LEARNING DEPENDENCIES AND LOCALIZING ANOMALIES WITH NEURAL ROSENBLATT TRANSFORM

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series anomaly localization remains relatively understudied, despite its importance for diagnosing anomalies in many real-world complex systems. Existing models may result in ambiguous per-series anomaly scores, mainly due to inaccurate modeling of cross-series dependencies. Furthermore, methods focused on dependency learning typically do not address anomaly localization and are mostly designed for continuous-valued time series, limiting their applicability to mixed continuous–discrete settings common in real-world systems. To address these limitations, **NeuralRos** is proposed to jointly learn cross-series dependencies and localize anomalies in mixed-type multivariate time series. NeuralRos reformulates the Rosenblatt transformation from a diagnostic tool into a learning mechanism, combining a learnable variable ordering with neural conditional distribution models to capture cross-series dependencies. Two complementary localization scores are derived: a likelihood-based score from the learned normal conditional distributions and a Rosenblatt-based score from deviations of the transformed variables from their theoretical behavior. Experiments on synthetic and real-world datasets demonstrate the effectiveness of NeuralRos across both tasks.

1 INTRODUCTION

Multivariate time series (MTS) data arise in many real-world systems where multiple interacting variables are continuously monitored to characterize system behavior. As these systems become increasingly complex, they also become more vulnerable to faults (anomalies). Consequently, MTS anomaly detection has received considerable attention, with substantial progress made in recent years (Correia et al., 2024). However, without identifying which time series contributes mostly to the observed anomalies, appropriate mitigation actions cannot be effectively taken. This task, known as anomaly localization, remains relatively understudied despite its importance. At the same time, effective diagnosis requires understanding the system behavior and how its variables interact. Therefore, it is crucial for a model to properly learn MTS cross-series dependencies and localize its anomalous behavior.

A key challenge of machine-learning and deep-learning models for MTS localization is that they can produce ambiguous or misleading per-series anomaly scores (Li et al., 2021; Shimillas et al., 2026). This is less problematic for detection, where scores are typically aggregated across time series, but becomes critical for localization, which requires reliable attribution to individual series. One major factor is the limited ability of existing models to accurately capture cross-series dependencies, which greatly determine how anomalous behavior is attributed across series (Han et al., 2025; Shimillas et al., 2026). Moreover, several methods cannot explicitly infer the dependencies learned by the model, limiting their practical value for system diagnosis. Despite the close and complementary nature of cross-series dependency learning and anomaly localization, only a few methods address both tasks jointly. Cross-series dependency learning is closely related to causal discovery, where most methods aim to recover causal structure from observational data

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

using existing causal discovery techniques. However, these methods do not directly provide mechanisms for anomaly localization or broader diagnostic tasks. In addition, most real-world MTS data comprise a mixture of continuous and discrete time series. However, this mixed-type setting remains relatively understudied (Debaly & Truquet, 2023) and is not taken into account by many existing methods. Hence, current state-of-the-art methodologies have limitations when applied to real-world datasets.

To address these limitations, the Neural Rosenblatt (**NeuralRos**) model is proposed to jointly learn cross-series dependencies and localize anomalies in MTS containing continuous- and discrete-valued time series. NeuralRos converts the Rosenblatt transformation from a diagnostic tool into a learning mechanism by building a multivariate time-series learning model around its probabilistic structure and using its theoretical properties to guide the learning process. The main contributions are as follows:

- **NeuralRos** is introduced for cross-series dependency learning. It combines a *Learnable Rosenblatt Ordering (LRO)* module, which learns the variable ordering for the Rosenblatt factorization, with neural conditional distribution models (neural conditioners) to capture cross-series dependencies for both continuous and discrete-valued time series. NeuralRos also has generative capabilities, which are important for system diagnosis.
- Two anomaly localization scores are derived from the learned NeuralRos model: **NeuralRos-L**, based on the learned conditional distributions, and **NeuralRos-R**, based on the Rosenblatt-transformed variables.
- A comprehensive evaluation of anomaly localization and cross-series dependency identification is conducted on synthetic and real-world datasets, including mixed-type settings.

2 Related work

Anomaly localization remains relatively underexplored despite its practical importance. Since abnormal-event labels are often unavailable, unsupervised methods are generally more applicable. Most approaches learn normal MTS behavior from training data and use test-time deviations as anomaly scores, commonly through reconstruction or forecasting models with per-series residuals used for localization (Zhang et al., 2024; Correia et al., 2024). A range of architectures has been explored, including CNN-based models (Hsu & Liu, 2021; Liang et al., 2022; Zhang et al., 2019), recurrent models (Su et al., 2019), and GAN-based approaches (Choi et al., 2020). However, directly using per-series anomaly scores can yield ambiguous localization, motivating methods that further refine these scores. InterFusion (Li et al., 2021) uses Markov Chain Monte Carlo to refine the scores. However, its computational cost limits localization to anomalous segments rather than individual time steps. DAEMON (Chen et al., 2023) uses integrated gradients (Sundararajan et al., 2017), counterfactual methods (Trifunov et al., 2021) replace anomalous observations with normal values, and STAS-SFAS (Shimillas et al., 2025) refines reconstruction scores using spatio-temporal and statistical signals. ϵ-Diagnosis (Shan et al., 2019) detects distribution shifts, while RCD (Ikram et al., 2022) tests conditional independence.

Despite these advances, most methods give limited attention to accurately learning and explicitly inferring cross-series dependencies. In this direction, SARAD (Dai et al., 2024) uses attention over transposed input windows to better capture cross-series dependencies, improving localization over other Transformer-based methods (Song et al., 2023). AERCA (Han et al., 2025) derives an exogenous variable for each series and localizes anomalies from deviations from normal training behavior. More recently, ALoRa-Loc (Shimillas et al., 2026) derives weights that explain the cross-series dependencies learned by a Transformer-based model and uses them to associate anomalies to specific time series.

Cross-Series Dependency Learning: Classical approaches include VAR (Brockwell & Davis, 1991), which models interactions through linear autoregressive coefficients, and GVAR (Marcinkevičs & Vogt, 2021), which extends this formulation to nonlinear dependencies using neural networks. DYNOTEARS (Pamfil

et al., 2020) learns sparse lagged and contemporaneous dependencies within a dynamic Bayesian network. PCMCI (Runge et al., 2019) identifies lagged dependencies through variable selection and conditional-independence testing, while PCMCI+ (Runge, 2020) additionally captures contemporaneous relationships.

Other neural approaches infer dependency structures through specialized architectural mechanisms. TCDF (Nauta et al., 2019) uses attention to identify candidate input series for each target and validates the resulting relationships through permutation testing, whereas eSRU (Khanna & Tan, 2020) applies group-wise regularization to recurrent units to obtain a sparse dependency structure. cMLP and cLSTM (Tank et al., 2022) employ sparsity-regularized component-wise neural predictors to identify relevant input series for each target, with cLSTM additionally capturing longer-range temporal dependencies. More recently, CUTS (Cheng et al., 2023) jointly optimizes data imputation and a sparsity-regularized adjacency matrix to determine which past series contribute to predicting each variable. AERCA (Han et al., 2025) combines an encoder–decoder reconstruction framework with Granger causality to learn cross-series dependencies.

3 PRELIMINARIES

Rosenblatt Transformation: Let $X = (X^{(1)}, \ldots, X^{(d)}) : \Omega \to \mathbb{R}^d$ be a d-variate random vector ($X^{(i)} \in \mathbb{R}$) with an absolute continuous distribution $F(X^{(1)}, \ldots, X^{(d)})$. The Rosenblatt (1952) transform is given by

$$\begin{aligned} u^{(1)} &= F_1(x^{(1)}), \\ u^{(2)} &= F_{2|1}(x^{(2)} \mid x^{(1)}), \\ &\vdots \\ u^{(d)} &= F_{d|1:d-1}(x^{(d)} \mid x^{(1)}, \ldots, x^{(d-1)}). \end{aligned}$$

Here, F_1 denotes the marginal distribution function of $X^{(1)}$, and $F_{j|1:j-1}$ denotes the conditional distribution function of $X^{(j)}$ given $(X^{(1)}, \ldots, X^{(j-1)})$.

Proposition 1. *(Rosenblatt, 1952) The transformed vector $U = T(X) = (U^{(1)}, \ldots, U^{(d)})$ is uniformly distributed on the d-dimensional unit hypercube, $U \sim \mathrm{Unif}([0,1]^d)$. Consequently, its components $U^{(1)}, \ldots, U^{(d)}$ are mutually independent, with $U^{(j)} \sim \mathrm{Unif}(0,1),\ j = 1, \ldots, d$.*

MTS Data: Let $\{X_t\}_{t=1}^N$, with $X_t = (X_t^{(1)}, \ldots, X_t^{(d)}) \in \mathbb{R}^d$, denote a d-variate time series, where $X_t^{(i)}$ is the value of variable i at time t. Probabilistically, an MTS is a time-indexed sequence of random vectors.

4 METHODOLOGY

Traditionally, the Rosenblatt transformation has been used as a diagnostic tool for assessing the goodness-of-fit of probabilistic models (Brockwell, 2007; Kheifets, 2018; Hofert & Mächler, 2014). In this work, this classical diagnostic principle is reformulated as a learning mechanism for MTS data through NeuralRos, a model for jointly learning cross-series dependencies and localizing anomalies. The NeuralRos model uses a history-conditioned Rosenblatt transform with a learnable variable ordering and is designed to handle mixed continuous–discrete time series. Section 4.1 introduces this formulation and its mixed-type extension. Section 4.2 then presents the Learnable Rosenblatt Ordering (LRO) module, which learns the ordering of the variables in the conditional factorization, while Section 4.3 describes the neural conditioners used to model the corresponding conditional distributions. The resulting factorization, together with the theoretical uniformity and independence properties established in Proposition 1, guides the learning procedure described in Section 4.4. Finally, Section 4.5 shows how the learned conditional structure is used for cross-series dependency learning and anomaly localization.

4.1 NEURALROS FORMULATION

Learning from time-series data requires conditioning on past observations. The information available from the past T observations before observing X_t is represented by the sigma-algebra $\mathcal{H}_{t-1} = \sigma(X_{t-1}, \ldots, X_{t-T})$. Conditioning on $\mathcal{H}_{t-1}$ incorporates temporal context into the Rosenblatt transform. For a fixed variable ordering, the history-conditioned Rosenblatt transformation at time t is

$$U_t^{(1)} = F_{1|\mathcal{H}_{t-1}}\left(X_t^{(1)} \mid \mathcal{H}_{t-1}\right),\ U_t^{(i)} = F_{i|\mathcal{H}_{t-1},1:i-1}\left(X_t^{(i)} \mid \mathcal{H}_{t-1}, X_t^{(1)}, \ldots, X_t^{(i-1)}\right),\ i = 2, \ldots, d.$$

The above formulation assumes a natural processing order for the time series, from $X_t^{(1)}$ first to $X_t^{(d)}$ last. Although any ordering is theoretically valid, the chosen order can affect the ease with which dependencies are learned. Rather than fixing this order a priori, NeuralRos learns the processing order across transformation stages, allowing the factorization to adapt to the underlying dependency structure. The learnable ordering vector is denoted by

$$\pi = (\pi_1, \ldots, \pi_d) \in \mathbb{R}^d.$$

The value $\pi_i = j$ means that the j-th time series is processed at the i-th stage of Rosenblatt's transformation. At this stage, its conditional distribution is estimated using the historical information $\mathcal{H}_{t-1}$ together with the current-time observations of the series that appear earlier in the ordering, namely $X_t^{(\pi_1)}, \ldots, X_t^{(\pi_{i-1})}$. Thus, the NeuralRos model with a learnable ordering is defined by

$$U_t^{(\pi_1)} = F_{\pi_1|\mathcal{H}_{t-1}}\left(X_t^{(\pi_1)} \mid \mathcal{H}_{t-1}\right),\ U_t^{(\pi_i)} = F_{\pi_i|\mathcal{C}_t^{\pi_i}}\left(X_t^{(\pi_i)} \mid \mathcal{C}_t^{\pi_i}\right), \quad i = 2, \ldots, d, \tag{1}$$

where $\mathcal{C}_t^{\pi_i} = \mathcal{H}_{t-1} \vee \sigma(X_t^{(\pi_1)}, \ldots, X_t^{(\pi_{i-1})})$. Here, $\vee$ denotes the union of sigma-algebras, so $\mathcal{C}_t^{\pi_i}$ contains both the past history and the current-time variables preceding $X_t^{(\pi_i)}$ in the learned ordering.

Mixed-Type Time Series: The transformation in Eq. (1) applies directly only to continuous-valued time series. This limitation is important in practice, since real-world MTS often contain a mixture of continuous and discrete series. To handle such mixed data, NeuralRos builds on Brockwell (2007) and introduces an auxiliary variable $\widetilde{U}_t^{(\pi_i)} \sim \mathrm{Uniform}(0,1)$ at each transformation step. For discrete variables, this auxiliary variable randomizes the transformed value within the jump of the conditional CDF associated with the observation. As a result, each transformed component $U_t^{(\pi_i)}$ remains uniformly distributed on $[0,1]$. For $i = 1, \ldots, d$, the transformation is defined as:

$$U_t^{(\pi_i)} = F_{\pi_i|\mathcal{C}_t^{\pi_i}}\left(X_t^{(\pi_i)-} \mid \mathcal{C}_t^{\pi_i}\right) + \tilde{U}_t^{(\pi_i)}\left[F_{\pi_i|\mathcal{C}_t^{\pi_i}}\left(X_t^{(\pi_i)} \mid \mathcal{C}_t^{\pi_i}\right) - F_{\pi_i|\mathcal{C}_t^{\pi_i}}\left(X_t^{(\pi_i)-} \mid \mathcal{C}_t^{\pi_i}\right)\right]. \tag{2}$$

The quantity $F_{\pi_i|\mathcal{C}_t^{\pi_i}}(X_t^{(\pi_i)-} \mid \mathcal{C}_t^{\pi_i}) = \mathbb{P}(X_t^{(\pi_i)} < x_t^{(\pi_i)} \mid \mathcal{C}_t^{\pi_i})$ denotes the left limit of the conditional CDF. For a continuous variable, $F_{\pi_i}(X^{(\pi_i)-} \mid C_t^{\pi_i}) = F_{\pi_i}(X^{(\pi_i)} \mid C_t^{\pi_i})$, so the expression reduces to the transform of Eq. (1). Thus, NeuralRos handles both types of time series. For further details, see Appendix C.

4.2 LEARNABLE ROSENBLATT ORDERING (LRO)

Ordering Score Estimation: Let $\mathbf{X}_{[t]} = [X_{t-1}, \ldots, X_{t-T}] \in \mathbb{R}^{d\times T}$ denote the history window, where each row $X_{[t]}^{(i)} \in \mathbb{R}^T$ contains the T time steps of the i-th time series. To guide the learning of the ordering vector, the LRO module uses a variant of multi-head attention (MHA) (Vaswani et al., 2017). For each attention head, Query and Key representations are constructed from the history window. Unlike standard MHA, the Query is computed from the raw time series histories, while the Key is computed after applying a head-specific component-wise transformation. For head h, they are defined as

$$Q_{[t]}^{(h)} = X_{[t]}W_Q^{(h)} \in \mathbb{R}^{d\times T}, \qquad K_{[t]}^{(h)} = \Phi_h(X_{[t]})W_K^{(h)} \in \mathbb{R}^{d\times T}, \quad h = 1, \ldots, H\ , \tag{3}$$

where $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{T\times T}$. Here, $\Phi_h(\cdot)$ denotes a head-specific element-wise transformation applied only to the key representation ($K_{[t]}^{(h)}$), while the query is obtained through a linear projection of the raw data. This construction allows the subsequent Query–Key similarity computation to compare each raw time series with transformed representations of the other time series. The first head is defined as the identity mapping, $\Phi_1(X_{[t]}) = X_{[t]}$, to capture linear dependencies, while the remaining heads use nonlinear transformations. In this work, polynomial transformations are used, i.e., $\Phi_h(X_{[t]}) = X_{[t]}^{\odot h}$, where $\odot h$ denotes element-wise exponentiation. The attention matrix for each head is then computed as

$$A_{[t]}^{(h)} = \overline{Q}_{[t]}^{(h)} \left(\overline{K}_{[t]}^{(h)}\right)^\top, \quad A_{[t]}^{(h)} = [A_{[t],i,j}^{(h)}]_{i,j=1}^{d} \in [-1,1]^{d\times d},$$

where $\overline{Q}_{[t]}^{(h)}$ and $\overline{K}_{[t]}^{(h)}$ denote the row-normalized query and key matrices, respectively. The attention asymmetry provides directional ordering evidence. For each head h, the directional score is defined as $D_{[t],i\to j}^{(h)} = A_{[t],j,i}^{(h)} - A_{[t],i,j}^{(h)} \in \mathbb{R}$. A positive value of $D_{[t],i\to j}^{(h)}$ indicates that head h favors placing the i-th series before the j-th series. The head-specific scores are combined through a weighted average to define the probability that series i precedes series j in the transformation of Eq. (1) at time t:

$$p_t(i \prec j) = \sigma\left(\sum_{h=1}^{H} \alpha_h D_{[t],i\to j}^{(h)}\right), \qquad \alpha_h \geq 0, \quad \sum_{h=1}^{H} \alpha_h = 1, \qquad \sigma(z) = \frac{1}{1+\exp(-z)}.$$

The ordering score for series i is then defined by aggregating its pairwise relations with all other series:

$$s_{t,i} = \frac{1}{d-1} \sum_{j\neq i} \left(2p_t(i \prec j) - 1\right) .$$

The window-specific scores are averaged over the N normal windows to obtain the continuous score vector.

$$s = (s_1, \ldots, s_d) \in \mathbb{R}^d, \qquad s_i = \frac{1}{N}\sum_{t=1}^{N} s_{t,i}, \quad i = 1, \ldots, d. \tag{4}$$

Larger s_i values indicate that variable i should appear earlier in the Rosenblatt ordering. Sorting the scores in descending order gives $\pi = \mathrm{argsort}_{\downarrow}(s),\ \pi \in \{1, \ldots, d\}^d$, with corresponding hard permutation matrix $P^{\mathrm{hard}} \in \{0,1\}^{d\times d}$:

$$P_{i,j}^{\mathrm{hard}} = \begin{cases} 1, & \text{if } j = \pi_i, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, d. \tag{5}$$

However, sorting is discrete and non-differentiable, preventing gradient propagation through the hard ordering. To enable gradient-based optimization, we use a straight-through estimator: the hard permutation is used in the forward pass, i.e., when applying the model to compute its output, while a differentiable soft permutation is used in the backward pass, i.e., during training to compute gradients for parameter updates. Let $P^{\mathrm{soft}} \in [0,1]^{d\times d}$ denote a differentiable approximation of the hard permutation matrix. We define

$$P^{\mathrm{ST}} = P^{\mathrm{hard}} + P^{\mathrm{soft}} - \mathrm{stopgrad}\left(P^{\mathrm{soft}}\right). \tag{6}$$

In the forward pass, the term $(P^{\mathrm{soft}} - \mathrm{stopgrad}(P^{\mathrm{soft}}))$ is numerically zero. Therefore, $P^{\mathrm{ST}} = P^{\mathrm{hard}}$. In the backward pass, P^{hard} is treated as constant, and $\mathrm{stopgrad}(\cdot)$ assigns zero gradient to its argument. Therefore, the only term that contributes gradients is P^{soft}.

Computation of the P^{soft} matrix: The ordering scores (Eq. (4)) are mapped to an assignment-score matrix $L \in \mathbb{R}^{d\times d}$, with $L_{r,i} = a_r s_i$, where rows correspond to positions, columns to variables, and $a_1 > a_2 > \cdots > a_d$ are fixed position scalars (e.g., $a_r = d - r + 1$). A differentiable approximation to a permutation matrix is then obtained via Sinkhorn normalization (Sinkhorn, 1964) as $P^{\mathrm{soft}} = \mathrm{Sinkhorn}(\exp(L)) \in [0,1]^{d\times d}$. The resulting matrix is approximately doubly stochastic, meaning that each row and column sums to one, thereby approximating a one-to-one assignment between positions and variables. Details of the Sinkhorn normalization are given in Appendix C.

4.3 NEURAL CONDITIONAL DISTRIBUTION MODELING

Each conditional distribution in Eq. (1) is estimated from the normal training data. Let $\Pi_{i-1} = \{\pi_1, \ldots, \pi_{i-1}\}$ denote the series that precede $X_t^{(\pi_i)}$ in the Rosenblatt ordering. Their current-time values are collected in the masked vector $\left(X_t \odot \mathbf{1}_{\Pi_{i-1}}\right)$, whose j-th component is

$$\left(X_t \odot \mathbf{1}_{\Pi_{i-1}}\right)^{(j)} = \begin{cases} X_t^{(j)}, & j \in \Pi_{i-1}, \\ 0, & j \notin \Pi_{i-1}, \end{cases} \tag{7}$$

where $\odot$ denotes element-wise multiplication. Then, the conditioning information available when modeling $X_t^{(\pi_i)}$ is denoted as:

$$C_t^{\pi_i} = \left[X_t \odot \mathbf{1}_{\Pi_{i-1}}, X_{t-1}, \ldots, X_{t-T}\right]^\top \in \mathbb{R}^{(T+1)\times d}.$$

Here $X_{t-\ell} \in \mathbb{R}^d$, $\ell = 1, \ldots, T$, denotes the full MTS vector at lag ℓ. The conditional distribution is specified according to the variable type. For a continuous-valued time series, a mixture of M Gaussian distributions is used:

$$F_{\pi_i}\left(x_t^{(\pi_i)} \mid C_t^{\pi_i}\right) = \sum_{m=1}^{M} w_{t,i,m}\, \Phi\left(\frac{x_t^{(\pi_i)} - \mu_{t,i,m}}{\sigma_{t,i,m}}\right), \tag{8}$$

where $\Phi(\cdot)$ is the standard normal CDF and $w_{t,i,m}$, $\mu_{t,i,m}$, and $\sigma_{t,i,m}$ are the weight, mean, and scale of component m, respectively. The weights satisfy $w_{t,i,m} \geq 0$ and $\sum_{m=1}^{M} w_{t,i,m} = 1$, while $\sigma_{t,i,m} > 0$. The selection of M is discussed in Appendix A.1.

Neural Conditioner: The distribution parameters are learned by a neural conditioner $G(\cdot)$. Let $C_t^{\pi_i,(j)} = \left(X_t^{(j)}\mathbf{1}_{\{j\in\Pi_{i-1}\}}, X_{t-1}^{(j)}, \ldots, X_{t-T}^{(j)}\right)^\top \in \mathbb{R}^{T+1}$ denote the j-th column of $C_t^{\pi_i}$. It contains the T lagged values of $X^{(j)}$, together with its current value only if variable j has already been processed in the ordering. For a continuous-valued series, the conditioner is defined as

$$G(C_t^{\pi_i}) = \underbrace{b_{\pi_i}}_{\in\mathbb{R}^{3M}} + \sum_{j=1}^{d} \underbrace{\gamma_{\pi_i,j}}_{\in[0,1]} \underbrace{\phi_{\pi_i,j}\left(C_t^{\pi_i,(j)}\right)}_{\in\mathbb{R}^{3M}} \in \mathbb{R}^{3M}, \qquad \phi_{\pi_i,j} : \mathbb{R}^{T+1} \to \mathbb{R}^{3M}. \tag{9}$$

Here, b_{π_i} is a learnable bias vector representing the baseline distribution parameters. Each $\phi_{\pi_i,j}$ is modeled using a KAN-style neural network (Liu et al., 2025) that models the nonlinear contribution of series j to the conditional distribution of $X_t^{(\pi_i)}$. The transformation also includes a normalization step so that the resulting contributions are on comparable scales across series. $\gamma_{\pi_i,j}$ is a learnable scalar that controls the strength of this contribution. Further details on the architecture and normalization of $\phi_{\pi_i,j}$ are provided in Appendix C. The conditioner outputs the raw Gaussian-mixture parameters $(\widetilde{w}_{t,\pi_i,1:M}, \mu_{t,\pi_i,1:M}, \widetilde{\sigma}_{t,\pi_i,1:M})$. Valid weights and scales are obtained as $w_{t,\pi_i,m} = \exp(\widetilde{w}_{t,\pi_i,m}) / \sum_{q=1}^{M} \exp(\widetilde{w}_{t,\pi_i,q})$ and $\sigma_{t,\pi_i,m} = \log(1+\exp(\widetilde{\sigma}_{t,\pi_i,m}))$, ensuring that the weights are non-negative and sum to one and that the scales are positive.

Discrete-valued time series: In many real-world datasets, discrete series are count-valued and therefore take non-negative integer values. For such series, the conditional distribution is modeled using a Negative Binomial distribution:

$$X_t^{(\pi_i)} \mid C_t^{\pi_i} \sim \mathrm{NB}(\mu_{t,\pi_i}, \sigma_{t,\pi_i}),$$

where $\mu_{t,\pi_i}, \sigma_{t,\pi_i} > 0$ are the mean and dispersion parameters, respectively. The conditioner outputs the raw distributional parameters $\widetilde{\theta}_{t,\pi_i} = G(C_t^{\pi_i}) = (\widetilde{\mu}_{t,\pi_i}, \widetilde{\sigma}_{t,\pi_i}) \in \mathbb{R}^2$. The valid positive parameters are obtained as $\mu_{t,\pi_i} = \log(1+\exp(\widetilde{\mu}_{t,\pi_i}))$ and $\sigma_{t,\pi_i} = \log(1+\exp(\widetilde{\sigma}_{t,\pi_i}))$. When a discrete-valued series can take both negative and positive integer values, a distribution with support on $\mathbb{Z}$ is required. In NeuralRos, we use a discretized Gaussian distribution, which provides a simple location–scale parameterization and a tractable conditional CDF for the Rosenblatt transformation (Karlis & Mamode Khan, 2023).

4.4 TRAINING

Training proceeds in three stages. First, the conditional negative log-likelihood is minimized to fit the conditional distributions. Second, uniformity and independence losses are minimized to enforce the Rosenblatt transform properties in Proposition 1. Finally, all objectives are optimized jointly to refine both the conditional distributions and the learned transformation. Let Θ denote all trainable parameters of the model.

Stage 1 - Conditional likelihood. Using the chain-rule factorization induced by the learned Rosenblatt ordering (see Eq. (1)), the fitted conditional joint density is $\widehat{p}_{\pi,\Theta}(X_t \mid \mathcal{H}_{t-1}) = \prod_{i=1}^{d} \widehat{f}_{\pi_i,\Theta}(X_t^{(\pi_i)} \mid \mathcal{C}_t^{\pi_i})$. For N training windows, the negative log-likelihood loss is

$$\mathcal{L}_{\mathrm{NLL}}(\Theta) = -\frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{d} \log \widehat{f}_{\pi_i,\Theta}\left(X_t^{(\pi_i)} \mid \mathcal{C}_t^{\pi_i}\right).$$

Stage 2a - Uniformity property. For N training windows, let $U_{(1)}^{(i)} \le \cdots \le U_{(N)}^{(i)}$ denote the ordered transformed values of the i-th time series obtained from Eq. (1). Under an ideal $\mathrm{Uniform}(0,1)$ distribution, these values should be close to the reference quantiles $q_r = (r-\frac{1}{2})/N$, for $r = 1, \ldots, N$ (David & Nagaraja, 2003). The marginal uniformity loss is therefore defined as

$$\mathcal{L}_{\mathrm{unif}}(\Theta) = \frac{1}{Nd}\sum_{i=1}^{d}\sum_{r=1}^{N}\left(U_{(r)}^{(i)} - \frac{r-\frac{1}{2}}{N}\right)^2.$$

This Cramér–von Mises-type discrepancy (Stephens, 2016) compares the empirical distribution of each Rosenblatt coordinate with the target $\mathrm{Uniform}(0,1)$ distribution, encouraging uniformity over $[0,1]$.

Stage 2b – Independence property. Mutual independence among the transformed variables $U^{(i)}$ is enforced using the dHSIC penalty (Pfister et al., 2018): $\mathcal{L}_{\mathrm{ind}}(\Theta) = \widehat{\mathrm{dHSIC}}\left(U^{(1)}, \ldots, U^{(d)}\right)$. Details of its construction and empirical estimation are provided in Appendix C.

Stage 3 – Joint optimization. The three objectives are jointly optimized: $\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{NLL}} + \mathcal{L}_{\mathrm{unif}} + \mathcal{L}_{\mathrm{ind}}$.

4.5 CROSS-SERIES DEPENDENCY LEARNING AND ANOMALY LOCALIZATION:

From Eq. (9), the weights $\gamma_{\pi_i,j}$ are collected into a dependency matrix

$$\Gamma = [\Gamma_{i,j}]_{i,j=1}^{d} \in [0,1]^{d\times d}, \qquad \Gamma_{i,j} = \gamma_{i,j}. \tag{10}$$

Each entry $\Gamma_{i,j}$ represents the learned dependency from series $X^{(j)}$ to series $X^{(i)}$. Larger values indicate a stronger dependency, while $\Gamma_{i,j} = 0$ means that $X^{(j)}$ has no effect on that conditional distribution.

Localization Scores: After fitting NeuralRos on normal data, anomaly localization is performed by assigning a score to each time series at each test time. For variable i, the model first computes two direct sources of evidence. The first, referred to as the **NeuralRos-L** score, is the per-variable negative log-likelihood,

$$\ell_{t,i} = -\log \widehat{f}_{i,\Theta}\left(X_t^{(i)} \mid \mathcal{C}_t^{i}\right), \tag{11}$$

which measures the compatibility of the observation $X_t^{(i)}$ with its learned conditional distribution given $\mathcal{C}_t^i$. The second, referred to as the **NeuralRos-R** score, is the Rosenblatt tail-surprise score,

$$\rho_{t,i} = -\log\left(2\min\{U_t^{(i)}, 1-U_t^{(i)}\}\right), \qquad \rho_{t,i} \sim \mathrm{Exp}(1), \tag{12}$$

which becomes large when the transformed value is close to either boundary of $[0,1]$. See Appendix C for properties of the score.

5 EXPERIMENTS

Datasets: NeuralRos is evaluated on eleven datasets: eight synthetic datasets with ground-truth dependencies for cross-series identification and three real-world datasets for anomaly localization. The eight synthetic datasets comprise four continuous-valued systems: Linear, Lorenz-96 (Marcinkevičs & Vogt, 2021), Lotka–Volterra, Nonlinear (Absar et al., 2023), and four mixed-type systems with linear (LM1, LM2) and nonlinear (NLM1, NLM2) dynamics. Results for five representative datasets are reported in the main text, with the remainder in Appendix A.2. For anomaly localization, we use SMD (Su et al., 2019), MSDS (Nedelkoski et al., 2020), and SWaT (Mathur & Tippenhauer, 2016). Additional details are provided in Appendix A.2.

Baselines: For cross-series dependency identification, NeuralRos is compared with ten state-of-the-art methods, including VAR (Brockwell & Davis, 1991), cMLP (Tank et al., 2022), cLSTM (Tank et al., 2022), TCDF (Nauta et al., 2019), eSRU (Khanna & Tan, 2020), PCMCI (Runge et al., 2019), PCMCI+ (Runge, 2020), GVAR (Marcinkevičs & Vogt, 2021), CUTS (Cheng et al., 2023), and AERCA (Han et al., 2025). **For anomaly localization**, NeuralRos is compared with Omni-Anomaly (Su et al., 2019), DAEMON (Chen et al., 2023), STAS-SFAS (Shimillas et al., 2025), MEMTO (Song et al., 2023), SARAD (Dai et al., 2024), ε-diagnosis (Shan et al., 2019), RCD (Ikram et al., 2022), AERCA (Han et al., 2025), and ALoRa-Loc (Shimillas et al., 2026). For all baselines, official or publicly available implementations with recommended hyperparameters are used.

Evaluation Metrics: Dependency identification is evaluated using F1 score, Hamming distance (HD), hybrid Hamming distance (HHD), and mean absolute error (MAE). F1 assesses edge recovery, while HD measures thresholded structural errors. HHD, defined in Appendix A.3, is introduced in this study as a stricter metric that also accounts for dependency strengths before thresholding and, as illustrated in Fig. 5, provides a more informative evaluation than HD. MAE is computed as the average absolute difference between the learned and ground-truth dependency weights. See Appendix A.3 for further details. **Anomaly localization** is evaluated using Hit Rate (Su et al., 2019), Normalized Discounted Cumulative Gain (NDCG) (Järvelin & Kekäläinen, 2002), and Interpretation Score (IPS) (Li et al., 2021). These metrics assess whether the highest-ranked variables correspond to the true anomalous time series. For both tasks, each method is evaluated under the same protocol using its best achievable performance, avoiding method-specific thresholding or selection procedures and thereby assessing the quality of the method itself. See Appendix A.3 for details.

Table 1: Cross-series dependency identification using mean HHD, HD, MAE (↓: lower is better), and F1 score (↑: higher is better). Best results are **bold**; second-best are underlined.

Model	Nonlinear				Lotka–Volterra				Lorenz–96				LM1				NLM1			
	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑
VAR	0.53	0.26	0.44	0.47	0.22	0.04	0.26	0.53	0.50	0.36	0.47	0.41	0.42	0.33	0.41	0.45	0.72	0.63	0.71	0.37
cMLP	0.48	0.34	0.47	0.41	0.18	0.05	0.25	0.51	0.28	0.19	0.27	0.47	0.39	0.35	0.40	0.45	0.50	0.38	0.49	0.38
cLSTM	0.82	0.77	0.79	0.38	0.47	0.40	0.47	0.36	0.32	0.23	0.34	0.45	0.44	0.39	0.45	0.41	0.53	0.41	0.53	0.37
TCDF	0.40	0.31	0.42	0.47	0.12	0.02	0.18	0.85	0.35	0.26	0.33	0.43	0.52	0.40	0.51	0.42	0.55	0.41	0.56	0.41
eSRU	0.39	0.27	0.36	0.41	0.15	0.05	0.19	0.42	0.40	0.22	0.38	0.20	0.51	0.43	0.49	0.35	0.64	0.56	0.65	0.36
PCMCI	0.34	0.27	0.35	0.61	0.28	0.11	0.25	0.47	0.60	0.54	0.61	0.37	0.44	0.40	0.46	0.36	0.63	0.57	0.64	0.40
PCMCI+	0.31	0.23	0.33	0.51	0.19	0.02	0.18	0.71	0.34	0.16	0.37	0.50	0.44	0.37	0.41	0.44	0.61	0.53	0.61	0.42
GVAR	0.58	0.49	0.60	0.42	0.14	0.03	0.16	0.79	0.31	0.14	0.29	0.57	0.40	0.40	0.41	0.46	0.63	0.54	0.64	0.39
CUTS	0.63	0.51	0.64	0.36	0.16	0.01	0.15	0.88	0.59	0.41	0.53	0.34	0.18	0.12	0.22	0.8	0.25	0.29	0.22	0.28
AERCA	0.16	0.09	0.08	0.83	0.07	0.03	0.11	0.86	0.12	0.11	0.15	0.80	0.39	0.36	0.19	0.47	0.15	0.11	0.15	0.61
NeuralRos	**0.06**	**0.001**	**0.07**	**0.99**	**0.02**	**0.001**	**0.05**	**0.99**	**0.02**	**0.001**	**0.11**	**0.99**	**0.09**	**0.04**	**0.18**	**0.90**	**0.05**	**0.02**	**0.14**	**0.92**

Dependency Identification Results: All experiments were conducted over 30 runs, and the average performance is reported in Table 1. The standard deviations are provided in Appendix B, Table B. NeuralRos achieves the best overall performance across all five representative datasets. On the continuous datasets, AERCA is the closest competitor, particularly in HD and F1 score, while NeuralRos consistently achieves lower HHD and MAE, indicating more accurate recovery of the dependency structure. On the mixed-type datasets (LM1 and NLM1), NeuralRos outperforms all applicable baselines across all reported metrics.

Ablation studies: The effectiveness of the key components of NeuralRos is evaluated through ablation studies. (i) The Rosenblatt-based learning process is removed while retaining the same conditional model. Its qualitative results are shown in Fig. 1, while quantitative results across all datasets are reported in Appendix B. (ii) The learnable ordering vector is replaced with a fixed ordering. Due to space constraints, the corresponding results are provided in Appendix B. Both components significantly improve model performance, demonstrating their importance to NeuralRos.

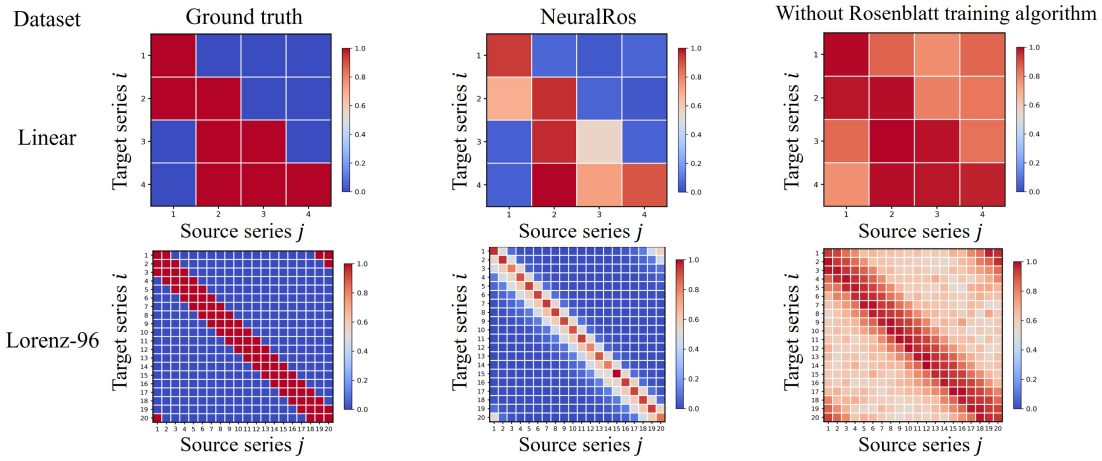

Figure 1: Ablation study of the Rosenblatt-based learning mechanism on the Linear (top) and Lorenz-96 (bottom) datasets. From left to right: the ground-truth dependencies, those learned by NeuralRos, and those learned by the ablated variant, which uses the same conditional distribution model as NeuralRos but without the Rosenblatt-based learning mechanism. The Rosenblatt-based mechanism substantially improves dependency recovery.

Localization Results: All experiments were conducted over 30 runs, and the average performance is reported in Table 2 (standard deviations in Table 5). NeuralRos achieves the best overall anomaly-localization performance across the three datasets. It also consistently outperforms AERCA and ALoRa-Loc, two methods that likewise jointly address dependency learning and anomaly localization.

Table 2: **Localization performance** using mean HR@P%, NDCG@P%, and range-based IPS@P% for $P \in \{100, 150\}$. Best results are **bold**; second-best are underlined.

Method	SMD						MSDS						SWaT					
	HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P	
	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150
OMNI	0.29	0.46	0.24	0.34	0.17	0.26	0.13	0.35	0.09	0.21	0.01	0.03	0.01	0.01	0.01	0.01	0.04	0.04
MEMTO	0.32	0.48	0.26	0.36	0.19	0.28	0.14	0.33	0.10	0.22	0.02	0.04	0.01	0.01	0.01	0.02	0.05	0.05
DAEMON	0.26	0.39	0.31	0.40	0.24	0.26	0.23	0.36	0.27	0.34	0.02	0.03	0.02	0.02	0.02	0.03	0.06	0.07
ε-diagnosis	0.40	0.52	0.42	0.48	0.28	0.46	0.34	0.52	0.34	0.44	0.18	0.28	0.02	0.03	0.02	0.03	0.06	0.07
RCD	0.27	0.31	0.41	0.48	0.24	0.32	0.40	0.65	0.37	0.51	0.16	0.28	0.04	0.05	0.04	0.05	0.08	0.11
STAS-SFAS	0.42	0.51	0.39	0.46	0.55	0.70	0.31	0.48	0.29	0.38	0.06	0.1	0.03	0.04	0.03	0.04	0.11	0.14
SARAD	0.45	0.57	0.47	0.55	0.61	0.74	0.30	0.47	0.31	0.39	0.04	0.06	0.03	0.04	0.03	0.04	0.12	0.16
ALoRa-Loc	**0.49**	0.62	0.51	0.57	0.58	0.80	0.31	0.58	0.33	0.47	0.04	0.05	0.04	0.07	0.04	0.05	**0.14**	0.18
AERCA	0.21	0.25	0.36	0.45	0.13	0.17	0.33	0.56	0.31	0.40	0.08	0.12	0.01	0.02	0.03	0.04	0.03	0.03
NeuralRos-R	0.48	0.61	0.50	0.58	0.50	0.64	0.34	0.59	0.33	0.45	0.13	**0.56**	**0.06**	**0.08**	**0.06**	**0.07**	0.13	**0.22**
NeuralRos-L	**0.49**	**0.64**	**0.51**	**0.60**	**0.72**	**0.86**	**0.44**	**0.70**	**0.44**	**0.58**	**0.25**	0.45	0.05	0.07	0.05	0.05	0.12	0.19

Computational Analysis: We evaluate NeuralRos in terms of model size, training time, and per-sample inference time against SOTA methods. Due to page limitations, the results are reported in Appendix B. As shown in Table 7, the proposed method is computationally efficient and suitable for real-world applications.

Reproducibility statement

To ensure reproducibility, the source code is included in the supplementary material. A README file provides detailed instructions for reproducing the experiments and results reported in this work. Appendix A.1 describes the parameter selection procedure. Links to download the datasets, together with details of the synthetic data generation process, are provided in Appendix A.2. Finally, for each baseline, we use the recommended settings from the original studies.

AI use statement

Large language models (LLMs) were used solely to assist with refining and polishing the writing. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content.

References

Saima Absar, Yongkai Wu, and Lu Zhang. Neural time-invariant causal discovery from time series data. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2023. doi: 10.1109/IJCNN54540.2023.10192004.

Anthony E. Brockwell. Universal residuals: A multivariate transformation. *Statistics Probability Letters*, 77 (14):1473–1478, 2007. ISSN 0167-7152. doi: https://doi.org/10.1016/j.spl.2007.02.008. URL https://www.sciencedirect.com/science/article/pii/S0167715207000715.

Peter J. Brockwell and Richard A. Davis. *Time Series: Data Analysis and Theory*. Springer, New York, 2nd edition, 1991.

Xuanhao Chen, Liwei Deng, Yan Zhao, and Kai Zheng. Adversarial autoencoder for unsupervised time series anomaly detection and interpretation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, pp. 267–275. Association for Computing Machinery, 2023. ISBN 9781450394079.

Yuxiao Cheng, Runzhao Yang, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. CUTS: Neural causal discovery from irregular time-series data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UG8bQcD3Emv.

Yeji Choi, Hyunki Lim, Heeseung Choi, and Ig-Jae Kim. Gan-based anomaly detection and localization of multivariate time series data for power plant. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 71–74, 2020.

Lucas Correia, Jan-Christoph Goos, Philipp Klein, Thomas Bäck, and Anna V. Kononova. Online model-based anomaly detection in multivariate time series: Taxonomy, survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*, 138:109323, 2024. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2024.109323.

Zhihao Dai, Ligang He, Shuang-Hua Yang, and Matthew Leeke. SARAD: Spatial association-aware anomaly detection and diagnosis for multivariate time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Herbert A. David and Haikady N. Nagaraja. *Order Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, NJ, 3rd edition, 2003. ISBN 9780471389262. doi: 10.1002/0471722162.

Zinsou-Max Debaly and Lionel Truquet. Multivariate time series models for mixed data. *Bernoulli*, 29(1): 669–695, 2023.

Xiao Han, Saima Absar, Lu Zhang, and Shuhan Yuan. Root cause analysis of anomalies in multivariate time series through granger causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k38Th3x4d9.

Marius Hofert and Martin Mächler. A graphical goodness-of-fit test for dependence models in higher dimensions. *Journal of Computational and Graphical Statistics*, 23(3):700–716, 2014. doi: 10.1080/10618600.2013.812518. URL https://doi.org/10.1080/10618600.2013.812518.

Chia-Yu Hsu and Wei-Chen Liu. Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 32: 823–836, 2021. ISSN 0956-5515. doi: 10.1007/s10845-020-01591-0.

Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures in microservices through causal discovery. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 31158–31170. Curran Associates, Inc., 2022.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418.

Dimitris Karlis and Naushad Mamode Khan. Models for integer data. *Annual Review of Statistics and Its Application*, 10:297–323, 2023. ISSN 2326-831X. doi: 10.1146/annurev-statistics-032921-022516.

Saurabh Khanna and Vincent Y. F. Tan. Economy statistical recurrent units for inferring nonlinear granger causality. In *International Conference on Learning Representations*, 2020.

Igor L. Kheifets. Multivariate specification tests based on a dynamic rosenblatt transform. *Computational Statistics & Data Analysis*, 124:1–14, 2018. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2018.01.022. URL https://www.sciencedirect.com/science/article/pii/S0167947318300318.

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3220–3230. ACM, August 2021. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467075.

Haoran Liang, Lei Song, Junrong Du, Xuzhi Li, and Lili Guo. Consistent anomaly detection and localization of multivariate time series via cross-correlation graph-based encoder–decoder gan. *IEEE Transactions on Instrumentation and Measurement*, 71:1–10, 2022.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 70367–70413, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/afaed89642ea100935e39d39a4da602c-Paper-Conference.pdf.

Ričards Marcinkevičs and Julia E Vogt. Interpretable models for granger causality using self-explaining neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=DEa4JdMWRHp.

Aditya P. Mathur and Nils Ole Tippenhauer. SWaT: A water treatment testbed for research and training on ICS security. In *Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*, pp. 31–36. IEEE, 2016. doi: 10.1109/CySWater.2016.7469060.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019. ISSN 2504-4990. doi: 10.3390/make1010019. URL https://www.mdpi.com/2504-4990/1/1/19.

Sasho Nedelkoski, Jasmin Bogatinovski, Ajay Kumar Mandapati, Soeren Becker, Jorge Cardoso, and Odej Kao. Multi-source distributed system data for ai-powered analytics. In *Service-Oriented and Cloud Computing*, pp. 161–176, Cham, 2020. Springer International Publishing.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1595–1605. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/pamfil20a.html.

Niklas Pfister, Peter Bühlmann, Bernhard Schölkopf, and Jonas Peters. Kernel-based tests for joint independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):5–31, 2018. doi: 10.1111/rssb.12235.

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23 (3):470–472, 1952. doi: 10.1214/aoms/1177729394.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In Jonas Peters and David Sontag (eds.), *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 1388–1397. PMLR, 03–06 Aug 2020.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996, 2019. doi: 10.1126/sciadv.aau4996. URL https://www.science.org/doi/abs/10.1126/sciadv.aau4996.

Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. ε-diagnosis: Unsupervised and real-time diagnosis of small- window long-tail latency in large-scale microservice platforms. In *The World Wide Web Conference*, WWW '19, pp. 3215–3222, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313653.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios M. Polycarpou. Transformer-based multivariate time series anomaly localization. In *2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES)*, pp. 1–8, 2025. doi: 10.1109/CIES64955.2025.11007637.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964. doi: 10.1214/aoms/1177703591.

Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. MEMTO: Memory-guided transformer for multivariate time series anomaly detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Michael A. Stephens. *Empirical Distribution Function (EDF) Statistics*, pp. 1–13. John Wiley & Sons, Ltd, 2016. ISBN 9781118445112. doi: https://doi.org/10.1002/9781118445112.stat02711.pub2. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat02711.pub2.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837. ACM, July 2019. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330672.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328. JMLR.org, 2017.

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. doi: 10.1109/TPAMI.2021.3065601.

Violeta Teodora Trifunov, Maha Shadaydeh, Björn Barz, and Joachim Denzler. Anomaly attribution of multivariate time series using counterfactual reasoning. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 166–172, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1.

Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, and Shirui Pan. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (10):6775–6794, 2024. doi: 10.1109/TPAMI.2024.3387317.

APPENDIX TO: LEARNING DEPENDENCIES AND LOCALIZING ANOMALIES WITH NEURAL ROSENBLATT TRANSFORM

A EXPERIMENTAL SETTING

A.1 TRAINING DETAILS

Training is performed using the ADAM optimizer with a learning rate of 10^{-4}, and early stopping is applied to prevent overfitting. For the LRO module, the number of attention heads is fixed to $H = 4$ for all datasets considered in this study. Two important hyperparameters are the number of Gaussian mixture components (M) used by the neural conditioners (Eq. (9)) and the size of the input historical window.

Selection of the number of Gaussian mixture components: The number of Gaussian mixture components M in Eq. (8) is selected using the validation data. Candidate values $M \in \{1, 2, 4, 8\}$ are evaluated using the conditional negative log-likelihood (NLL). Lower validation NLL indicates a better fit to the conditional distribution. A larger value of M is selected only when it provides a clear reduction in validation NLL. When the improvement is not significant, the smaller value of M is preferred to avoid unnecessary model complexity. Figures 2 and 3 show the selection of M for the cross-series dependency identification and anomaly localization datasets, respectively. The selection is based only on validation NLL. Test data and downstream task performance are not used during this process.

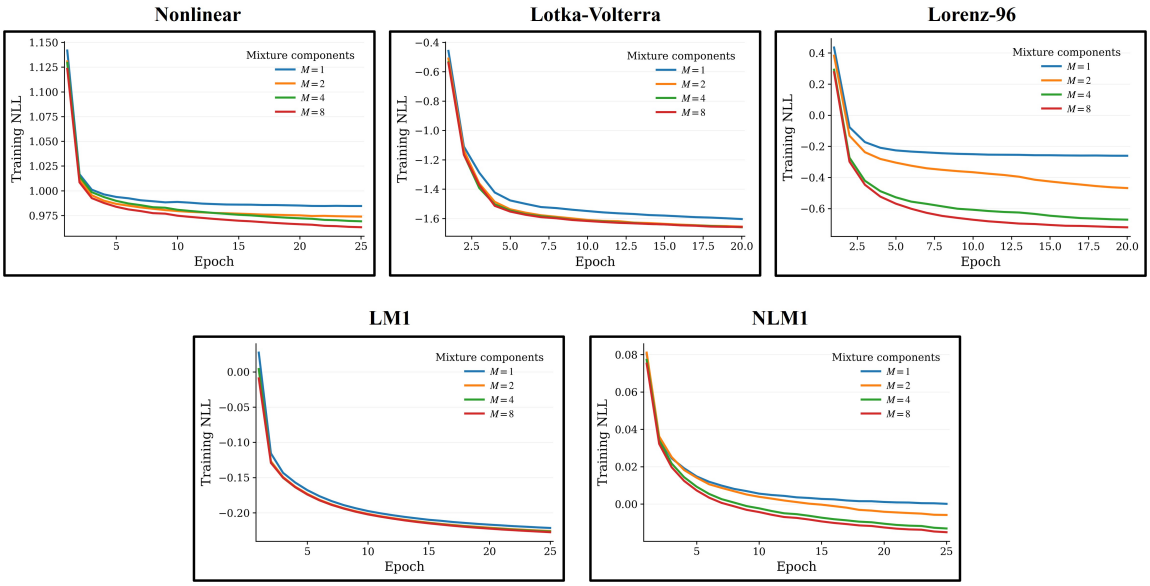

Figure 2: Validation negative log-likelihood (NLL) across training epochs for different numbers of Gaussian mixture components M on the five datasets used in the cross-series dependency experiments in Table 1.

Based on Fig. 2, $M = 2$ is selected for the Linear and Lotka–Volterra datasets, while $M = 4$ is selected for Lorenz-96. For the mixed-type datasets, $M = 1$ is sufficient for the linear mixed-type dataset (LM1) due to its simpler linear dynamics, whereas $M = 4$ is selected for the nonlinear mixed-type dataset (NLM1) to capture its more complex conditional distributions.

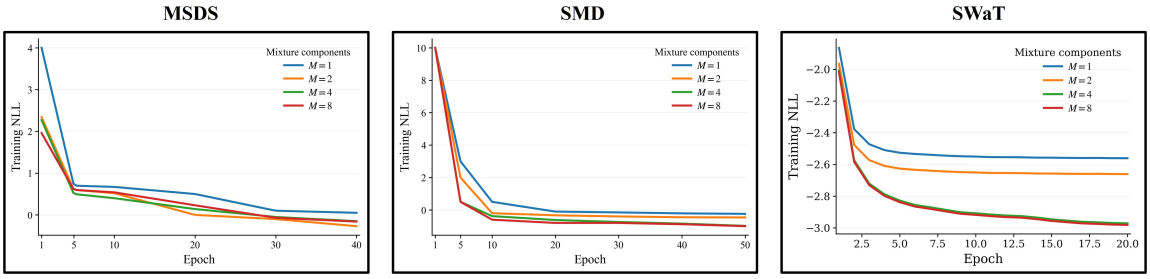

Figure 3: Validation negative log-likelihood (NLL) across training epochs for different numbers of Gaussian mixture components M on the three real-world datasets used in the anomaly localization experiments in Table 2.

Based on Fig. 3, $M = 2$ is selected for MSDS, while $M = 4$ is selected for both SMD and SWaT.

Window size selection: The window size T determines the amount of historical information available to NeuralRos at each time step. To assess the sensitivity of NeuralRos to the history window size T, different values of T are evaluated on the validation set for both cross-series dependency identification and anomaly localization, as shown in Figure 4. For dependency identification, performance is measured using HHD, while anomaly localization is evaluated using HR@100 with the NeuralRos-L score. Similar behavior is observed for NeuralRos-R. Across both tasks, performance remains stable over the tested values of T, indicating that NeuralRos is not sensitive to the precise choice of the history window. Based on these results, $T = 10$ is used for the synthetic dependency identification datasets and $T = 20$ for all real-world anomaly localization datasets.

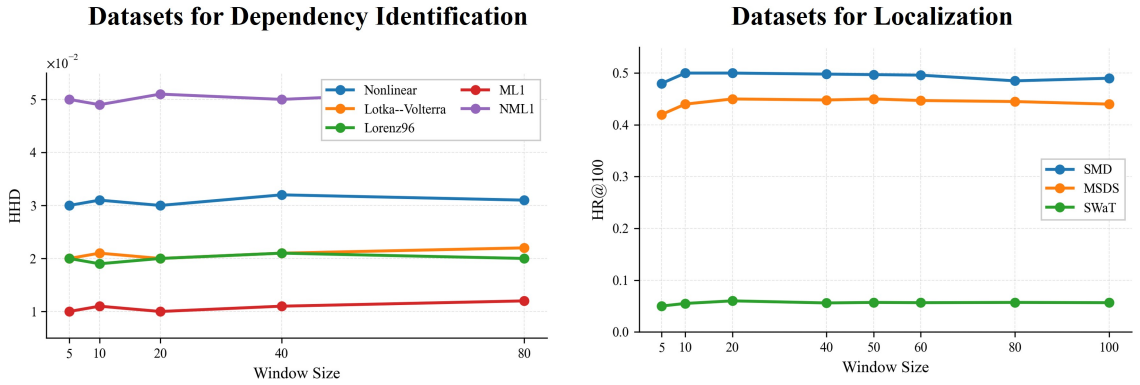

Figure 4: Validation performance of NeuralRos-L for different history window sizes T. The left panel shows results for the synthetic datasets used in cross-series dependency identification, while the right panel shows results for the real-world datasets used in anomaly localization.

A.2 DATASETS

Real-world datasets: The real-world datasets used for evaluating MTS anomaly localization are described below, together with their public download links.

- **Secure Water Treatment (SWaT)** (Mathur & Tippenhauer, 2016): SWaT is a real-world industrial water-treatment dataset collected from a system monitored by 51 sensors and actuators over 11 days. The first 7 days correspond to normal operation, while the remaining 4 days contain 41 manually injected anomalies. `https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/`
- **Server Machine Dataset (SMD)** (Su et al., 2019): SMD is a large-scale server-monitoring dataset collected over five weeks from machines in a major Internet company. It contains 38 system-performance features. `https://github.com/NetManAIOps/OmniAnomaly/tree/master/ServerMachineDataset`
- **Multi-Source Distributed System (MSDS)** (Nedelkoski et al., 2020): MSDS was collected from a cloud-based OpenStack testbed. It contains multi-source monitoring data, including metrics, logs, and traces, under normal and fault-injected conditions. The dataset also provides root-cause labels, making it suitable for anomaly localization. `https://zenodo.org/records/3484801`

Synthetic datasets: Eight synthetic datasets are generated to cover diverse dependency structures, including continuous-only and mixed continuous–discrete settings, for evaluating cross-series dependency identification. All data-generating processes are constructed to be well-defined and stable.

Nonlinear dataset: The Nonlinear dataset contains nonlinear interaction dynamics. Its structural equation is given by

$$X_t = A^\top \sum_{m=1}^{T} \beta_m \cos\left(X_{t-m} + 1\right) + \epsilon,$$

where A is the adjacency matrix of the underlying causal graph, β_m is the regression coefficient, and ϵ is standard Gaussian noise. The noise scale is kept below 1 and is proportional to the dimension d. The nonlinear dependency among time-series variables is introduced through the cosine function, and the adjacency matrix A is generated using the Erdős–Rényi model.

Lorenz-96 dataset. The Lorenz-96 dataset is generated using nonlinear differential equations. The i-th variable is defined as

$$\frac{dx^{(i)}}{dt} = \left(x^{(i+1)} - x^{(i-2)}\right) x^{(i-1)} - x^{(i)} + F, \qquad 1 \le i \le d,$$

with circular boundary conditions: $x^{(0)} := x^{(d)}$, $x^{(-1)} := x^{(d-1)}$, $x^{(d+1)} := x^{(1)}$. The constant F controls the nonlinearity of the generated time series.

Lotka–Volterra dataset. The Lotka–Volterra dataset simulates an ecosystem with prey and predator species. The structural equations are

$$\begin{aligned}
\frac{dx^{(i)}}{dt} &= \alpha x^{(i)} - \beta \sum_{j \in Pa(x^{(i)})} y^{(j)} - \eta \left(x^{(i)}\right)^2, \qquad 1 \le i \le p, \\
\frac{dy^{(j)}}{dt} &= \delta y^{(j)} \sum_{k \in Pa(y^{(j)})} x^{(k)} - \rho y^{(j)}, \qquad 1 \le j \le p, \\
x_t^{(i)} &= x_t^{(i)} + \epsilon_t^{(i)}, \quad y_t^{(j)} = y_t^{(j)} + \epsilon_t^{(j)}.
\end{aligned}$$

Here, $x^{(i)}$ and $y^{(j)}$ denote the population sizes of prey and predator species, respectively. The parameters α, β, η, δ, and ρ control the strengths of the interactions. The parent sets $Pa(x^{(i)})$ and $Pa(y^{(j)})$ define the Granger-causal relationships between prey and predator species. In the experiments, the dataset contains 20

prey species and 20 predator species.

Linear mixed dataset (LM1 & LM2). The linear mixed dataset contains continuous and discrete time series with linear dependency dynamics. We consider two variants, LM1 and LM2, that share the same data-generation process but differ in the proportion of continuous and discrete variables. LM1 consists of ten time series, including six continuous variables $X_t \in \mathbb{R}^6$ and four discrete variables $C_t \in \mathbb{N}^4$. The dependency structure among the continuous variables is generated using a directed Erdős–Rényi graph. The continuous variables are generated as

$$X_t = A \sum_{m=1}^{T} \beta_m X_{t-m} + \epsilon_t,$$

where A is the adjacency matrix of the directed Erdős–Rényi graph, β_m is the regression coefficient associated with lag m, and ϵ_t is Gaussian noise. In our adjacency convention, the rows correspond to target variables and the columns correspond to lagged source variables, such that $A_{ij} = 1$ indicates the dependency $X_{j,t-m} \to X_{i,t}$. To ensure stable dynamics, the lag coefficients are rescaled such that

$$\|A\|_2 \sum_{m=1}^{T} |\beta_m| < 1,$$

where $\|A\|_2$ denotes the spectral norm of A.

The discrete-valued time series are generated using conditional Poisson distributions with a log link, where each discrete variable depends on one lagged continuous variable and exactly one lagged discrete variable. For each discrete variable $C_{k,t}$, the linear predictor is defined as

$$\eta_{k,t} = b_k + \alpha_k X_{p_k,t-1} + \gamma_k \log\left(1 + C_{q_k,t-1}\right), \qquad k = 1, 2, 3, 4.$$

The corresponding Poisson rate is obtained through the exponential link,

$$\lambda_{k,t} = \exp\left(\eta_{k,t}\right).$$

The dependencies are specified by $(p_1, q_1) = (1, 4), \quad (p_2, q_2) = (3, 1), \quad (p_3, q_3) = (5, 2), \quad (p_4, q_4) = (6, 3)$. Thus, $C_{1,t}$ depends on $X_{1,t-1}$ and $C_{4,t-1}$, $C_{2,t}$ depends on $X_{3,t-1}$ and $C_{1,t-1}$, $C_{3,t}$ depends on $X_{5,t-1}$ and $C_{2,t-1}$, and $C_{4,t}$ depends on $X_{6,t-1}$ and $C_{3,t-1}$. The coefficients controlling these dependencies are sampled as

$$b_k, \alpha_k, \gamma_k \sim \mathcal{U}(0.2, 0.8), \qquad k = 1, 2, 3, 4.$$

The exponential link guarantees that the Poisson rate remains strictly positive. The discrete variables are therefore sampled according to

$$C_{k,t} \mid X_{p_k,t-1}, C_{q_k,t-1} \sim \text{Poisson}\left(\lambda_{k,t}\right), \qquad k = 1, 2, 3, 4.$$

LM2 contains a higher proportion of discrete time series compared with LM1. Specifically, LM2 contains five continuous and five discrete time series. Compared with LM1, one additional discrete variable, $C_{5,t}$, is introduced. The discrete variables follow the same Poisson generation mechanism as in LM1, each depending on one lagged continuous and one lagged discrete variable. In particular, the additional variable is defined through $\eta_{5,t} = b_5 + \alpha_5 X_{p_5,t-1} + \gamma_5 \log(1 + C_{q_5,t-1})$, with the corresponding Poisson rate given by $\lambda_{5,t} = \exp(\eta_{5,t})$.

Nonlinear mixed dataset (NLM1 & NLM2). The nonlinear mixed dataset contains both continuous and discrete time-series variables with nonlinear dependency dynamics. We consider two variants, NLM1 and

NLM2, which follow the same data-generation process but differ in the proportion of continuous and discrete variables. We first describe NLM1, which contains seven continuous series $X_t \in \mathbb{R}^7$ and three discrete series $C_t \in \mathbb{N}^3$. The continuous variables follow the nonlinear structural equation

$$X_t = A \sum_{m=1}^{T} \beta_m \cos\left(X_{t-m} + 1\right) + \epsilon_t, \tag{13}$$

where $T = 5$ is the maximum time lag, $A \in \mathbb{R}^{7\times 7}$ is the adjacency matrix of the underlying continuous dependency graph, β_m is the regression coefficient associated with lag m, and ϵ_t is Gaussian noise. In our adjacency convention, rows correspond to target variables and columns correspond to lagged source variables. The nonlinear dependencies among the continuous time series are introduced through the cosine transformation. The adjacency matrix A is generated using a directed Erdős–Rényi graph, and the lag coefficients are rescaled such that

$$\|A\|_2 \sum_{m=1}^{T} |\beta_m| < 1, \tag{14}$$

where $\|A\|_2$ denotes the spectral norm of A.

The discrete-valued time series are generated using conditional Poisson distributions with a log link, where each variable depends on one lagged continuous variable and exactly one lagged discrete variable. For each discrete variable $C_{k,t}$, the log-rate is defined as

$$\eta_{k,t} = b_k + \alpha_k f_k\left(X_{p_k,t-1}\right) + \gamma_k \tanh\left(C_{q_k,t-1}\right), \qquad k = 1, 2, 3,$$

where $X_{p_k,t-1}$ and $C_{q_k,t-1}$ denote the lagged continuous and discrete parents of $C_{k,t}$, respectively. The function f_k introduces a nonlinear dependency on the continuous parent, while $\tanh(\cdot)$ introduces a bounded nonlinear dependency on the discrete parent.

The dependencies are specified by

$$(p_1, q_1) = (2, 3), \qquad (p_2, q_2) = (4, 1), \qquad (p_3, q_3) = (6, 2),$$

with nonlinear transformations

$$f_1(x) = \sin(x), \qquad f_2(x) = \tanh(x), \qquad f_3(x) = \cos(x).$$

Thus, $C_{1,t}$ depends on $X_{2,t-1}$ and $C_{3,t-1}$, $C_{2,t}$ depends on $X_{4,t-1}$ and $C_{1,t-1}$, and $C_{3,t}$ depends on $X_{6,t-1}$ and $C_{2,t-1}$. The coefficients controlling these dependencies are sampled once and kept fixed across all generated trajectories:

$$b_k, \alpha_k, \gamma_k \sim \mathcal{U}(0.2, 0.8), \qquad k = 1, 2, 3. \tag{15}$$

The corresponding Poisson rates are obtained through the exponential link,

$$\lambda_{k,t} = \exp\left(\eta_{k,t}\right), \qquad k = 1, 2, 3, \tag{16}$$

and the discrete variables are sampled according to

$$C_{k,t} \mid X_{p_k,t-1}, C_{q_k,t-1} \sim \text{Poisson}\left(\lambda_{k,t}\right), \qquad k = 1, 2, 3. \tag{17}$$

NLM2 contains a higher proportion of discrete time series compared with NLM1. Specifically, it contains five continuous and five discrete variables. The discrete variables follow the same nonlinear Poisson generation mechanism as in NLM1, but the dependency structure is extended so that each count variable has one lagged continuous parent and exactly one lagged count parent. Specifically, the dependencies are $C_{1,t} \leftarrow \{X_{1,t-1}, C_{5,t-1}\}$, $C_{2,t} \leftarrow \{X_{2,t-1}, C_{1,t-1}\}$, $C_{3,t} \leftarrow \{X_{3,t-1}, C_{2,t-1}\}$, $C_{4,t} \leftarrow \{X_{4,t-1}, C_{3,t-1}\}$, and $C_{5,t} \leftarrow \{X_{5,t-1}, C_{4,t-1}\}$. Their log-rates are defined as $\eta_{k,t} = b_k + \alpha_k f_k(X_{p_k,t-1}) + \gamma_k \tanh(C_{q_k,t-1})$, with corresponding Poisson rates $\lambda_{k,t} = \exp(\eta_{k,t})$, for $k = 1, \ldots, 5$.

Linear dataset. The Linear dataset is generated using linear interaction dynamics among four time-series variables. The structural equations are defined as

$$x_t^{(1)} = a_1 x_{t-1}^{(1)} + u_t^{(1)} + \epsilon_t^{(1)}, \tag{18}$$

$$x_t^{(2)} = a_2 x_{t-1}^{(2)} + a_3 x_{t-1}^{(1)} + u_t^{(2)} + \epsilon_t^{(2)}, \tag{19}$$

$$x_t^{(3)} = a_4 x_{t-1}^{(3)} + a_5 x_{t-1}^{(2)} + u_t^{(3)} + \epsilon_t^{(3)}, \tag{20}$$

$$x_t^{(4)} = a_6 x_{t-1}^{(4)} + a_7 x_{t-1}^{(2)} + a_8 x_{t-1}^{(3)} + u_t^{(4)} + \epsilon_t^{(4)}. \tag{21}$$

The coefficients are sampled as

$$a_i \sim \mathcal{U}\big([-0.8, -0.2] \cup [0.2, 0.8]\big), \tag{22}$$

and the innovation terms follow

$$u_t^{(\cdot)} \sim \mathcal{N}(0, 0.16), \tag{23}$$

where $\epsilon_t^{(\cdot)}$ denotes the anomaly term.

A.3 Evaluation metrics

Cross-Series Dependency Identification Metrics: Cross-series dependency identification is evaluated by comparing the learned dependency structure with the ground-truth dependency graph using four metrics: the F1 score, Hamming Distance (HD), Hybrid Hamming Distance (HHD), and Mean Absolute Error (MAE). Following AERCA (Han et al., 2025), we use the F1 score and HD to evaluate structural edge recovery. In addition, we introduce HHD in this study as a stricter and more informative alternative to HD (see the discussion below and Fig. 5). Finally, MAE is reported as a threshold-free metric that directly evaluates the continuous dependency-score matrix.

Let $\Gamma \in \mathbb{R}^{d\times d}$ denote the continuous dependency-score matrix produced by a method. For NeuralRos, Γ is defined in Eq. (10). Let $A \in \{0,1\}^{d\times d}$ denote the ground-truth dependency graph and $\widehat{A} \in \{0,1\}^{d\times d}$ the corresponding thresholded learned graph.

Fair Evaluation Across Methods: To ensure a fair comparison across all methods and avoid dependence on method-specific threshold-selection procedures, we apply the same score-based evaluation protocol to every method. Specifically, the entries of each learned dependency-score matrix are ranked in descending order, and the top-K scores are selected as predicted dependencies, where $K = \sum_{i=1}^{d}\sum_{j=1}^{d} A_{ij}$ is the number of ground-truth dependencies. The thresholded learned graph is then defined as

$$\widehat{A}_{ij} = \begin{cases} 1, & \Gamma_{ij} \text{ is among the top-}K \text{ scores,} \\ 0, & \text{otherwise.} \end{cases}$$

The same ranking and top-K selection procedure is applied to NeuralRos and all baseline methods. This ensures that the comparison reflects how well each method ranks and identifies the true dependencies, rather than differences in threshold-selection strategies.

F1 Score: The F1 score evaluates edge-recovery accuracy by balancing precision and recall between the thresholded learned graph $\widehat{A}$ and the ground-truth graph A:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$

Hamming Distance (HD): HD measures the structural disagreement between the thresholded learned graph and the ground-truth graph:

$$\mathrm{HD} = \sum_{i=1}^{d}\sum_{j=1}^{d} \mathbf{1}\left\{\widehat{A}_{ij} \neq A_{ij}\right\}.$$

Lower HD indicates fewer structural errors. However, because HD considers only the binary top-K edge decisions, it does not distinguish between methods that recover the same set of edges but assign substantially different scores to the remaining dependencies.

Hybrid Hamming Distance (HHD): HHD extends HD by also using the continuous dependency scores. The top-K entries of Γ are treated as binary predicted edges, while all remaining entries retain their original continuous scores:

$$\widetilde{\Gamma}_{ij} = \begin{cases} 1, & \widehat{A}_{ij} = 1, \\ \Gamma_{ij}, & \widehat{A}_{ij} = 0. \end{cases}$$

The HHD score is then defined as

$$\mathrm{HHD} = \sum_{i=1}^{d} \sum_{j=1}^{d} \left| A_{ij} - \widetilde{\Gamma}_{ij} \right|.$$

Unlike HD, which only compares thresholded edge decisions, HHD also penalizes non-selected entries with nonzero dependency scores. Thus, a lower HHD indicates a cleaner dependency matrix and better agreement with the ground truth. Figure 5 illustrates this difference. Methods A and B recover the same top-K edges and therefore obtain the same HD. However, Method B assigns large scores to several non-existing dependencies, producing a more ambiguous dependency matrix. HHD captures this difference by incorporating the continuous dependency scores in addition to the thresholded edge decisions.

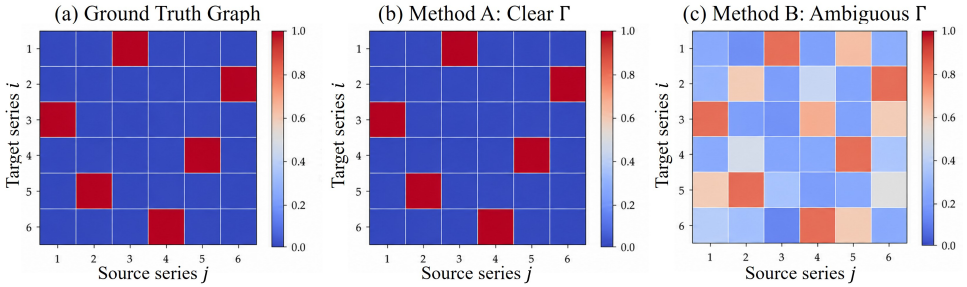

Figure 5: Illustration of the difference between HD and HHD. Both methods recover the same top-K edges after thresholding and obtain $\mathrm{HD}_A = \mathrm{HD}_B = 0$. However, Method A produces a cleaner dependency matrix with near-zero scores for non-existing dependencies, whereas Method B assigns large scores to several non-existing dependencies. HHD captures this difference, with $\mathrm{HHD}_A = 0.00$ and $\mathrm{HHD}_B = 2.80$ in this illustrative example.

Mean Absolute Error (MAE). MAE directly measures the average discrepancy between the continuous learned dependency scores and the ground-truth dependency graph:

$$\mathrm{MAE} = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} |A_{ij} - \Gamma_{ij}|.$$

Unlike F1 and HD, MAE does not require thresholding and evaluates the full continuous dependency matrix. Lower MAE indicates that the learned dependency scores are, on average, closer to the ground-truth graph.

Anomaly Localization Metrics: We evaluate localization performance using three widely adopted metrics: *Hit Rate* (HR), *Normalized Discounted Cumulative Gain* (NDCG), and the *Interpretation Score* (IPS), following recent state-of-the-art anomaly localization methods (Shimillas et al., 2026; Han et al., 2025).

At each time step t, anomaly localization aims to determine the individual time series responsible for the detected anomaly. The corresponding ground truth is represented by a binary vector $G_t \in \{0,1\}^N$, where each element indicates whether the associated time series is anomalous. Specifically, $G_{t,i} = 1$ denotes that the i-th time series is anomalous at time t, while $G_{t,i} = 0$ indicates otherwise.

Fair Evaluation Across Methods: To ensure a fair comparison across the methods reported in Table 2, we apply the same score-based evaluation protocol to all methods. Instead of using method-specific thresholding procedures, the time series are ranked by their anomaly scores, and the top-k series are selected as anomalous. This allows the comparison to reflect the quality of the anomaly scores rather than differences in threshold-selection strategies. For NeuralRos, after computing the anomaly score $l_t^{(i)}$ (or $\rho_t^{(i)}$) for each series using Eq. (11) (or Eq. (12)), the time series are ranked in descending order of their scores to identify those most likely to be responsible for the anomaly. The same ranking procedure is applied to the anomaly scores produced by each baseline method. The top-k series are then selected as

$$A_{t@P\%} = \text{Top-}k \text{ series ranked by } l_t^{(i)}, \quad \text{where } k = \lceil |G_i| \times P\% \rceil$$

where $|G_t|$ denotes the number of truly anomalous time series at time t. For example, if three time series are anomalous, i.e., $|G_t| = 3$, and the evaluation is performed at $P = 150$, then $k = \lceil 3 \times 1.5 \rceil = 5$.

Localization performance is then quantified using the *Hit Rate (HR)* at $P\%$, which measures the fraction of truly anomalous features that appear in the top-k predictions:

$$\text{HR}_{t@P\%} = \frac{|G_i \cap A_{t@P\%}|}{|G_i|} \tag{24}$$

Normalized Discounted Cumulative Gain (NDCG): Let $r_j \in \{0,1\}$ indicate whether the j-th ranked series in $A_{t@P\%}$ is truly anomalous (i.e., belongs to G_i). NDCG measures the quality of the ranking by comparing the discounted cumulative gain (DCG) of the predicted ranking with the ideal discounted cumulative gain (IDCG):

$$\text{NDCG}_{t@P\%} = \frac{\text{DCG}_{t@P\%}}{\text{IDCG}_t}, \qquad \text{DCG}_{t@P\%} = \sum_{j=1}^{k} \frac{r_j}{\log_2(j+1)}, \qquad \text{IDCG}_t = \sum_{j=1}^{|G_t|} \frac{1}{\log_2(j+1)}. \tag{25}$$

NDCG ranges from 0 to 1, with higher values indicating better-ranked localization.

Interpretation Score (IPS): To evaluate anomaly localization at the segment level, we use the *Interpretation Score (IPS)* proposed by Li et al. (2021). For each anomaly segment S_i, we first compute a single score for each time series by taking the maximum anomaly score within the segment:

$$l_{S_i}^{(j)} = \max_{t \in S_i} l_t^{(j)} \quad (\rho_{S_i}^{(j)} = \max_{t \in S_i} \rho_t^{(j)}) \tag{26}$$

We then identify the top-ranked series as the predicted anomalous series A_{S_i}, and compare them against the ground-truth anomalous series G_{S_i}. The IPS measures the average proportion of correctly predicted time series across all segments:

$$\text{IPS} = \frac{1}{N} \sum_{i=1}^{N} \frac{|G_{S_i} \cap A_{S_i}|}{|G_{S_i}|}, \tag{27}$$

where N is the total number of anomalous segments. Each segment is equally weighted in the final score.

B EXPERIMENTS - SUPPLEMENT

Additional experiments: Results on the additional cross-series dependency identification datasets (Linear, LM2, and NLM2) are presented in Table 3, providing a broader evaluation of NeuralRos.

Table 3: Cross-series dependency identification performance on the remaining datasets using mean HHD, HD, MAE, and F1 score. For HHD, HD, and MAE, lower values indicate better performance (↓). Best results are in **bold**; second-best results are underlined.

Model	Linear				LM2				NLM2			
	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑	HHD↓	HD↓	MAE↓	F1↑
VAR	0.08	0.01	0.05	0.97	0.50	0.47	0.36	0.33	0.44	0.39	0.39	0.26
cMLP	0.37	0.23	0.33	0.75	0.49	0.45	0.30	0.34	0.52	0.43	0.54	0.34
cLSTM	0.58	0.31	0.53	0.68	0.61	0.51	0.39	0.28	0.31	0.23	0.24	0.35
TCDF	0.1	0.03	0.1	0.94	0.56	0.44	0.33	0.32	0.57	0.40	0.59	0.38
eSRU	0.07	0.02	0.06	0.96	0.56	0.43	0.47	0.29	0.69	0.59	0.68	0.33
PCMCI	0.12	0.03	0.09	0.97	0.52	0.41	0.44	0.31	0.70	0.59	0.66	0.39
PCMCI+	0.04	0.00	0.04	1.00	0.39	0.34	0.36	0.28	0.66	0.53	0.63	0.40
GVAR	0.21	0.13	0.17	0.86	0.57	0.48	0.35	0.30	0.65	0.56	0.65	0.37
CUTS	0.18	0.10	0.19	0.81	0.24	0.15	0.23	0.65	0.17	0.12	0.15	0.26
AERCA	**0.01**	**0.001**	**0.01**	**1.00**	0.52	0.45	0.32	0.34	0.24	0.20	0.19	0.43
NeuralRos	**0.01**	**0.001**	**0.01**	**0.99**	**0.06**	**0.001**	**0.18**	**0.95**	**0.02**	**0.04**	**0.17**	**0.94**

Standard deviations: Standard deviations for cross-series dependency identification and anomaly localization are reported in Tables B and 5, respectively.

Table 4: **Standard deviations of cross-series dependency identification scores** for the metrics HHD, HD, F1 score, and MAE over 30 runs.

Model	Nonlinear				Lotka–Volterra				Lorenz–96				LM1				NLM1			
	HHD	HD	F1	MAE	HHD	HD	F1	MAE	HHD	HD	F1	MAE	HHD	HD	F1	MAE	HHD	HD	F1	MAE
VAR	0.2	0.1	0.2	0.3	0.1	0.01	0.01	0.1	0.2	0.2	0.2	0.3	0.2	0.2	0.2	0.3	0.2	0.1	0.2	0.2
cMLP	0.2	0.2	0.1	0.2	0.1	0.01	0.01	0.1	0.1	0.06	0.03	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
cLSTM	0.1	0.01	0.01	0.1	0.5	0.4	0.2	0.5	0.1	0.04	0.05	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.2
TCDF	0.1	0.07	0.1	0.2	0.1	0.01	0.03	0.1	0.04	0.01	0.01	0.04	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2
eSRU	0.1	0.07	0.2	0.2	0.1	0.01	0.04	0.1	0.03	0.01	0.02	0.04	0.1	0.06	0.1	0.1	0.2	0.1	0.1	0.2
PCMCI	0.2	0.2	0.1	0.2	0.1	0.01	0.03	0.1	0.1	0.02	0.01	0.1	0.2	0.2	0.1	0.2	0.2	0.1	0.1	0.1
PCMCI+	0.1	0.1	0.1	0.1	0.1	0.01	0.03	0.1	0.07	0.01	0.02	0.06	0.1	0.1	0.1	0.1	0.1	0.03	0.1	0.1
GVAR	0.2	0.1	0.1	0.1	0.04	0.01	0.01	0.04	0.2	0.1	0.3	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
CUTS	0.1	0.1	0.04	0.1	0.04	0.01	0.03	0.1	0.03	0.01	0.01	0.04	0.04	0.1	0.03	0.04	0.1	0.1	0.04	0.1
AERCA	0.01	0.001	0.06	0.02	0.01	0.001	0.001	0.01	0.01	0.001	0.001	0.01	0.01	0.001	0.001	0.01	0.01	0.001	0.001	0.01
NeuralRos	0.01	0.001	0.001	0.02	0.01	0.001	0.001	0.01	0.01	0.001	0.001	0.02	0.02	0.001	0.02	0.02	0.02	0.001	0.03	0.02

Table 5: **Standard deviations of localization scores** for the localization metrics HR@P%, NDCG@P%, and range-based IPS@P%, with $P \in \{100, 150\}$, over 30 runs.

Method	SMD						MSDS						SWaT					
	HR		NDCG		IPS		HR		NDCG		IPS		HR		NDCG		IPS	
MEMTO	0.02	0.02	0.02	0.03	0.02	0.04	0.03	0.05	0.02	0.04	0.01	0.02	0.003	0.004	0.004	0.003	0.01	0.03
OMNI	0.04	0.04	0.05	0.05	0.03	0.03	0.04	0.05	0.02	0.04	0.02	0.02	0.001	0.003	0.002	0.001	0.001	0.001
STAS-SFAS	0.02	0.03	0.02	0.02	0.04	0.04	0.04	0.04	0.03	0.03	0.02	0.02	0.002	0.003	0.002	0.002	0.01	0.01
ALoRa-Loc	0.07	0.09	0.07	0.09	0.09	0.07	0.05	0.04	0.05	0.05	0.01	0.01	0.001	0.001	0.001	0.001	0.001	0.001
SARAD	0.02	0.02	0.02	0.02	0.05	0.04	0.04	0.07	0.03	0.04	0.01	0.01	0.002	0.005	0.002	0.003	0.01	0.03
DAEMON	0.05	0.08	0.07	0.06	0.08	0.1	0.06	0.06	0.05	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.2
ε-diagnosis	0.02	0.02	0.01	0.02	0.03	0.04	0.03	0.03	0.01	0.01	0.02	0.03	0.003	0.004	0.001	0.002	0.004	0.004
RCD	0.02	0.02	0.03	0.02	0.02	0.04	0.03	0.03	0.02	0.02	0.02	0.04	0.004	0.003	0.002	0.002	0.005	0.005
AERCA	0.01	0.02	0.01	0.02	0.03	0.03	0.02	0.02	0.01	0.01	0.03	0.04	0.03	0.01	0.01	0.01	0.01	0.01
NeuralRos-R	0.04	0.03	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.04	0.03	0.1	0.01	0.01	0.01	0.01	0.03	0.1
NeuralRos-L	0.03	0.02	0.04	0.03	0.03	0.02	0.03	0.03	0.03	0.02	0.03	0.1	0.01	0.01	0.01	0.01	0.01	0.01

Supplementary Ablation Studies:

To complement the qualitative analysis presented in the main manuscript, we provide a quantitative evaluation of the Rosenblatt-based learning mechanism across all datasets. The results offer a systematic assessment of its contribution to cross-series dependency recovery and complement the representative visual examples shown in Fig. 1. Figure 6 provides additional qualitative results for two further datasets, Nonlinear and Lotka–Volterra. In both cases, the dependency graphs learned with the Rosenblatt-based learning mechanism, more closely match the ground-truth structure than those of the ablated variant, providing further visual evidence of the effectiveness of the proposed learning mechanism.

Second, we report quantitative results for the proposed LRO module to evaluate the effectiveness. Specifically, we compare the learned ordering with a fixed natural ordering, thereby isolating the benefit of adapting the variable ordering during training. Table 6 summarizes the two ablation studies. Part (a) reports the effect of Rosenblatt-based training using HD, while part (b) reports the effect of the learnable ordering vector using HHD. The results in Table 6(a) show that Rosenblatt-based training consistently reduces HD across all datasets. This indicates that using the Rosenblatt transformation as a learning signal improves the identification of the true cross-series dependency structure. The results in Table 6(b) show that the learnable ordering vector substantially reduces HHD compared with the fixed natural-ordering variant. This suggests that adapting the variable ordering to the data helps NeuralRos assign dependency weights that are closer to the ground-truth dependency structure.

Table 6: Ablation studies of Rosenblatt-based training and the learnable ordering vector. Results are reported as mean $\pm$ std. Lower HD and HHD values indicate better agreement with the ground-truth dependency structure.

(a) Effect of Rosenblatt-based training

Dataset	With (HD)	w/o (HD)
Linear	**0.001±0.00**	0.20±0.05
Nonlinear	**0.001±0.001**	0.92±0.05
Lotka–Volterra	**0.001±0.001**	0.70±0.05
Lorenz-96	**0.001±0.001**	0.59±0.10

(b) Effect of the learnable ordering vector

Dataset	Ordinary (HHD)	Learnable (HHD)
Linear	0.22±0.02	**0.01±0.001**
Nonlinear	0.65±0.06	**0.03±0.01**
Lotka–Volterra	0.60±0.07	**0.02±0.01**
Lorenz-96	0.38±0.05	**0.02±0.01**

In addition to these quantitative evaluations, further qualitative examples illustrating the effect of the Rosenblatt-based training procedure are provided in Fig. 6.

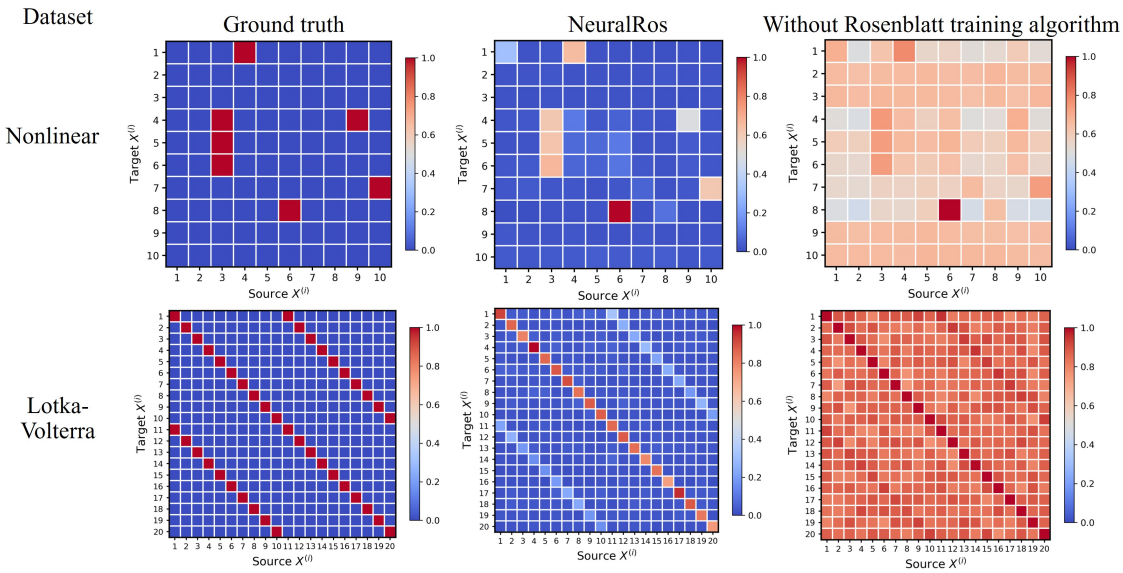

Figure 6: Ablation study of the Rosenblatt-based learning objective on the Nonlinear (first row) and Lotka-Volterra (second row) datasets. In each row, the left, middle, and right panels show the ground-truth dependency graph, the cross-series dependency graph learned by NeuralRos, and the graph learned by the ablated variant, respectively. The ablated variant uses the same conditional distribution model as NeuralRos but removes the Rosenblatt-based learning objective. The comparison shows that using the Rosenblatt transformation as a learning principle improves cross-series dependency learning and produces dependency graphs that better match the ground truth.

Computational Analysis:

The practical efficiency of NeuralRos for the localization task is evaluated by comparing it with four state-of-the-art baselines: MEMTO, SARAD, AERCA, and ALoRa-Loc. The comparison considers model size, expressed as the number of learnable parameters, total training time in seconds, and inference time per sample in milliseconds. The results across the three benchmark datasets are reported in Table 7.

Table 7: Computational efficiency comparison on MSDS, SMD, and SWaT. Params denotes the number of learnable parameters, Inf. denotes inference time per sample in milliseconds, and Train denotes total training time in seconds.

Method	**MSDS** ($d = 10$)			**SMD** ($d = 38$)			**SWaT** ($d = 51$)		
	Params	Inf. (ms)	Train (s)	Params	Inf. (ms)	Train (s)	Params	Inf. (ms)	Train (s)
MEMTO	5.8M	0.56	69	5.9M	1.21	124	5.9M	1.33	500
SARAD	9.5M	76	<u>66</u>	9.6M	0.13	163	9.5M	0.18	644
ALoRa-Loc	**7K**	<u>0.10</u>	70	<u>3.2M</u>	<u>0.12</u>	81	<u>3.2M</u>	0.15	55
AERCA	9.34M	55	934	49.5M	253	166	28.9M	45	210
NeuralRos	<u>116K</u>	**0.06**	300	**2.6M**	**0.08**	400	**2.8M**	**0.11**	182

NeuralRos is one of the most parameter-efficient methods, using substantially fewer learnable parameters than the other methods across all three datasets. Although its training time is not consistently the lowest, further implementation-level parallelization could potentially improve training efficiency. For anomaly lo-

calization, inference efficiency is particularly important for enabling timely diagnosis. **NeuralRos achieves the fastest inference across all three datasets**, highlighting its practical suitability for real-world systems in which rapid anomaly localization is essential. Compared with AERCA, the other method considered here that jointly addresses dependency learning and anomaly localization, NeuralRos uses substantially fewer parameters and achieves markedly lower inference times across all three datasets.

C Methodology Supplement

Proof of the extension of the transformation in Eq. (1) to discrete-valued time-series data (see Eq. (2)).

Consider a random variable X with CDF F, and let $\widetilde{U} \sim \mathrm{Uniform}(0,1)$ be independent of X. Define

$$U = F(X^-) + \widetilde{U}\big[F(X) - F(X^-)\big].$$

To show that $U \sim \mathrm{Uniform}(0,1)$, it is enough to verify that $\mathbb{P}(U \le u) = u, \qquad u \in [0,1]$.

Continuous case. If X is continuous, then $F(X^-) = F(X)$ almost surely. Hence, $U = F(X)$. Thus, the proposed transformation reduces to the standard probability integral transform used in Eq. (1). By Proposition 1, $U \sim \mathrm{Uniform}(0,1)$ (see also Rosenblatt (1952)).

Discrete case. Suppose that X takes ordered values $x_1, x_2, \ldots$, with $\mathbb{P}(X = x_j) = p_j > 0$. For each x_j, define $a_j = F(x_j^-),\ b_j = F(x_j)$. The difference $b_j - a_j$ is the size of the jump of the CDF at x_j, and therefore

$$b_j - a_j = \mathbb{P}(X = x_j) = p_j.$$

By conditioning on $X = x_j$, the transformation becomes

$$U = a_j + \widetilde{U}(b_j - a_j).$$

It follows that $U \mid X = x_j \sim \mathrm{Uniform}(a_j, b_j)$. This can be verified directly by considering any $u \in [a_j, b_j]$:

$$\begin{aligned}\mathbb{P}(U \le u \mid X = x_j) &= \mathbb{P}\Big(a_j + \widetilde{U}(b_j - a_j) \le u\Big) \\ &= \mathbb{P}\left(\widetilde{U} \le \frac{u - a_j}{b_j - a_j}\right) \\ &= \frac{u - a_j}{b_j - a_j}.\end{aligned}$$

It remains to show that U is uniformly distributed on the entire interval $[0,1]$. Fix any $u \in [0,1)$. Then there exists a value x_j such that

$$a_j \le u \le b_j.$$

The event $U \le u$ can now be examined according to whether $X < x_j$, $X = x_j$, or $X > x_j$. First, suppose that $X = x_k < x_j$. Conditional on $X = x_k$, the transformation is

$$U = F(x_k^-) + \widetilde{U}\big[F(x_k) - F(x_k^-)\big].$$

Since $0 \le \widetilde{U} \le 1 \implies F(x_k^-) \le U \le F(x_k)$. Because $x_k < x_j$ and F is nondecreasing: $F(x_k) \le F(x_j^-) = a_j$. Therefore,

$$U \le a_j \le u.$$

Hence, whenever $X < x_j$, the event $U \le u$ occurs with probability one. Second, suppose that $X = x_j$. In this case,

$$U = a_j + \widetilde{U}(b_j - a_j),$$

so U is uniformly distributed over $[a_j, b_j]$. Therefore: $\mathbb{P}(U \le u \mid X = x_j) = \frac{u-a_j}{b_j-a_j}$. Finally, suppose that $X = x_k > x_j$. Again,

$$U \ge F(x_k^-).$$

Since $x_k > x_j \implies F(x_k^-) \ge F(x_j) = b_j$. Because $u < b_j$,

$$U \ge b_j > u.$$

Thus, values $X > x_j$ cannot contribute to the event $U \le u$. Consequently, $U \le u$ can occur only in two mutually exclusive ways: either $X < x_j$, in which case $U \le u$ automatically, or $X = x_j$ and the randomized value falls below u. Hence,

$$\{U \le u\} = \{X < x_j\} \cup \big(\{X = x_j\} \cap \{U \le u\}\big).$$

Since the two events are disjoint,

$$\begin{aligned}\mathbb{P}(U \le u) &= \mathbb{P}(X < x_j) + \mathbb{P}(X = x_j)\mathbb{P}(U \le u \mid X = x_j) \\ &= a_j + (b_j - a_j)\frac{u - a_j}{b_j - a_j} \qquad\qquad = u,\end{aligned}$$

where $\mathbb{P}(X < x_j) = F(x_j^-) = a_j$ and $\mathbb{P}(X = x_j) = F(x_j) - F(x_j^-) = b_j - a_j$.

Thus,

$$\mathbb{P}(U \le u) = u, \qquad 0 \le u < 1.$$

For $u = 1$, it follows trivially that $\mathbb{P}(U \le 1) = 1$. Therefore, $U \sim \mathrm{Uniform}(0, 1)$. □

Computation of the dHSIC objective. Given a minibatch of n transformed observations $\mathbf{U}_a = (U_a^{(1)}, \ldots, U_a^{(d)}),\ a = 1, \ldots, n$, we first construct, for each coordinate i, a Gaussian RBF Gram matrix

$$K_{ab}^{(i)} = \exp\left(-\frac{(U_a^{(i)} - U_b^{(i)})^2}{2\tau_i^2}\right), \qquad a, b = 1, \ldots, n,$$

where $\tau_i > 0$ is the kernel bandwidth. The empirical multivariate independence loss is then computed as

$$\widehat{\mathrm{dHSIC}} = \frac{1}{n^2}\sum_{a,b=1}^{n}\prod_{i=1}^{d}K_{ab}^{(i)} + \prod_{i=1}^{d}\left(\frac{1}{n^2}\sum_{a,b=1}^{n}K_{ab}^{(i)}\right) - \frac{2}{n}\sum_{a=1}^{n}\prod_{i=1}^{d}\left(\frac{1}{n}\sum_{b=1}^{n}K_{ab}^{(i)}\right).$$

We use this quantity as $\mathcal{L}_{\mathrm{ind}} = \widehat{\mathrm{dHSIC}}$. All terms are computed directly from the Rosenblatt coordinates produced by the model; no additional probability distribution is estimated. Since the Gaussian RBF kernel

is characteristic, minimizing this objective drives the empirical joint dependence among all transformed coordinates toward zero and therefore encourages mutual independence.

Computation of the P^{soft} Matrix: The ordering scores (Eq. (4)) are mapped to an assignment-score matrix $L \in \mathbb{R}^{d\times d}$, with rows representing positions and columns representing variables: $L_{r,i} = a_r s_i,\ r, i = 1,\dots,d$, where $a_1 > \cdots > a_d$ are fixed position scalars, e.g., $(a_r = d - r + 1)$. Since L is unconstrained, Sinkhorn (1964) normalization is used to obtain a differentiable soft approximation of a valid permutation matrix (one-to-one variable–position assignment). The scores are first exponentiated as $S^{(0)} = \exp(L) \in \mathbb{R}_+^{d\times d}$, after which rows and columns are alternately normalized for K_{S} iterations:

$$\bar{S}^{(\ell)}_{r,i} = S^{(\ell)}_{r,i} / \sum_{j=1}^{d} S^{(\ell)}_{r,j}, \qquad S^{(\ell+1)}_{r,i} = \bar{S}^{(\ell)}_{r,i} / \sum_{q=1}^{d} \bar{S}^{(\ell)}_{q,i}, \qquad \ell = 1,\dots,K_{\text{S}}. \tag{28}$$

Then, the soft permutation matrix is defined as $P^{\text{soft}} = S^{(K_{\text{S}})} \in [0,1]^{d\times d}$. The alternating normalization makes P^{soft} approximately doubly stochastic, providing a differentiable relaxation of a permutation matrix. Each row corresponds to an ordering position, and each column corresponds to a variable.

Construction and normalization of the KAN-style contribution functions. Each contribution function $\phi_{\pi_i,j}$, appearing in Eq. (9), is constructed using a KAN-style network that models the nonlinear contribution of source series j to the conditional distribution of $X_t^{(\pi_i)}$. KAN-style edge functions can be parameterized using different basis functions, such as B-splines or radial basis functions (RBFs). In this work, we use Gaussian RBFs, which provide a simple and efficiently vectorizable parameterization.

Let

$$h_{\pi_i,j} : \mathbb{R}^{T+1} \to \mathbb{R}^{3M}$$

denote the resulting raw KAN-style mapping. In our implementation, eight Gaussian basis functions are used, together with an additional linear term, allowing each edge function to capture both linear and nonlinear contributions. To make the learned contributions comparable across source series, the output of each $h_{\pi_i,j}$ is normalized using its root-mean-square magnitude over the N normal training windows. Specifically, we define

$$\phi_{\pi_i,j}\left(C_t^{\pi_i,(j)}\right) = \frac{h_{\pi_i,j}\left(C_t^{\pi_i,(j)}\right)}{s_{\pi_i,j}}, \qquad s_{\pi_i,j} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left\|h_{\pi_i,j}\left(C_t^{\pi_i,(j)}\right)\right\|_2^2},$$

where $s_{\pi_i,j}$ is computed from the normal training data. This normalization places the nonlinear contributions on a common scale independently of the corresponding weight $\gamma_{\pi_i,j}$. Consequently, $\phi_{\pi_i,j}$ captures the nonlinear functional form of the contribution, while $\gamma_{\pi_i,j}$ controls its relative magnitude in the neural conditioner.

Rosenblatt tail-surprise score. The score defined in Eq. (12), $\rho_{t,i} = -\log\Big(2\min\{U_t^{(i)}, 1 - U_t^{(i)}\}\Big)$, measures the conditional extremeness of the Rosenblatt-transformed observation. The term $\min\{U_t^{(i)}, 1 - U_t^{(i)}\}$ gives the distance to the nearest boundary of $[0,1]$, so symmetric values such as $U_t^{(i)} = 0.05$ and $U_t^{(i)} = 0.95$ receive the same score. Under a correctly specified model, $U_t^{(i)} \sim \mathcal{U}(0,1)$. Defining $V_{t,i} = 2\min\{U_t^{(i)}, 1 - U_t^{(i)}\}$, for $v \in [0,1]$,

$$\Pr(V_{t,i} \le v) = \Pr\Big(U_t^{(i)} \le \frac{v}{2}\Big) + \Pr\Big(U_t^{(i)} \ge 1 - \frac{v}{2}\Big) = v,$$

and hence $V_{t,i} \sim \mathcal{U}(0,1)$. Since $\rho_{t,i} = -\log V_{t,i}$, we obtain $\Pr(\rho_{t,i} > r) = \Pr(V_{t,i} < e^{-r}) = e^{-r}$ for $r \geq 0$, implying

$$\rho_{t,i} \sim \mathrm{Exp}(1).$$

Thus, the score is symmetric across both tails and provides a calibrated scale on which more extreme conditional observations receive larger values. Moreover, the value of $\rho_{t,i}$ directly indicates how unusual the observation is under the learned normal regime. Since $\rho_{t,i} \sim \mathrm{Exp}(1)$, its tail probability is $\Pr(\rho_{t,i} \geq r) = e^{-r}$; therefore, larger scores correspond to increasingly rare Rosenblatt-transformed observations. For example, $\rho_{t,i} = 2.996$ has a tail probability 0.05, meaning that $U_t^{(i)}$ lies outside the central 95% Rosenblatt region, while $\rho_{t,i} = 4.605$ corresponds to the central 99% region.