparsity-inducing penalty that encourages the selection of entire groups of variables, encoded as the columns of $\boldsymbol{W}_v$. Note that our penalty differs from adaptive lasso [Yuan and Lin, 2006], where the weights are not learned but are treated as fixed heuristics. Note that standard gradient descent methods cannot optimize such a penalty directly due to its non-differentiability. We thus adaptively compress by learning a reduction coefficient $\boldsymbol{\alpha}_v \in \mathbb{R}^V$ that selects which of the V columns of $\boldsymbol{W}_v$ are redundant. We perform this compression of $\boldsymbol{W}_v$ in a joint procedure with the general optimization of the forecasting model, as provided in Algorithm 1. Specifically, we perform two updates per optimization step for each of the variates, as Figure 2a depicts. Firstly, we optimize the projection weights $\boldsymbol{\phi}$, the reduction coefficients $\boldsymbol{\alpha}$, and the xLSTM parameters $\boldsymbol{\theta}$ using mini-batch gradient descent. This corresponds to lines 10 to 15 in Algorithm 1. It optimizes the following loss expected over the time series data $\boldsymbol{S}$:

$$\min_{\boldsymbol{\phi}_v, \boldsymbol{\alpha}_v, \boldsymbol{\theta}_v} \mathcal{L}_{\text{pred}}(\boldsymbol{S}; \boldsymbol{\phi}_v, \boldsymbol{\theta}_v) + \underbrace{\lambda \log\left(\sum_{w=1}^{V} \alpha_v^w \left\|\text{sg}\left(\boldsymbol{W}_v^w\right)\right\|_2\right)}_{\mathcal{L}_{\text{red}}(\boldsymbol{\alpha}_v, \boldsymbol{W}_v)} \tag{3}$$

Note that we, crucially, only descend on the reduction coefficient $\boldsymbol{\alpha}$ in $\mathcal{L}_{\text{red}}$ and not on $\boldsymbol{W}_v$, as the stop-gradient $\text{sg}(\cdot)$ denotes. This sparsity optimization is instead performed by the second step in the procedure, shown in line 16, where a proximal gradient descent step dynamically shrinks $\boldsymbol{W}_v$ proportional to $\boldsymbol{\alpha}_v$. The compression update takes a descent step towards the gradient of

$$\lambda \sum_{w=1}^{V} \alpha_v^w \left\|\boldsymbol{W}_v^w\right\|_2 \tag{4}$$

followed by a soft thresholding. Intuitively, the $\mathcal{L}_{\text{red}}$ component of Eq. (3) keeps $\boldsymbol{W}_v$ fixed while learning $\boldsymbol{\alpha}_v$, and Eq. (4) keeps $\boldsymbol{\alpha}_v$ fixed in the proximal step to compress $\boldsymbol{W}_v$. Figure 2b depicts the intuition of the proximal gradient step and soft-thresholding in line 16.

Details on learning the reduction loss $\mathcal{L}_{\text{red}}$. It is worth briefly discussing the use of the logarithm in Eq. (3). It mainly gives more equal weight to the decreases in $\boldsymbol{W}_v$ column norms and encourages learning of better sparse Granger causal relations. It furthermore normalizes the gradient updates to $\boldsymbol{\alpha}_v$. Empirically, this loss engineering allowed training models that were significantly more robust to noise and changes to the sparsity hyperparameter λ. This was reflected by a more stable variable usage and predictive loss $\mathcal{L}_{\text{pred}}$.

Ensuring non-negativity of the reduction coefficients $\boldsymbol{\alpha}$. For the proximal update step to be well-behaved, we need to ensure that $\boldsymbol{\alpha}_v$ results in a convex combination of column weights, i.e., that $\boldsymbol{\alpha}_v^w > 0$ for all $w \in \mathcal{V}$ and $\boldsymbol{\alpha}_v^T \mathbb{1} = 1$. We achieve this by re-parameterizing it as $\boldsymbol{\alpha}_v = \operatorname{softmax}(\boldsymbol{\beta}_v)$, and learning $\boldsymbol{\beta}_v$ instead of $\boldsymbol{\alpha}_v$.

Intuitive dynamics of the gradient update step. The weights $\boldsymbol{W}_v$ in the reduction loss $\mathcal{L}_{\text{red}}$ of Eq. (3) only serve to learn good reduction coefficients $\boldsymbol{\alpha}_v$, and are not optimized themselves. Deriving the gradient of the penalty term $\mathcal{L}_{\text{reg}}$ with respect to the underlying $\boldsymbol{\beta}$ provides a helpful intuition of the training dynamics:

$$\frac{\partial}{\partial \beta_v^w} \sum_{w=1}^{V} \alpha_v^w \left\| \boldsymbol{W}_v^w \right\|_2 = \frac{\partial}{\partial \beta_v^w} \sum_{w=1}^{V} \operatorname{softmax}(\boldsymbol{\beta}_v)^w \left\| \boldsymbol{W}_v^w \right\|_2 = \alpha_v^w \left(\left\| \boldsymbol{W}_v^w \right\|_2 - \sum_{w=1}^{V} \alpha_v^w \left\| \boldsymbol{W}_v^w \right\|_2 \right)$$

$$\implies \frac{\partial}{\partial \beta_v^w} \mathcal{L}_{\text{reg}} = \lambda \frac{\partial}{\partial \beta_v^w} \log \left(\sum_{w=1}^{V} \alpha_v^w \left\| \boldsymbol{W}_v^w \right\|_2 \right) = \lambda \alpha_v^w \left(\frac{\left\| \boldsymbol{W}_v^w \right\|_2}{\sum_{w=1}^{V} \alpha_v^w \left\| \boldsymbol{W}_v^w \right\|_2} - 1 \right)$$

We can see that if the norm of a column $\left\| \boldsymbol{W}_v^w \right\|_2$ is large, that corresponding $\frac{\partial}{\partial \beta_v^w} \mathcal{L}_{\text{reg}}$ will be large. Gradient descent will thus decrease α_v^w and effectively allocate less weight to its removal in the compression step. Furthermore, $\frac{\partial}{\partial \beta_v^w} \mathcal{L}_{\text{reg}}$ also scales with α_v^w, resulting in a self-reinforcing loop that aids learning sparse representations.

Practical considerations. Furthermore, we perform staged optimization of $\boldsymbol{\alpha}$, which is initialized to a uniform distribution by setting all $\boldsymbol{\beta} = 0$. We only start training $\boldsymbol{\alpha}$ after exploring the prediction loss and having obtained a reasonably compressed forecaster, which is controlled by the hyperparameter K in Algorithm 1 (see line 13). While we present the method with mini-batch gradient descent for conciseness, modern optimizers, such as Adam [Kingma and Ba, 2017], can further improve convergence.

3.3 Theoretical Analysis

Ultimately, we want to ensure that, provided real-world data, we can find hyperparameters such that Algorithm 1 discovers all and only those edges of the unique underlying GC graph $(\mathcal{V}, \mathcal{E})$ as per Definition 1. Providing convergence guarantees in full generality is notoriously hard for such practical architectures and optimization schemes, and thus rarely attempted. However, we can at least investigate whether the chosen model class containing $\mathcal{M}_{\boldsymbol{\theta},v}$ can approximate g_v to arbitrary precision. If that is the case, we can be reasonably sure that gradient-based optimization schemes will yield satisfactory approximations, even without formal guarantees.

The forecasting component of GC-xLSTM consists of two main steps: the sparse initial projection $\boldsymbol{W}_v$ and the subsequent xLSTM blocks. One might think that the sparsity of $\boldsymbol{W}_v$ hinders learning the correct g_v. However, the true underlying g_v is independent of all variates w without ingoing edges into variate v. Thus, depending on an appropriate choice of the sparsity hyperparameter λ, the projection $\boldsymbol{W}_v$ can encode exactly those as zero entries. It remains to investigate the approximation capabilities of the sLSTM blocks, which we present in Appendix B in more detail. In summary, we can assume that sLSTM blocks are at least as powerful as RNNs, which are, in turn, universal function approximators. Thus, the overall GC-xLSTM architecture is sufficiently rich to model g_v to adequate precision. We continue by confirming this empirically in the next section.

Table 1: **GC-xLSTM is highly accurate at discovering GC relations in the chaotic and non-linear Lorentz-96 system.** For each setting and baseline, we provide the accuracy (Acc.), balanced accuracy (BA), and AUROC. The best models are highlighted as **bold**.

		$F = 10$			$F = 40$		
Model	Year	Acc. (↑)	BA (↑)	AUROC (↑)	Acc. (↑)	BA (↑)	AUROC (↑)
VAR	2021	91.8±1.2	83.8±1.6	94.0±1.6	86.4±0.8	58.5±1.7	74.5±4.7
cLSTM	2022	97.0±1.0	95.0±2.8	95.8±2.6	84.4±1.2	65.6±3.7	66.1±3.8
cMLP	2022	97.2±0.5	95.6±1.6	96.3±1.8	68.3±2.7	80.5±1.7	**97.9**±1.6
GC-KAN	2024	–	–	92.1±0.3	–	–	87.1±0.4
TCDF	2019	87.1±1.2	70.9±4.4	85.7±2.7	77.5±2.3	62.2±3.0	67.9±3.1
eSRU	2020	96.6±1.1	95.1±2.0	96.3±2.0	86.7±0.9	88.6±1.4	93.4±2.1
GVAR	2021	98.2±0.3	98.2±0.6	**99.7**±0.1	94.5±1.0	88.5±4.6	97.0±0.9
GC-xLSTM	**ours**	**99.1**±0.2	**98.5**±1.0	99.3±0.3	**96.3**±0.3	**96.6**±0.3	88.0±0.2

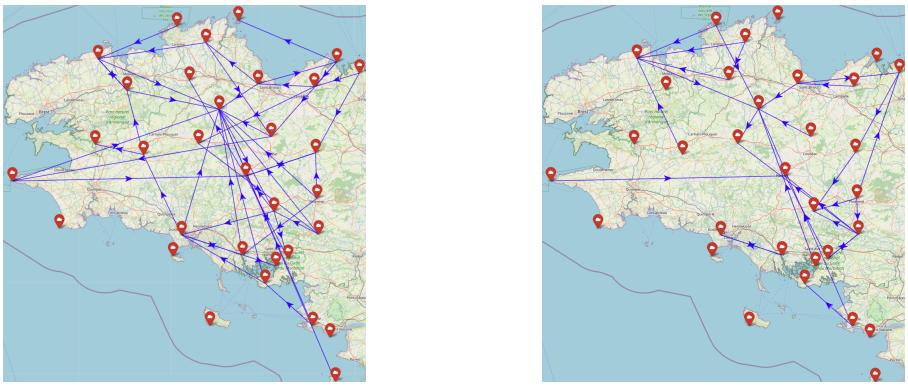
4 Experimental Evaluation

We conduct extensive experiments on six datasets to assess the practical effectiveness of GC-xLSTM. We will now explain the chosen architecture and parameters used to train GC-xLSTM and then discuss the datasets used before presenting our obtained results in detail. The results are split into investigating the general GC modeling capabilities on a diverse range of applications (Section 4.1) and a subsequent model analysis, including an ablation study and using different numbers of variates (Section 4.2).

Architecture Details. For our component-wise networks, we use a single xLSTM block comprised of one sLSTM layer, followed by a linear layer to predict the next time step from the preceding $C = 10$ time steps (see Table 5 in Appendix C for deviations). We then directly optimize the architecture on each time step. The hidden dimension of the sLSTM block is set to 32 for all datasets. We find that the presence of a gated MLP to up- and down-project the hidden states of the sLSTM block does not significantly improve performance, so we omit it in all our experiments for simplicity. We deliberately do not use any mLSTM blocks, as we also find that the sLSTM blocks are superior at capturing long-range dependencies in the data [Kraus et al., 2025].

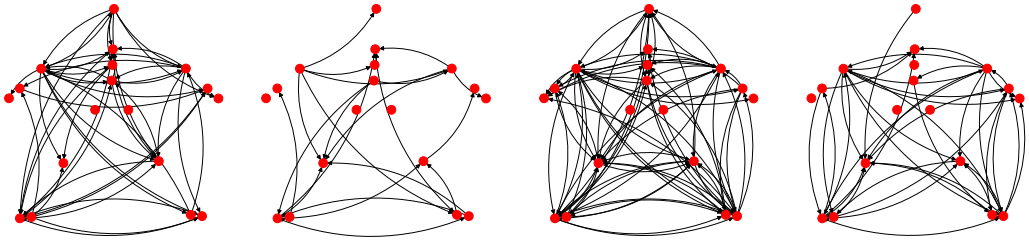
Training and Evaluation Details. We use the Adam optimizer [Kingma and Ba, 2017] for full gradient descent training with a weight decay of 0.1. We schedule the learning rate to follow a linear warmup of 2,000 iterations to $\eta = 10^{-4}$, followed by cosine annealing until the end of training for a total of 13,000 steps. We start learning the reduction coefficients after a warmup period of $K = 1,500$ iterations, during which the uniform compression across all columns combined with the prediction loss gives reasonable priors for the gradient directions of the reduction coefficients. Due to the moderate size of the datasets, we performed full-batch gradient descent. Only the sparsity hyperparameter λ was tuned specifically for each setting. This allows obtaining degrees of sparsity specifically tailored to the characteristics and requirements of each dataset and task. Note that, except for the customization of λ, we use essentially the same hyperparameter configuration for an extensive set of datasets, underpinning the robustness of GC-xLSTM. Following [Tank et al., 2022, Sec. 6.1], we compute all AUROC scores by sweeping over $\lambda \in \{5, \dots, 15\}$ in steps of one. This effectively integrates this hyperparameter as it covers the entire empirically viable range. All training runs were carried out on a single NVIDIA RTX A6000 GPU and concluded in at most 1.5 hours. Mini-batching could further decrease this modest training time.

Datasets. We evaluate GC detection with GC-xLSTM on six diverse datasets. Obtaining objective truth about the underlying graph for real-world scenarios is a constant challenge in Granger causality research. We thus employ the Lorenz-96 system of differential equations [Karimi and Paul, 2010], realistic fMRI brain activity simulations [Smith et al., 2011], and simulated linear VAR data following Tank et al. [2022]. For further qualitative insights on real-world data, we additionally analyze the Moléne weather dataset [Girault, 2015], human motion capture recordings [CMU, 2009], and company fundamentals [Divo et al., 2025]. An overview and more details are provided in Appendix C.



(a) $\lambda = 8$. (b) $\lambda = 10$.

Figure 3: **GC-xLSTM uncovers dynamic GC weather patterns in the Moléne dataset.** We observe that the sparsity of the learned Granger causal relations increases with higher λ.



(a) Salsa with $\lambda = 4$. (b) Salsa with $\lambda = 5$. (c) Running with $\lambda = 4$. (d) Running with $\lambda = 6$.

Figure 4: **GC-xLSTM captures complex human motions.** GC-xLSTM is able to uncover complex real-world dependencies in the Human Motion Capture dataset, giving us an intuitive understanding of the learned interactions.

4.1 Main Results

Lorenz-96. Table 1 evaluates GC detection with GC-xLSTM on three metrics. As a qualitative comparison, we provide results for seven well-known baselines: VAR as classic F-tests for Granger causality taken from Marcinkevičs and Vogt [2021], cMLP and cLSTM [Tank et al., 2022], GC-KAN [Lin et al., 2024], TCDF [Nauta et al., 2019], eSRU [Khanna and Tan, 2020], and GVAR [Marcinkevičs and Vogt, 2021]. Their scores are taken as reported in the original papers. First, GC-xLSTM outperforms all baseline methods for both $F = 10$ and $F = 40$ in both accuracy and balanced accuracy. This shows that GC-xLSTM reliably captures the underlying Granger causal relationships in the presence of limited and noisy data. Second, it demonstrates that the sparsity hyperparameter λ is rather well-behaved, as the AUROC score resulting from a sweep over a range of λ values still provides very competitive results. Third, as a qualitative validation, Figure 5 visually confirms the strong prediction accuracies of GC-xLSTM.

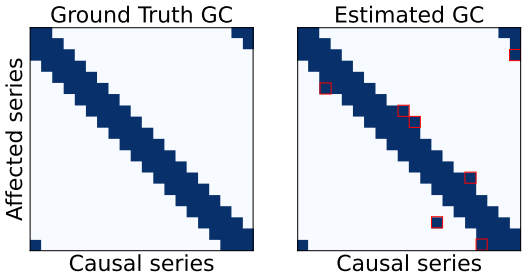


Figure 5: **GC-xLSTM uncovers the vast majority of GC edges.** In the highly chaotic $F = 40$ setting of the Lorenz-96 system GC-xLSTM is accurate in predicting the GC edges, shown in dark blue ■. Errors are marked red □.

Moléne. Unlike Zoroddu et al. [2024], who incorporate graph prior knowledge based on sensor locations, our approach learns the GC structure solely from the temperature observations. This ensures that GC-xLSTM does not inherently favor regional connections over long-range dependencies,

allowing it to discover dominant weather patterns operating both locally and across broader spatial scales. Adjusting λ allows balancing granularity and interpretability for insights into both local and regional dependencies. The dense structure of the resulting Figure 3a exhibits a richer set of GC interactions, while the more sparse Figure 3b highlights only the most pronounced edges.

fMRI. Next, we evaluated the efficacy of GC-xLSTM in noisy settings by considering the rich and realistic BOLD deconvolved time series provided by the fMRI data. As the balanced accuracies (BA) in Table 2 show, GC-xLSTM significantly outperforms the baseline models.

Table 2: **GC-xLSTM discovers brain connectivity highly accurately.**

Model	BA (↑)
TCDF	72.8±6.3
GVAR	65.2±4.5
VAR	51.3±1.5
cMLP	61.4±6.8
cLSTM	65.5±5.3
GC-xLSTM	**73.3**±3.0

Human Motion Capture. To analyze GC relations between body joints, we focus on two specific activities: Salsa dancing and running, which provide interpretable motion patterns. Figure 4 shows the results of the learned graphs for those activities. A closer look offers an intuitive understanding of the learned interactions. For example, in the Salsa dance, we observe edges from the feet to the knees and the arms, supporting the characteristic movements of the lower driving the upper body. We can also see the cross-limb correlation, with movements initiating on one side of the body and propagating across. Similarly, the results for running strongly establish the lower limbs as primary motion drivers, with edges from the feet and knees to the arms. The cyclic dependencies between the knees, ankles, and feet capture the repetitive, alternating nature of the gait.

Company Fundamentals. Lastly, we evaluated how well GC-xLSTM can uncover relationships between financial indicators of large companies. The dataset consists of short time series of quarterly performance indicators of 2527 large publicly traded companies. As the excerpt in Figure 6 shows and as verified by a financial expert, those extracted edges are mostly economically sensible. Results on the full set of 19 features are provided in Appendix D.

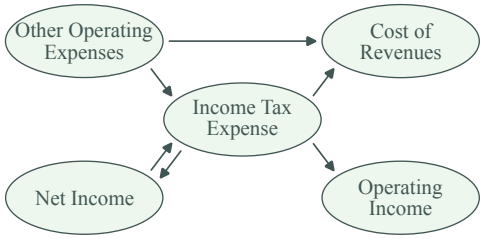


Figure 6: **GC-xLSTM extracts relations between company fundamentals.**

4.2 Model Analysis

Ablation Study. GC-xLSTM comprises two key innovations: Employing the xLSTM architecture (cf. Section 3.1) and a novel joint optimization strategy (cf. Section 3.2). To disentangle the empirical impact of each, we performed an ablation study. Table 3 to the right shows balanced accuracy results on the Lorentz ($F = 40$) and fMRI datasets. First, to assess the architectural contribution, we replaced the xLSTM block with a standard LSTM. The resulting substantial performance drop (GC-xLSTM → (I)) validates the importance of the xLSTM's advanced modeling capacity. Second, to evaluate the optimization strategy, we substituted it with standard Group Lasso [Simon and Tibshirani, 2012]. This also led to a marked decline in performance (GC-xLSTM → (II)), confirming its ability to enforce strict input sparsity.

Table 3: **Both the xLSTM forecaster and the novel joint optimization of GC-xLSTM drive its improved performance.**

Ablation	Forecaster	Optimization	Lorentz	fMRI
GC-xLSTM	xLSTM	Joint	**96.6±0.3**	**73.3±3.0**
(I)	LSTM	Joint	93.0±0.3	62.8±2.0
(II)	xLSTM	Group Lasso	73.0±4.6	65.4±2.0

Variable Number of Lags. GC-xLSTM can naturally be extended to learn separate projections $\boldsymbol{W}^{(\ell)}$ per time lag ℓ, effectively inflating $\boldsymbol{W}$ to rank three. We evaluate it on the simulated VAR dataset across different time series lengths T and number of variates V in Table 4, proving its ability to learn multiple lag-specific relations without training additional xLSTM models.

Table 4: **GC-xLSTM discovers specific relations per lag.** Balanced Accuracies for simulated VAR at different lengths T and number of variates V.

BA (↑)	$T = 250$	$T = 500$	$T = 1000$
$V = 10$	93.1±3.0	92.5±1.0	95.5±1.0
$V = 20$	83.9±1.0	89.1±1.5	88.5±2.0

Complexity and Scaling Behaviour. To discover the entire GC graph, we need to fit V models $\mathcal{M}_{\boldsymbol{\theta}_v, v}$ to obtain the respective sparse projection matrices $\boldsymbol{W}_v$ containing all edges arriving at each v. Assuming for brevity that the latent dimension $D \propto V$, i.e., is a fixed multiple of the number of variates V, each fitting runs in $O(TV^2/N_h)$ time and requires $O(TV^2/N_h)$ space, with T being the time steps and N_h the number of sLSTM heads. Their block-wise structure permits that, despite the squared effort, forward and backward passes are efficiently computable even for thousands of dimensions. Depending on whether all V models are estimated in sequence or parallel, either the time or memory complexity multiplies by V to arrive at the total cost. In practice, however, GC-xLSTM is extremely efficient on contemporary computing platforms due to the availability of highly optimized implementations for xLSTM layers. Figure 8 in Appendix E shows that it effectively scales linearly in the number of variates for the ranges relevant to standard GC detection settings.

Inspecting Training Dynamics. Here, we elaborate on the utility of the logarithm in the reduction loss $\mathcal{L}_{\text{red}}$ of Eq. (3). The logarithm term incentivizes the model to explore sparser solutions to GC discovery by allowing the training to move forward over any local minima that use the complete set of input variates. As the projection matrix becomes sparser and the input variates vanish from consideration by the model, the prediction loss $\mathcal{L}_{\text{pred}}$ will slightly increase. As seen in Figure 9 in Appendix F, an increase in the sparsity of the feature selectors $\boldsymbol{W}$ drives down the loss and enables learning more meaningful GC relations. It also shows how the variable usage quickly stabilizes.

5 Conclusion

We presented GC-xLSTM, a novel xLSTM-based model to uncover Granger causal relations from the underlying time series data. GC-xLSTM first enforces sparsity between the time series components and then learns a weight per time series to decide the importance of each time series for the underlying task. Each time series component is then modeled using a separate xLSTM model, which enables it to better discover Granger causal relationships between the time series variables. We validated GC-xLSTM in six scenarios, showing its effectiveness and adaptability in uncovering Granger causal relationships even in the presence of complex and noisy data.

Limitations. While Section 3.3 provides a theoretical analysis of GC-xLSTM, it does not give guarantees in the form of mathematical proofs. The rigor is limited by a lack of formal analysis of xLSTM blocks (see also Appendix B), which are yet to be formally analyzed to the same degree as more established architectures like LSTMs. While we discuss and measure the scaling behaviour of GC-xLSTM in Section 4.2, we only focused on dozens of variates as common in the literature.

Future work. This includes using more sophisticated architectures such as TimeMixer [Wang et al., 2024] or xLSTM-Mixer [Kraus et al., 2025]. Furthermore, discovering causal links specific to certain lags could be refined, where per-lag projections are learned for the near past and a remainder projection for the more distant lags. Finally, extending our evaluations to more real-world datasets encompassing domains such as climate change or ecology is an essential next step.

Acknowledgments and Disclosure of Funding

This work received funding from the ACATIS Investment KVG mbH project "Temporal Machine Learning for Long-Term Value Investing" and the KompAKI project of the German Federal Ministry of Research, Technology and Space within the "The Future of Value Creation – Research on Production, Services and Work" program (funding number 02L19C150), managed by the Project Management Agency Karlsruhe (PTKA). The TU Eindhoven author received support from their Dep. of Mathematics and Computer Science and the Eindhoven AI Systems Institute. Furthermore, this work benefited from the DYNAMIC Centre funded by the LOEWE program of the Hessian Ministry of Science and Research, Arts and Culture (HMWK) as LOEWE 1/16/519/03/09.001 (0009)98 and the HMWK project "The Third Wave of Artificial Intelligence – 3AI". It also benefited from the early stage of the German Federal Ministry for Economic Affairs and Energy project "Souveräne KI für Europa" (13IPC040G) as part of the EU funding program IPCEI-CIS; funding has not started yet, and from early stages of the Cluster of Excellence "Reasonable AI" (EXC-3057) funded by the German Research Foundation (DFG) under Germany's Excellence Strategy; funding will begin in 2026. Funding for H. Poonia to attend NeurIPS was provided by the CMU GSA/Provost Conference Funding. The authors are responsible for the content of this publication. Map data © OpenStreetMap contributors, licensed under the ODbL and available from openstreetmap.org.

References

Musleh Alharthi and Ausif Mahmood. xLSTMTime: Long-Term Time Series Forecasting with xLSTM. *MDPI AI*, 5(3):1482–1495, 2024.

Zahra Atashgahi, Tennison Liu, Mykola Pechenizkiy, Raymond Veldhuis, Decebal Constantin Mocanu, and Mihaela van der Schaar. Unveiling the Power of Sparse Neural Networks for Feature Selection, August 2024.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-Shot Forecasting Across Long and Short Horizons with Enhanced In-Context Learning, May 2025. arXiv:2505.23719v1.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael K. Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended Long Short-Term Memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11525–11533, 2024.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179.

CMU. Carnegie mellon university motion capture database, 2009. Data retrieved from CMU, `http://mocap.cs.cmu.edu/`.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K. Mathur, Rajat Sen, and Rose Yu. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Transactions on Machine Learning Research*, May 2023. ISSN 2835-8856.

Proloy Das and Behtash Babadi. Non-Asymptotic Guarantees for Reliable Identification of Granger Causality via the LASSO. *IEEE transactions on information theory*, 69(11):7439–7460, November 2023. ISSN 0018-9448. doi: 10.1109/tit.2023.3296336.

Felix Divo, Eric Endress, Kevin Endler, Kristian Kersting, and Devendra Singh Dhami. Forecasting Company Fundamentals. *Transactions on Machine Learning Research*, May 2025. ISSN 2835-8856.

Jacek Dmochowski. Granger components analysis: Unsupervised learning of latent temporal dependencies. *Advances in Neural Information Processing Systems*, 36:78168–78180, 2023.

Benjamin Girault. Stationary graph signals using an isometric graph translation. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1516–1520, 2015. doi: 10.1109/EUSIPCO.2015.7362637.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.

James J. Heckman. Econometric causality. *International statistical review*, 76(1):1–27, 2008.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

A. Karimi and M. R. Paul. Extensive chaos in the lorenz-96 model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(4), November 2010. ISSN 1089-7682. doi: 10.1063/1.3496397.

Saurabh Khanna and Vincent Y. F. Tan. Economy Statistical Recurrent Units For Inferring Nonlinear Granger Causality. In *Eighth International Conference on Learning Representations*, April 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ArXiv:1412.6980*, 2017.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xLSTM-Mixer: Multivariate Time Series Forecasting by Mixing via Scalar Memories. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hongyu Lin, Mohan Ren, Paolo Barucca, and Tomaso Aste. Granger causality detection with kolmogorov-arnold networks. *arXiv preprint arXiv:2412.15373*, 2024.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pages 509–525. PMLR, 2022.

Matteo Marchi, Bahman Gharesifard, and Paulo Tabuada. Training deep residual networks for uniform approximation guarantees. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, pages 677–688. PMLR, May 2021. ISSN: 2640-3498.

Ričards Marcinkevičs and Julia E. Vogt. Interpretable Models for Granger Causality Using Self-explaining Neural Networks. In *International Conference on Learning Representations*, January 2021.

Ricardo P. Masini, Marcelo C. Medeiros, and Eduardo F. Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

Manfred Mudelsee. Trend analysis of climate time series: A review of methods. *Earth-science reviews*, 190:310–322, 2019.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal Discovery with Attention-Based Convolutional Neural Networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, March 2019. ISSN 2504-4990. doi: 10.3390/make1010019.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Davide Pettenuzzo and Jr White. Granger Causality, Exogeneity, Cointegration, and Economic Policy Analysis, October 2011. doi: 10.2139/ssrn.2182749.

Paul K Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal Consistency of Structural Equation Models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Sydney, Australia, August 2017. AUAI Press.

Ali Shojaie and Emily B. Fox. Granger causality: A review and recent advances. *Annual Review of Statistics and Its Application*, 9(1):289–319, 2022.

H. T. Siegelmann and E. D. Sontag. On the Computational Power of Neural Nets. *Journal of Computer and System Sciences*, 50(1):132–150, February 1995. ISSN 0022-0000. doi: 10.1006/jcss.1995.1013.

Elsa Siggiridou and Dimitris Kugiumtzis. Granger causality in multivariate time series using a time-ordered restricted vector autoregressive model. *IEEE Transactions on Signal Processing*, 64 (7):1759–1773, 2015.

Noah Simon and Robert Tibshirani. Standardization and the Group Lasso Penalty. *Statistica Sinica*, 22(3):983–1001, July 2012. ISSN 1017-0405. doi: 10.5705/ss.2011.075.

Stephen M. Smith, Karla L. Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F. Beckmann, Thomas E. Nichols, Joseph D. Ramsey, and Mark W. Woolrich. Network modelling methods for FMRI. *NeuroImage*, 54(2):875–891, January 2011. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2010.08.063.

Chang hoon Song, Geonho Hwang, Jun ho Lee, and Myungjoo Kang. Minimal Width for Universal Property of Deep RNN. *Journal of Machine Learning Research*, 24(121):1–41, 2023. ISSN 1533-7928.

Nika Strem, Devendra Singh Dhami, Benedikt Schmidt, Benjamin Klöpper, and Kristian Kersting. Apt: Alarm prediction transformer. *Expert Systems with Applications*, 261:125521, 2025.

Paulo Tabuada and Bahman Gharesifard. Universal approximation power of deep residual neural networks via nonlinear control theory. In *International Conference on Learning Representations*, October 2020.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural Granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2022.

Hugues Turbé, Mina Bjelogrlic, Christian Lovis, and Gianmarco Mengaldo. Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, 5(3):250–260, 2023.

Gherardo Varando, Miguel-Angel Fernández-Torres, and Gustau Camps-Valls. Learning Granger causal feature representations. In *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting? *Neurocomputing*, 619:129178, February 2025. ISSN 0925-2312. doi: 10.1016/j.neucom.2024.129178.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning Granger causality for hawkes processes. In *International conference on machine learning*, pages 1717–1726. PMLR, 2016.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Wei Zhang, Thomas Panum, Somesh Jha, Prasad Chalasani, and David Page. Cause: Learning Granger causality from event sequences using attribution methods. In *International Conference on Machine Learning*, pages 11235–11245. PMLR, 2020.

Lucas Zoroddu, Pierre Humbert, and Laurent Oudre. Learning network Granger causality using graph prior knowledge. *Transactions on Machine Learning Research*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We explicitly list the contributions at the end of Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations as an explicitly named paragraph in Section 5. We acknowledge that any empirical results are by their very nature limited to the settings in which they were obtained, and thus strive to accurately describe them for best reproducibility (see also Question 4).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: While we perform theoretical anlaysis in Section 3.3 and appendix B, these are not in the form of mathematical theorems.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All implementation details, including dataset descriptions and experiment configurations, are provided in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We strive to use freely available datasets where possible and exclusively employ openly available software. Furthermore, we provide the source code for full reproducibility at github.com/harpoonix/GC-xLSTM.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All such details are provided in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All quantitative results are accompanied by standard deviations (specifically, see Table 1).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is provided at the beginning of Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We made sure to comply with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: While we acknowledge that many technologies have wide-ranging societal impacts, our primary focus is on technical innovation. We have thus not identified specific concerns requiring emphasis in this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Please see Question 10. Specifically, we do not provide any trained models or similar high-risk artifacts.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We carefully cite all immediately relevant scholarly works and provide URLs to any other resources in Section 4 and appendix C.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The documentation of the source code is also provided at github.com/harpoonix/GC-xLSTM.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We did not perform any such experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Please see Question 14.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We did not employ LLMs in any part of GC-xLSTM.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Full Algorithm

Algorithm 1 describes the entire procedure for extracting a Granger causal graph using GC-xLSTM. $(\cdot)_+$ denotes truncation as $\max(\cdot, 0)$.

Algorithm 1 Granger causality detection with GC-xLSTM

Input: Training data $\mathcal{D} = \left\{\boldsymbol{S}^{(i)}\right\}_{i\in\{1,\dots,N\}}$, sparsity hyperparameter $\lambda \in \mathbb{R}_+$, learning rate $\eta \in \mathbb{R}_+$, and compression schedule start $K \in \mathbb{N}$
Output: Granger causal graph

for all $v \in \mathcal{V}$ **do** ▷ Training decomposes over $\mathcal{V}$
  $\boldsymbol{\phi}_v : \boldsymbol{W}_v, \boldsymbol{b}_v \sim \mathcal{U}\left(-\sqrt{1/V}, \sqrt{1/V}\right)$ ▷ Kaiming/He initialization
  $\boldsymbol{\alpha}_v \leftarrow 1/V \mathbb{1}$ ▷ Uniform reduction coefficient
  $\boldsymbol{\theta}_v \leftarrow \boldsymbol{\theta}^{(0)}$ ▷ Standard xLSTM initialization
  $k \leftarrow 0$
  repeat
    Sample random mini-batch $\mathcal{B} \sim \mathcal{D}$
    $\boldsymbol{g}_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\phi},\boldsymbol{\alpha},\boldsymbol{\theta}} \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{S}\in\mathcal{B}} \left[\mathcal{L}_{\text{pred}}(\boldsymbol{S}; \boldsymbol{\phi}_v, \boldsymbol{\theta}_v) + \lambda \log\left(\sum_{w=1}^{V} \alpha_v^w \left\|\text{sg}\left(\boldsymbol{W}_v^w\right)\right\|_2\right)\right]$
    $\boldsymbol{\phi}_v \leftarrow \boldsymbol{\phi}_v - \eta \boldsymbol{g}_{\boldsymbol{\phi}}$ ▷ GD step
    $\boldsymbol{\theta}_v \leftarrow \boldsymbol{\theta}_v - \eta \boldsymbol{g}_{\boldsymbol{\theta}}$ ▷ GD step
    if $k \geq K$ **then** ▷ Optimize $\boldsymbol{\alpha}$ after K steps
      $\boldsymbol{\alpha}_v \leftarrow \boldsymbol{\alpha}_v - \eta \boldsymbol{g}_{\boldsymbol{\alpha}}$ ▷ GD step
    end if
    $\boldsymbol{W}_v \leftarrow \left(1 - \frac{\lambda\eta\boldsymbol{\alpha}_v}{\|\boldsymbol{W}_v\|_2}\right)_+ \boldsymbol{W}_v$ ▷ Compression step with proximal GD
    $k \leftarrow k + 1$
  until convergence
end for
$\mathcal{E} \leftarrow \{(v, w) \in \mathcal{V} \times \mathcal{V} \mid \|\boldsymbol{W}_v^w\|_2 > 0\}$
return extracted graph $(\mathcal{V}, \mathcal{E})$

B On the Approximation Capabilities of sLSTM blocks

Classic RNNs have long been known to be extremely powerful models of computation. Specifically, they are Turing complete [Siegelmann and Sontag, 1995] and, by extension, universal function approximators [Song et al., 2023]. Traditional LSTMs as proposed by Hochreiter and Schmidhuber [1997] are universal function approximators being at least as powerful as the RNNs [Song et al., 2023, Corollary 16]. Due to their novelty, the sLSTM *cells* as presented in Section 2.3 have not yet received the same degree of theoretical analysis.[2] Yet, it appears natural to extend the same reduction to RNNs as has been shown for LSTMs, since the main technicality is the differing normalization of the hidden state [Beck et al., 2024, Eq. 36] and the alternative exponential activation function, which Song et al. [2023] abstract away with. sLSTM *blocks* then wrap these cells with additional operations (cf. Beck et al. [2024], App. A.4). They, however, can be carefully configured to reduce to only rescaling Layer Normalizations and residual connections, which are known to, again, yield universal function approximators for many architectures [Tabuada and Gharesifard, 2020, Marchi et al., 2021]. This can be shown by first omitting the optional upfront convolution and Swish activation. We, furthermore, set the number of heads to $N_h = 1$, causing the per-head Group Normalization after the cell to degenerate to yet another Layer Normalization. Lastly, the final post up-projection of $\boldsymbol{x}_{\text{in}}$ to $\boldsymbol{x}_{\text{out}}$ is defined as

$$\boldsymbol{x}_{\text{out}} = \boldsymbol{W}_3 \left(\left(\boldsymbol{W}_1 \boldsymbol{x}_{\text{in}} + \boldsymbol{b}_1\right) \odot \text{GeLU}\left(\boldsymbol{W}_2 \boldsymbol{x}_{\text{in}} + \boldsymbol{b}_2\right)\right) + \boldsymbol{b}_3.$$

[2] A discussion of expressivity hierarchies can be found in Auer et al. [2025, App. A].

With the right instantiation, where α is chosen such that $\mathrm{GeLU}(\alpha) = 1$, we obtain

$$\begin{aligned}
\boldsymbol{x}_{\text{out}} &= \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & \mathbb{I} & 0 \end{pmatrix} \left(\left(\left(\begin{pmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & \mathbb{I} \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}_{\text{in}} + \mathbf{0} \right) \odot \mathrm{GeLU}\left(\mathbf{0} \boldsymbol{x}_{\text{in}} + \alpha \mathbf{1} \right) \right) + \mathbf{0} \right) \\
&= \begin{pmatrix} \mathbb{I} & 0 & 0 & 0 \\ 0 & \mathbb{I} & 0 & 0 \\ 0 & 0 & \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & \mathbb{I} \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{x}_{\text{in}} \odot \mathbf{1} \\
&= \boldsymbol{x}_{\text{in}}.
\end{aligned}$$

This, again, shows that sLSTM blocks are at least as general as LSTMs. While a rigorous proof is far beyond the scope of this work, this investigation still underpins the strong capabilities of this architecture. These theoretical considerations align well with empirical findings, showing that xLSTM blocks are at least as effective as LSTMs [Beck et al., 2024].

C Dataset Details

This section details all six datasets used in our empirical evaluation of GC-xLSTM. An overview is provided in Table 5.

Table 5: **Overview of the diverse collection of datasets.** It also shows the number of variates V, time steps T, samples N, and look-back context steps of GC-xLSTM C.

Name	Origin	Type	V	T	N	C
Lorenz-96	Karimi and Paul [2010]	Simulated	20	500	1	10
fMRI	Smith et al. [2011]	Real-world	15	200	1	10
Moléne	Girault [2015]	Real-world	32	744	1	10
Human MoCap (Run)	CMU [2009]	Real-world	54	1232	61	10
Human MoCap (Salsa)	CMU [2009]	Real-world	54	4136	30	10
Company Fundamentals	Divo et al. [2025]	Real-world	19	56	2527	40
VAR	Karimi and Paul [2010]	Simulated	10/20	mult.	1	5

Lorenz-96. The V-dimensional Lorenz-96 model [Karimi and Paul, 2010] is a chaotic multivariate dynamical system governed by the differential equations

$$\frac{dx_{t,i}}{dt} = (x_{t,i+1} - x_{t,i-2})x_{t,i-1} - x_{t,i} + F,$$

with the external forcing coefficient F regulating the non-linearity of the system. Low values of F correspond to near-linear dynamics, while higher values induce chaotic behavior. There are two sources of randomness in the system. Firstly, we sample i.i.d. starting conditions from $\mathcal{N}(0, 0.01)$. Secondly, in each step of the simulation, we add i.i.d. noise sampled from $\mathcal{N}(0, 0.1)$ to $x_{t,i}$. Following the setting of Tank et al. [2022] for best comparability, we simulate $V = 20$ variates with a sampling rate of $\Delta t = 0.05$ for a total of $T = 500$ time steps after a brief burn-in time. We use two forcing constants $F \in \{10, 40\}$ to test our model under different levels of non-linearity.

fMRI. Discovering connectivity networks within (human) brains is a key application of GC detection methods. To this end, brain activity is measured non-invasively using functional magnetic resonance imaging (fMRI) over time and grouped into regions between which connections are looked for. Specifically, we used the realistic simulations of blood-oxygen-level-dependent (BOLD) deconvolved data of Smith et al. [2011].

Moléne. The Moléne dataset [Girault, 2015] contains hourly temperatures recorded by sensors at $V = 32$ locations in Brittany, France, during $T = 744$ hours. The objective is to understand the spatio-temporal dynamics of the temperature and to assess the extent to which the model can uncover complex relationships in weather by considering only local observations.

Human Motion Capture. We also apply our methodology to detect complex, nonlinear dependencies in human motion capture (MoCap) recordings. In contrast to the Lorenz-96 simulated dataset results, this analysis allows us to visualize and interpret the learned network more easily. We consider a data set from the CMU MoCap database [CMU, 2009]. The data comprises $V = 54$ joint angle and body position recordings across multiple subjects. Since some regions, like the neck, have multiple degrees of freedom in both translation and rotation, we consider the GC relations between two joints based on edges between all movement directions. The motion ranges from locomotion (e.g., walking) over physical activities such as gymnastics and dancing to day-to-day social interactions.

Company Fundamentals. Another common field of application of GC detection methods is discovering links in economic data, where it is otherwise hard to maintain an overview. We, therefore, benchmark GC-xLSTM on company fundamentals data [Divo et al., 2025]. The dataset contains $V = 19$ economic variables, such as the Net Income and the Total Liabilities of 2527 companies. The data was collected quarterly from 2009 Q1 to 2023 Q3, resulting in only $T = 56$ time steps.

VAR. Following the well-known setup of Tank et al. [2022], we generate a two-step linear autoregressive process in $V \in \{10, 20\}$ variates. All variables depend on themselves, and an additional three random dependencies are added as targets to be discovered. The number of time steps varies in $T \in \{250, 500, 1000\}$, and we start recording the data after a brief burn-in time.

D Extended Experimental Results

This section supplements the experimental findings of Section 4.1. Specifically, Figure 7 provides the complete set of relations extracted from the Company Fundamentals dataset. On this challenging full feature set, only some of the discovered GC relations are economically plausible.

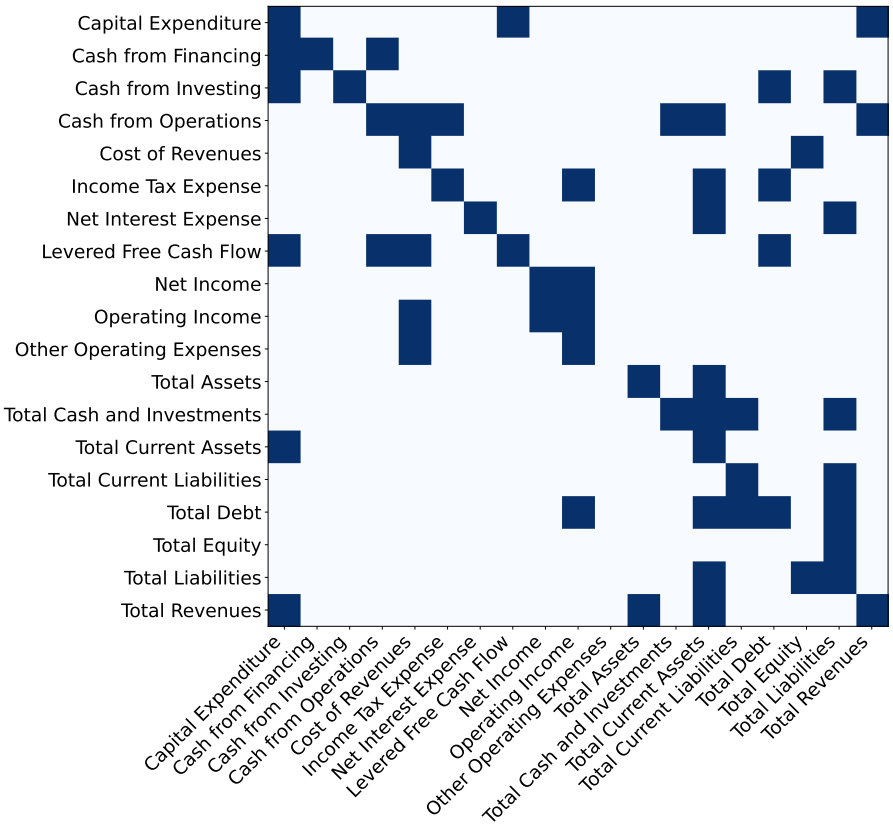


Figure 7: **GC-xLSTM uncovers relationships between economic variables within individual companies.** Established GC links are highlighted in **dark blue**.

E Scaling Behaviour

Figure 8 shows the training time and peak GPU memory reserved during the training of GC-xLSTM on the Lorenz-96 dataset with $T = 1000$ for various numbers of variates V.

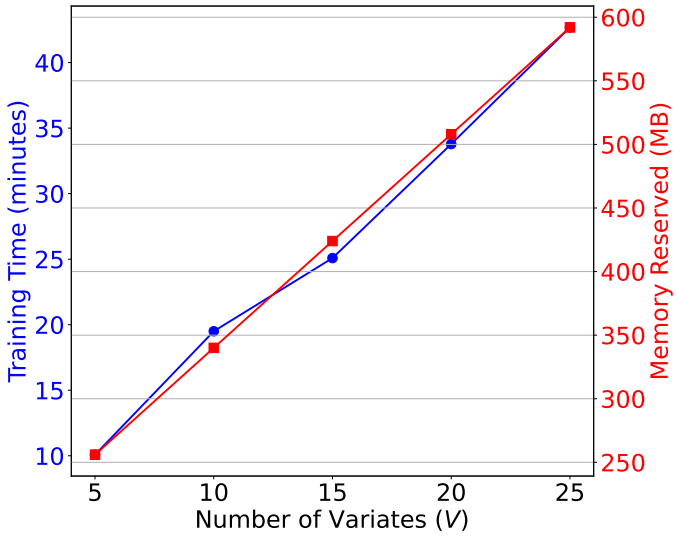
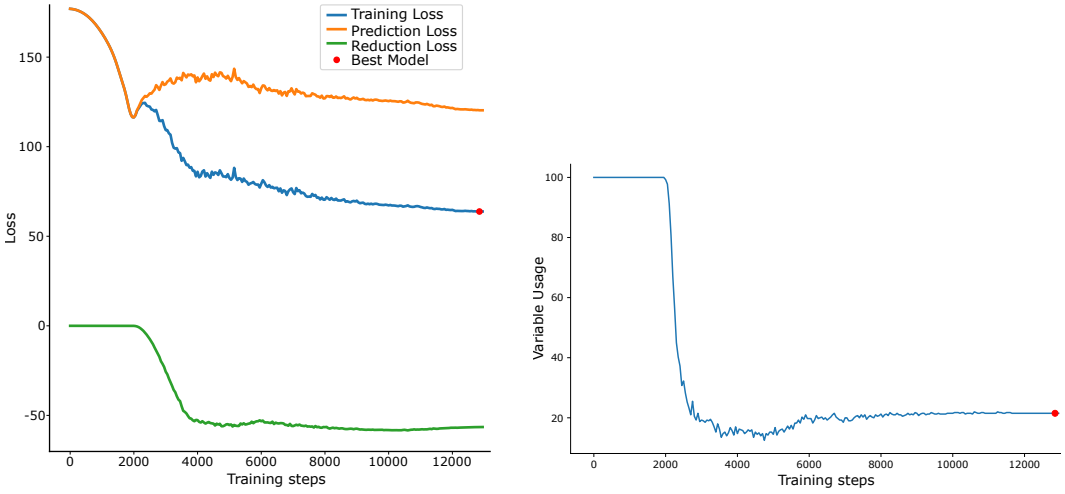


Figure 8: **GC-xLSTM effectively scales linearly in the relevant range of variate counts.**

F Additional Insights into Training

Figure 9a shows how the different loss components change during training. The robustness of training is reflected in the stability of variable usage once an optimal set is found, as Figure 9b shows.



(a) Visualization of the different loss components.

(b) Variable drops once the threshold $k \geq K$ is reached and then quickly stabilizes.

Figure 9: **Algorithm 1 jointly optimizes the prediction loss while adaptively establishing sparsity.** Results show training on Lorenz-96 with $F = 40$ and $T = 1000$.