# SLRCA: Lightweight Root Cause Analysis of Multivariate Time Series via Parameter Sharing

**Anonymous authors**
Paper under double-blind review

## Abstract

Tracing the root causes of anomalies in multivariate time series is an important and challenging task. Root cause localization requires modeling each variable's normal behavior, but assigning a separate neural network to each variable duplicates temporal prediction parameters, leading to overfitting and parameter proliferation. We propose SLRCA (Shared Lightweight Root Cause Analysis), a lightweight framework for root cause localization through parameter sharing. SLRCA uses vector autoregression (VAR) to model variable-specific linear lagged dependencies, then uses a single temporal convolutional network (TCN) to refine the linear predictions from each variable's residual history. Together, the two components predict each variable's normal behavior; the model uses deviations of actual observations from these predictions to rank candidate root causes. Across four types of synthetic systems and three real-world datasets, SLRCA achieves excellent performance while using only approximately 0.1%–1% as many model parameters as the million-parameter comparison methods.

## 1 Introduction

Root cause localization in multivariate time series is important for diagnosing faults in complex systems, with applications in industrial production, online services, and equipment maintenance. Dynamic dependencies can spread a fault's effects across multiple variables, obscuring its source among the resulting anomalous responses. Existing methods learn temporal patterns and associations between variables from normal data, then diagnose anomalies through deviations from predictions or reconstructions (Zhang et al., 2018; Zhao et al., 2020). Causal approaches further connect root cause localization to changes in conditional distributions or exogenous interventions (Li et al., 2022; Han et al., 2025).

These advances have improved root cause localization amid complex dependencies between variables and established normal-behavior modeling as an important basis for diagnosis. Yet the practical value of a diagnostic model also depends on the resources it requires. Some deep methods use millions of parameters (Shimillas et al., 2026), while industrial monitoring and online services often rely on computational and storage resources shared across multiple tasks. Maintaining effective root cause localization with smaller models therefore has direct practical value and motivates us to reconsider which modeling tasks require neural networks in multivariate diagnosis.

We therefore propose SLRCA, a lightweight root cause localization framework based on parameter sharing. We apply parameter sharing after variable-specific linear modeling, first representing differences in how variables respond to past states, then using a shared network to predict temporal variation left unexplained by the linear model. This division combines each variable's linear coefficients with the same temporal network to construct normal predictions for diagnosis. Changes consistent with normal dependencies can thus be explained from past observations, while deviations from these predictions provide evidence for identifying fault sources.

Our main contributions are as follows:

- We propose SLRCA, a lightweight root cause localization framework that models each variable's normal behavior through variable-specific linear prediction and a shared temporal network. The design retains differences in variable responses while avoiding the neural pa-



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.

rameter growth caused by separate per-variable networks, providing a compact diagnostic model for resource-constrained environments.

- We prove that variables with different residual distributions can share an optimal predictor: sharing does not increase the minimum prediction risk when they admit a common optimal prediction for the same input history. An analytical example further shows how variable-specific linear modeling can alleviate conflicting prediction requirements, providing a basis for designing parameter-efficient models that retain differences between variables.
- Experiments on four types of synthetic systems and three real-world datasets validate SLRCA's localization performance and parameter efficiency. It achieves the best or tied-best AC@1 on six of the seven datasets and uses fewer than 25,000 total model parameters on all three real-world datasets. Architectural comparisons further demonstrate the complementary roles of linear modeling and shared temporal prediction.

## 2 Related Work

Normal-behavior modeling provides a reference for detecting anomalies in multivariate time series. MSCRED and MTAD-GAT learn associations between variables through multiscale correlation matrix reconstruction and graph attention, respectively, and detect anomalies as deviations from normal patterns (Zhang et al., 2018; Zhao et al., 2020). To trace anomaly sources, causal diagnosis methods incorporate relationships between variables: CIRCA and RCD use causal relationships and distribution changes for root cause localization (Li et al., 2022; Ikram et al., 2022). AERCA models the distribution of exogenous variables under normal operation and treats anomalies as exogenous interventions to identify the affected variables and time points (Han et al., 2025). When lagged and instantaneous effects coexist, MDRCA separates the two and ranks root causes using the resulting residuals (Liu & Pi, 2026).

In time series forecasting, parameter sharing allows multiple series to use a common predictor. Research on local and global forecasting examines the relationship between separate per-series models and shared models, providing a theoretical basis for sharing across series (Montero-Manso & Hyndman, 2020). DeepGLO supplies global low-rank forecasts and each series' history to a shared TCN, allowing predictions to use both system-wide information and the state of each series (Sen et al., 2019). PatchTST processes each channel's history separately while sharing embedding and Transformer weights, retaining distinct channel inputs within a shared model (Nie et al., 2023).

We introduce SLRCA, a lightweight framework that is the first to combine variable-specific linear prediction with parameter sharing for root cause localization in multivariate time series. Our theoretical analysis characterizes when different variables can share a prediction function, while an analytical example shows how linear modeling can alleviate conflicting prediction requirements.

## 3 Method

### 3.1 Problem Definition and Framework Overview

Consider a multivariate time series $x_{1:T} = (x_1, \ldots, x_T)$ of length $T$, with $D$ variables and observations $x_t \in \mathbb{R}^D$. Root cause localization seeks to identify fault sources among the $D$ candidate variables. When a monitoring system raises an alert or an event is given for diagnosis, we take an observation window starting at $t_0$ with length $w$ and rank the candidates. We denote this window and its root cause set by

$$\begin{aligned} E &= \{t_0, \ldots, t_0 + w - 1\}, \\ \mathcal{G}_E &\subseteq \{1, \ldots, D\}, \qquad \mathcal{G}_E \neq \varnothing. \end{aligned} \tag{1}$$

$\mathcal{G}_E$ may contain one or more variables and is unknown at diagnosis time. Using the event observations and available history, the model assigns a root cause score to each candidate and ranks them in descending order:

$$\mathbf{S}(E) = \big(S_1(E), \ldots, S_D(E)\big), \qquad \pi_E = \text{argsort}^{\downarrow}\, \mathbf{S}(E). \tag{2}$$

Here, $\pi_E(k)$ denotes the variable at rank $k$. Our goal is to place the root causes in $\mathcal{G}_E$ near the top of the ranking.

We train a model on normal observations $\mathcal{D}_{\text{nor}}$ to predict each variable one step ahead and use its predictions as a reference for diagnosis. As Figure 1 shows, SLRCA first uses a first-order vector autoregression, VAR(1), to model each variable's linear response to the preceding system state. Subtracting this prediction from the observation gives the VAR residual. A shared TCN predicts the next residual from each variable's residual history and system-wide residual statistics, providing a correction to the linear prediction.

Let $z_t$ denote observations standardized using normal data. Adding the two components gives the complete one-step prediction for variable $i$:

$$\widehat{z}_{i,t} = \underbrace{(Az_{t-1})_i}_{\text{VAR linear prediction}} + \underbrace{d_i \widehat{u}_{i,t}}_{\text{shared TCN correction}} . \tag{3}$$

Here, $A \in \mathbb{R}^{D\times D}$ is the coefficient matrix, with row $i$ containing the linear coefficients for variable $i$. The scale of the VAR residuals under normal operation is $d_i > 0$, and $\widehat{u}_{i,t}$ is the shared TCN's prediction of the standardized VAR residual. Multiplication by $d_i$ restores its scale. Each variable retains its own coefficients, scale, and input history, so the complete model produces variable-specific predictions despite sharing the temporal network.

For diagnosis, we normalize the complete prediction errors by their scales under normal operation and aggregate them over window $E$ to obtain the root cause score $S_i(E)$.

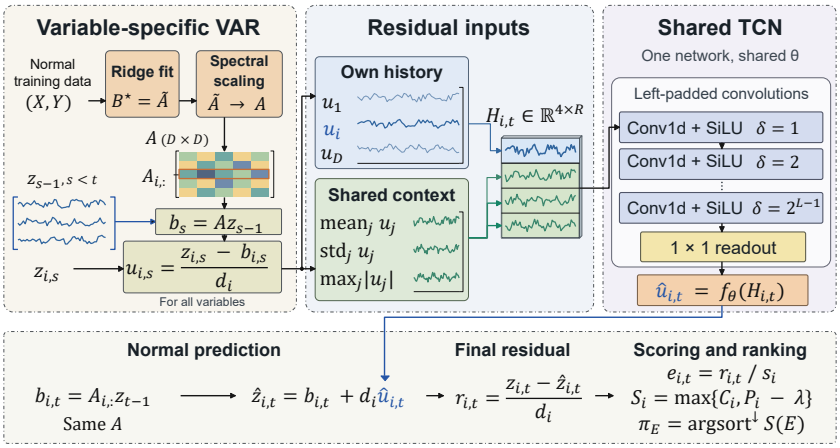



Figure 1: The SLRCA framework.

### 3.2 Variable-Specific Prediction and Conditions for Sharing

Different variables may respond differently to the same system state. We encode these linear responses in the respective rows of $A$, allowing the shared network to predict the temporal variation that the linear model leaves unexplained.

To learn these relationships, we construct $X = [z_{t-1}]$ and $Y = [z_t]$ from adjacent observations within normal segments, with one column per time step. Let $B \in \mathbb{R}^{D\times D}$ be the coefficient matrix to estimate. We obtain $\widetilde{A}$ by ridge regression and scale its spectral radius to obtain $A$:

$$\begin{aligned} \widetilde{A} &= \arg\min_B \left\{ \tfrac{1}{2}\|Y - BX\|_F^2 + \tfrac{\eta}{2}\|B\|_F^2 \right\} \\ &= YX^\top (XX^\top + \eta I_D)^{-1}, \\ A &= \begin{cases} \widetilde{A}, & \rho(\widetilde{A}) \le \rho_{\max}, \\ \dfrac{\rho_{\max}}{\rho(\widetilde{A})}\,\widetilde{A}, & \rho(\widetilde{A}) > \rho_{\max}. \end{cases} \end{aligned} \tag{4}$$

Here, $\|\cdot\|_F$ is the Frobenius norm, $\eta > 0$ is the ridge coefficient, and $I_D$ is the identity matrix. The spectral radius is denoted by $\rho(\cdot)$, with upper bound $0 < \rho_{\max} < 1$. The ridge penalty constrains

coefficient magnitudes, while spectral-radius scaling controls the stability of the linear recurrence and preserves the relative proportions of the coefficients.

We subtract the VAR linear prediction from the observation and normalize by the normal VAR residual scale $d_i$:

$$u_{i,t} = \frac{z_{i,t} - (Az_{t-1})_i}{d_i}. \tag{5}$$

We estimate $d_i$ from the residuals of the scaled VAR model (Appendix B.1). These residuals may still contain temporal structure, which the shared TCN learns to predict.

To predict this variation, we consider both the variable's own residual history and changes across the system. At each historical time step, we combine the variable's VAR residual with cross-variable statistics to form a four-channel input:

$$c_{i,t} = [u_{i,t},\ \text{mean}_j\, u_{j,t},\ \text{std}_j\, u_{j,t},\ \max_j |u_{j,t}|]. \tag{6}$$

The first channel retains the residual of variable $i$. The remaining three channels contain the mean, standard deviation, and maximum absolute residual across all variables $j = 1, \ldots, D$ at that time, providing information about the system as a whole.

The inputs preceding time $t$ form the network history $H_{i,t}$. Let $R$ denote the history window length. With sufficient history, $H_{i,t} = [c_{i,t-R}, \ldots, c_{i,t-1}]$; otherwise, we use the available history with its original starting boundary.

A shared network can produce different predictions from different histories (Montero-Manso & Hyndman, 2020). We now consider two variables with identical historical inputs but different prediction targets and examine the role of VAR. We first retain the four-channel construction in Eq. (6), replace VAR residuals with observations, and use the shared network to predict the next observation directly.

Let $z_{1,t} = z_{3,t-1}$, $z_{2,t} = -z_{3,t-1}$: the first two variables respond to the third in the same and opposite directions, respectively, with a one-step lag. Consider a history window in which the first two variables remain zero, while the third changes at the most recent time step to $z_{3,t-1} = \sqrt{2}$. This change affects the first two variables only at time $t$, so their own histories remain identical. The other three channels use statistics of the same observations and are also identical for both variables. The shared network therefore receives identical four-channel histories and no variable index to distinguish them. It must return the same prediction, although the correct targets are $\sqrt{2}$ and $-\sqrt{2}$ (see Appendix A.3 for the complete construction).

If the VAR uses the exact coefficients of these linear relationships, the residuals of the first two variables are both zero, and a shared network that predicts zero fits both. Their respective VAR coefficients express the opposite responses, leaving a common solution for residual prediction.

We next analyze the conditions for sharing while holding the residual representation and history construction fixed. We compare the minimum expected loss of a predictor that uses only history with that of a predictor that also receives the variable index.

Let $I$ denote the index of the variable being predicted, with $H = H_{I,t}$ and $U = u_{I,t}$ denoting its input history and next standardized VAR residual. Both the analysis and training use the Smooth-L1 loss with $\beta = 1$: for prediction error $v$, when $|v| \le 1$ we use $\ell(v) = v^2/2$, and otherwise $\ell(v) = |v| - 1/2$. For variable $i$ with history $h$, we define the conditional risk (conditional expected loss) of predicting $a \in \mathbb{R}$, together with an optimal prediction, as

$$L_i(a \mid h) = \mathbb{E}[\ell(U - a) \mid H = h, I = i], \qquad m_i(h) \in \arg\min_a L_i(a \mid h). \tag{7}$$

Here, $m_i(h)$ denotes an optimal conditional prediction for variable $i$ given history $h$.

Over all measurable functions, let $R^*_{\text{sh}} = \inf_f \mathbb{E}[\ell(U - f(H))]$ denote the minimum risk using only history, and let $R^*_{\text{id}} = \inf_g \mathbb{E}[\ell(U - g(H, I))]$ denote the minimum risk when the variable index is also available. The following proposition bounds this gap and characterizes when both predictors attain the same minimum risk.

**Proposition 1** (Parameter sharing and prediction risk)**.** *Assume $\mathbb{E}|U| < \infty$. Suppose that the conditional risks for each variable and their mixture admit measurable minimizers, and that there exists*

*a measurable choice of optimal conditional predictions satisfying* $\mathbb{E}[m_I(H)^2] < \infty$. *For any such choice* $m_i$, *the Smooth-L1 loss above satisfies*

$$0 \le \Delta_{\text{sh}} := R^*_{\text{sh}} - R^*_{\text{id}} \le \tfrac{1}{2}\mathbb{E}_H\left[\text{Var}_{I|H}\left(m_I(H)\right)\right]. \tag{8}$$

*Moreover,* $\Delta_{\text{sh}} = 0$ *if and only if, for almost every history* $h$, *all variables with positive conditional probability share a common optimal conditional prediction.*

The variance in the upper bound measures differences among the selected optimal predictions for the same history. Even when the conditional residual distributions differ, sharing preserves the minimum expected loss on normal data if the variables have a common optimal prediction for that history. Together with the example, this result explains our choice to share the network after VAR: the linear model captures variable-specific responses, and the shared network predicts the remaining temporal variation. Appendix A.1 gives the proof.

### 3.3 Shared Residual Prediction

The shared TCN predicts the next residual left by VAR. It uses left-padded dilated temporal convolutions (Bai et al., 2018) with SiLU activations and predicts from past observations only.

The network has $L$ convolutional layers with kernel width $k$. Layer $l$ has dilation $\delta_l = 2^{l-1}$, where $l = 1, \dots, L$. Let $v^{(l)}_{i,s}$ be the representation of variable $i$ at historical time $s$ in layer $l$, with $v^{(0)}_{i,s} = c_{i,s}$. The layers extract temporal features, and a final $1 \times 1$ convolution outputs the prediction:

$$\begin{aligned} v^{(l)}_{i,s} &= \text{SiLU}\left(b^{(l)} + \sum_{q=0}^{k-1} W^{(l)}_q v^{(l-1)}_{i,s-q\delta_l}\right), \\ \widehat{u}_{i,t} &= w_o^\top v^{(L)}_{i,t-1} + b_o. \end{aligned} \tag{9}$$

Here, $W^{(l)}_q$ is the weight matrix of layer $l$ at kernel position $q$, $b^{(l)}$ is the layer bias, and $w_o, b_o$ are the readout weight vector and scalar bias. These parameters form $\theta$, shared by all candidates.

We denote the mapping from history $H_{i,t}$ to prediction $\widehat{u}_{i,t}$ by $f_\theta$. The final residual is

$$r_{i,t} = u_{i,t} - \widehat{u}_{i,t} = u_{i,t} - f_\theta(H_{i,t}). \tag{10}$$

Substituting into Eq. (3) gives $z_{i,t} - \widehat{z}_{i,t} = d_i r_{i,t}$: the final residual $r_{i,t}$ is the complete prediction error normalized by $d_i$.

We estimate the standardization parameters and VAR coefficients using only normal training data and apply the same transformation throughout subsequent learning and diagnosis. We jointly train $f_\theta$ using normal samples from all variables. Let $n$ index normal segments and $\mathcal{I}$ be the set of segment, time, and variable indices used for training. We minimize the one-step prediction loss:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{I}|} \sum_{(n,t,i)\in\mathcal{I}} \ell\Big(u^{(n)}_{i,t} - f_\theta(H^{(n)}_{i,t})\Big). \tag{11}$$

We select the model with the lowest prediction loss on separate normal validation segments, then estimate the scoring scales and pulse penalty on additional held-out normal segments. Appendix B.1 provides the network configurations, training and model selection settings, and calibration details.

Sharing removes the need for a separate set of neural parameters for each variable. If one TCN has $M$ parameters, separate networks require $DM$ neural parameters, whereas SLRCA uses $M$, together with $D^2$ VAR coefficients.

### 3.4 Residual Scoring and Root Cause Ranking

Prediction errors persist even during normal operation, and their scales differ across variables. We therefore calibrate deviations in the fault window against each variable's prediction errors under normal operation. For normal histories, the final residual decomposes into a deviation from the optimal prediction under normal operation and model prediction bias:

$$r_{i,t} = \underbrace{u_{i,t} - m_i(H_{i,t})}_{\text{deviation from optimal prediction}} + \underbrace{m_i(H_{i,t}) - f_\theta(H_{i,t})}_{\text{model prediction bias}}. \tag{12}$$

With the model fixed, we estimate $s_i$ as the sample standard deviation of the final normal residuals, impose a positive floor, and define $e_{i,t} = r_{i,t}/s_i$. While $d_i$ normalizes the VAR residual, $s_i$ measures the normal variation remaining after the complete prediction.

After normalizing the residuals, we distinguish persistent shifts in a common direction from localized short pulses. To derive the scores, we model a candidate's window residuals $e_i = (e_{i,t})_{t\in E}$ under normal conditions as $e_i \sim \mathcal{N}(0, I_w)$, where $I_w$ is the $w$-dimensional identity matrix. A mean-shift vector $v \in \mathbb{R}^w$ has log-likelihood ratio $v^\top e_i - \|v\|_2^2/2$ relative to zero mean.

For a persistent shift, set $v = \mu \mathbf{1}_w$, where $\mu$ is the common shift amplitude and $\mathbf{1}_w$ is the all-ones vector. Maximizing the log-likelihood ratio gives $\widehat{\mu} = w^{-1} \sum_{t\in E} e_{i,t}$; substituting it yields the persistent-shift score. For a short pulse, let $v$ be nonzero only at time $\tau$. For fixed $\tau$, the optimal amplitude is $e_{i,\tau}$; maximizing over the position then yields the pulse score:

$$C_i = \frac{\left(\sum_{t\in E} e_{i,t}\right)^2}{2w}, \qquad P_i = \frac{1}{2} \max_{\tau\in E} e_{i,\tau}^2. \tag{13}$$

$C_i$ accumulates deviations in the same direction, while $P_i$ captures local peaks.

The pulse score also responds to normal spikes, so we use normal windows to calibrate its increment over the persistent-shift score. Across $N$ normal windows matched to the event length, let $C_i^{(n)}$ and $P_i^{(n)}$ be the two scores in window $n$ for variable $i$. We define the maximum increment across variables as $g_n$:

$$g_n = \max_i \left(P_i^{(n)} - C_i^{(n)}\right), \qquad \lambda = \max\left\{0, Q_{0.95}^{\text{higher}}(g_1, \ldots, g_N)\right\}. \tag{14}$$

Here, $Q_{0.95}^{\text{higher}}$ is the empirical 0.95 quantile. The full score is

$$S_i = \max\{C_i,\ P_i - \lambda\}. \tag{15}$$

The pulse term raises the final score only when its increment over the persistent-shift score exceeds $\lambda$. In at least 95% of the normal windows used for calibration, all candidates retain their persistent-shift scores (see Appendix A.2 for the derivation and calibration property). For event $E$, $S_i$ is the score in Eq. (2), denoted by $S_i(E)$. At inference, we fix all fitted parameters, make one-step predictions using the observed history, and sort $S_i$ in descending order to obtain the root cause ranking $\pi_E$.

# 4 Experiments

## 4.1 Experimental Setup

We evaluate SLRCA on four types of synthetic systems and three real-world datasets (Table 1). The synthetic systems are Linear (Marcinkevičs & Vogt, 2021), Nonlinear (Absar et al., 2023), Lotka–Volterra, and Lorenz96 (Marcinkevičs & Vogt, 2021). The real-world datasets, SWaT, MSDS, and WADI, represent a water treatment plant (Mathur & Tippenhauer, 2016), a distributed system (Nedelkoski et al., 2020), and a water distribution system (Mujeeb et al., 2017), respectively.

Table 1: Dataset overview. Synthetic event counts are listed for each of the three generation seeds.

| Dataset | Variables | System characteristics | Fault events |
|---|---|---|---|
| Linear | 4 | Linear lagged dependencies | 100 / 100 / 100 |
| Nonlinear | 6 | Nonlinear lagged dependencies | 100 / 100 / 100 |
| Lotka–Volterra | 40 | Predator–prey dynamics | 100 / 100 / 99 |
| Lorenz96 | 20 | Nonlinear chaotic dynamics | 100 / 100 / 100 |
| SWaT | 51 | Water treatment | 20 |
| MSDS | 10 | Distributed system | 23 |
| WADI | 123 | Water distribution | 11 |

Our seven published deep learning baselines cover causal modeling, graph-based prediction, reconstruction, and variable localization. On all four synthetic systems, we compare against AERCA (Han et al., 2025), GDN (Deng & Hooi, 2021), SARAD (Dai et al., 2024), TranAD (Tuli et al., 2022), and MTAD-GAT (Zhao et al., 2020); real-world comparisons also include MSCRED (Zhang et al., 2018) and ALoRa-Loc (Shimillas et al., 2026). To compare backbone performance and parameter cost, we construct PatchTST-RCA by replacing the TCN with PatchTST (Nie et al., 2023), retaining the VAR frontend and score $S$. Table 2 lists the methods evaluated per dataset. All results come from our experiments, with models selected by normal validation loss (Appendix B.2).

We use AC@1 and MRR to assess how highly root causes rank, and AC@3, AC@5, AC@10, and Avg@10 to measure root cause coverage in longer lists. AC@$k$ is the normalized root cause hit rate among the top $k$ candidates, Avg@10 averages AC@1 through AC@10, and MRR averages the reciprocal rank of the first root cause.

Table 2: Root cause localization (mean ± standard deviation over three seeds; best values in bold; ‡ denotes our architectural control).

| Dataset | Method | AC@1 | AC@3 | AC@5 | AC@10 | Avg@10 | MRR |
|---|---|---|---|---|---|---|---|
| **Linear** | AERCA | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 |
| | GDN | 0.990±0.010 | 0.989±0.012 | **1.000**±0.000 | **1.000**±0.000 | 0.996±0.004 | 0.995±0.005 |
| | SARAD | 0.917±0.040 | 0.946±0.014 | **1.000**±0.000 | **1.000**±0.000 | 0.978±0.005 | 0.955±0.023 |
| | TranAD | 0.963±0.021 | 0.981±0.015 | **1.000**±0.000 | **1.000**±0.000 | 0.992±0.004 | 0.982±0.010 |
| | MTAD-GAT | 0.997±0.006 | 0.993±0.008 | **1.000**±0.000 | **1.000**±0.000 | 0.998±0.002 | 0.998±0.003 |
| | **SLRCA** | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 |
| **Nonlinear** | AERCA | 0.967±0.049 | 0.960±0.012 | 0.992±0.004 | **1.000**±0.000 | 0.984±0.009 | 0.983±0.025 |
| | GDN | 0.977±0.015 | 0.975±0.020 | 0.993±0.001 | **1.000**±0.000 | 0.991±0.005 | 0.988±0.008 |
| | SARAD | 0.930±0.010 | 0.916±0.027 | 0.957±0.020 | **1.000**±0.000 | 0.966±0.010 | 0.963±0.007 |
| | TranAD | 0.993±0.012 | 0.987±0.013 | 0.994±0.005 | **1.000**±0.000 | 0.995±0.005 | 0.996±0.008 |
| | MTAD-GAT | **0.997**±0.006 | **0.989**±0.007 | **0.999**±0.001 | **1.000**±0.000 | **0.997**±0.003 | **0.998**±0.003 |
| | **SLRCA** | **0.997**±0.006 | 0.982±0.014 | 0.997±0.001 | **1.000**±0.000 | 0.994±0.004 | 0.998±0.004 |
| **Lotka–Volterra** | AERCA | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 |
| | GDN | 0.430±0.377 | 0.439±0.408 | 0.519±0.378 | 0.759±0.213 | 0.564±0.340 | 0.595±0.292 |
| | SARAD | 0.883±0.015 | 0.942±0.013 | 0.980±0.010 | **1.000**±0.000 | 0.969±0.004 | 0.941±0.007 |
| | TranAD | 0.916±0.040 | 0.921±0.011 | 0.943±0.013 | 0.998±0.002 | 0.959±0.004 | 0.958±0.019 |
| | MTAD-GAT | 0.853±0.037 | 0.890±0.020 | 0.945±0.009 | **1.000**±0.000 | 0.947±0.010 | 0.924±0.021 |
| | **SLRCA** | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 | **1.000**±0.000 |
| **Lorenz96** | AERCA | **0.900**±0.046 | **0.957**±0.017 | **0.964**±0.011 | **0.990**±0.004 | **0.961**±0.013 | **0.949**±0.023 |
| | GDN | 0.520±0.040 | 0.566±0.014 | 0.582±0.040 | 0.739±0.029 | 0.615±0.016 | 0.707±0.022 |
| | SARAD | 0.783±0.006 | 0.769±0.019 | 0.788±0.021 | 0.904±0.012 | 0.814±0.011 | 0.886±0.006 |
| | TranAD | 0.750±0.036 | 0.760±0.009 | 0.793±0.008 | 0.899±0.005 | 0.807±0.007 | 0.859±0.024 |
| | MTAD-GAT | 0.627±0.015 | 0.649±0.037 | 0.697±0.031 | 0.836±0.014 | 0.715±0.021 | 0.791±0.008 |
| | **SLRCA** | 0.887±0.046 | 0.885±0.050 | 0.879±0.045 | 0.940±0.022 | 0.897±0.036 | 0.942±0.023 |
| **SWaT** | AERCA | 0.100±0.000 | 0.250±0.000 | 0.300±0.050 | 0.483±0.029 | 0.320±0.013 | 0.221±0.006 |
| | GDN | 0.217±0.153 | 0.325±0.200 | 0.375±0.150 | 0.625±0.087 | 0.426±0.146 | 0.338±0.148 |
| | SARAD | 0.150±0.000 | 0.275±0.132 | 0.375±0.100 | 0.542±0.126 | 0.376±0.069 | 0.285±0.024 |
| | TranAD | 0.250±0.000 | 0.433±0.038 | 0.525±0.050 | 0.608±0.029 | 0.489±0.027 | 0.369±0.010 |
| | MTAD-GAT | 0.217±0.058 | 0.392±0.076 | 0.458±0.076 | 0.658±0.161 | 0.476±0.072 | 0.359±0.021 |
| | MSCRED | 0.167±0.029 | 0.367±0.058 | 0.417±0.029 | 0.517±0.052 | 0.393±0.035 | 0.298±0.005 |
| | ALoRa-Loc | 0.050±0.087 | 0.083±0.076 | 0.150±0.087 | 0.242±0.088 | 0.143±0.091 | 0.117±0.083 |
| | PatchTST-RCA‡ | 0.550±0.000 | 0.600±0.050 | 0.733±0.029 | 0.800±0.050 | 0.710±0.022 | 0.622±0.008 |
| | **SLRCA** | **0.667**±0.058 | **0.750**±0.087 | **0.817**±0.058 | **0.875**±0.000 | **0.805**±0.046 | **0.739**±0.062 |
| **MSDS** | AERCA | 0.406±0.223 | **0.899**±0.140 | **0.986**±0.025 | **1.000**±0.000 | **0.881**±0.063 | 0.626±0.157 |
| | GDN | 0.072±0.126 | 0.239±0.210 | 0.522±0.441 | **1.000**±0.000 | 0.500±0.309 | 0.264±0.165 |
| | SARAD | 0.000±0.000 | 0.464±0.251 | 0.725±0.239 | **1.000**±0.000 | 0.685±0.064 | 0.287±0.034 |
| | TranAD | 0.000±0.000 | 0.101±0.176 | 0.457±0.371 | **1.000**±0.000 | 0.474±0.168 | 0.186±0.059 |
| | MTAD-GAT | 0.000±0.000 | 0.333±0.025 | 0.775±0.141 | **1.000**±0.000 | 0.652±0.035 | 0.250±0.011 |
| | MSCRED | 0.000±0.000 | 0.246±0.214 | 0.283±0.188 | **1.000**±0.000 | 0.412±0.105 | 0.213±0.066 |
| | ALoRa-Loc | 0.000±0.000 | 0.283±0.434 | 0.543±0.325 | **1.000**±0.000 | 0.601±0.194 | 0.260±0.142 |
| | PatchTST-RCA‡ | 0.217±0.000 | 0.464±0.025 | 0.565±0.000 | **1.000**±0.000 | 0.601±0.010 | 0.419±0.006 |
| | **SLRCA** | **0.652**±0.000 | 0.667±0.100 | 0.877±0.177 | **1.000**±0.000 | 0.857±0.053 | **0.731**±0.021 |
| **WADI** | GDN | 0.030±0.052 | 0.197±0.131 | 0.258±0.184 | 0.303±0.224 | 0.233±0.159 | 0.168±0.031 |
| | SARAD | 0.091±0.000 | 0.136±0.000 | 0.227±0.000 | 0.227±0.000 | 0.186±0.000 | 0.163±0.003 |
| | TranAD | 0.091±0.000 | 0.258±0.052 | 0.258±0.052 | 0.338±0.023 | 0.267±0.026 | 0.225±0.018 |
| | MSCRED | 0.091±0.000 | 0.318±0.000 | 0.318±0.000 | 0.333±0.026 | 0.292±0.010 | 0.231±0.004 |
| | ALoRa-Loc | 0.091±0.000 | 0.040±0.009 | 0.042±0.023 | 0.040±0.023 | 0.046±0.016 | 0.116±0.008 |
| | PatchTST-RCA‡ | **0.364**±0.000 | **0.364**±0.000 | **0.379**±0.026 | **0.409**±0.000 | **0.383**±0.007 | **0.394**±0.004 |
| | **SLRCA** | **0.364**±0.000 | **0.364**±0.000 | 0.364±0.000 | 0.364±0.000 | 0.364±0.000 | 0.385±0.004 |

### 4.2 Root Cause Localization Performance

SLRCA achieves the best or tied-best AC@1 on six of the seven datasets and the best or tied-best MRR on four (Table 2). On SWaT and MSDS, it improves AC@1 over the strongest published baselines, TranAD and AERCA, by 41.7 and 24.6 percentage points, respectively. It also achieves the highest MRR on both datasets. Ranking root causes higher helps prioritize fault investigation. On MSDS, AERCA achieves higher AC@3 and AC@5, illustrating the distinction between root cause coverage in longer lists and the rank of the first root cause.

With models, normal residual scales, and scoring rules fixed, we evaluate localization on 1,194 new fault events from the four synthetic systems. AC@1 remains unchanged on Linear, Nonlinear, and Lotka–Volterra, and changes on Lorenz96 from 0.887 to 0.893 (Appendix C.4). Together, these results show that SLRCA delivers accurate root cause rankings with a compact model and retains its localization performance on new fault trajectories within the same systems.

### 4.3 Architectural Ablation

To examine how VAR affects parameter sharing, we train shared and independent TCNs on raw observations and VAR residuals (Figure 2). On SWaT and MSDS, VAR with a shared TCN achieves AC@1 of 0.667 and 0.652 under the full score $S$, outperforming the shared model on raw observations. Compared with independent TCNs using the same VAR frontend, it uses only $1/D$ as many neural parameters, yet improves AC@1 by 25.0 and 65.2 percentage points, respectively, and also improves MRR. Additional controls show why the two components work well together. VAR alone achieves AC@1 of 0.250 and 0 on these systems, and adding a four-dimensional learnable variable-index embedding to the shared TCN on raw observations also yields lower AC@1. Neither linear prediction alone nor explicit variable identity matches the combined model.

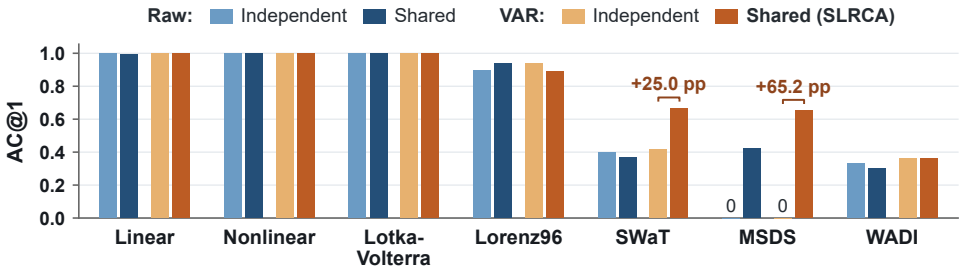



Figure 2: Architectural ablation. Annotations show the AC@1 gains, in percentage points, of shared over independent TCNs on VAR residuals.

The gains vary across systems. On WADI, VAR combined with either shared or independent TCNs achieves the same AC@1 as VAR alone, which achieves higher MRR. Multiple architectures attain perfect scores on Linear and Lotka–Volterra. On Lorenz96, the shared TCN on raw observations outperforms VAR with a shared TCN. For the combined model, the fault-onset residual alone ranks the true root cause first, whereas aggregating over the entire window allows larger subsequent deviations in other variables to displace it in some events. This reveals how window aggregation can obscure root cause evidence present at fault onset.

Differences between variables need not be modeled by separate temporal networks. Separating variable-specific linear responses from shared temporal prediction is key to SLRCA's gains in both localization accuracy and parameter efficiency.

### 4.4 Parameter Efficiency and Inference Cost

SLRCA substantially reduces model size for root cause localization (Figure 3). Its total parameter counts on SWaT, MSDS, and WADI are 9,258, 9,861, and 24,890, respectively. AERCA, MSCRED, and SARAD use millions of parameters in the corresponding configurations, or 129–1,044 times as many. On MSDS, SLRCA uses approximately $1/316$ as many inference parameters as AERCA while achieving higher MRR, demonstrating gains in both model size and localization.

Using the same VAR frontend and scoring rule as PatchTST-RCA, SLRCA uses 53.9% fewer total parameters on SWaT while improving MRR from 0.622 to 0.739. On WADI, the two models have identical AC@1; SLRCA's MRR is 0.009 lower, and it uses 23.7% fewer total parameters. A small shared TCN thus effectively complements linear prediction, maintaining competitive root cause rankings with fewer temporal prediction parameters.

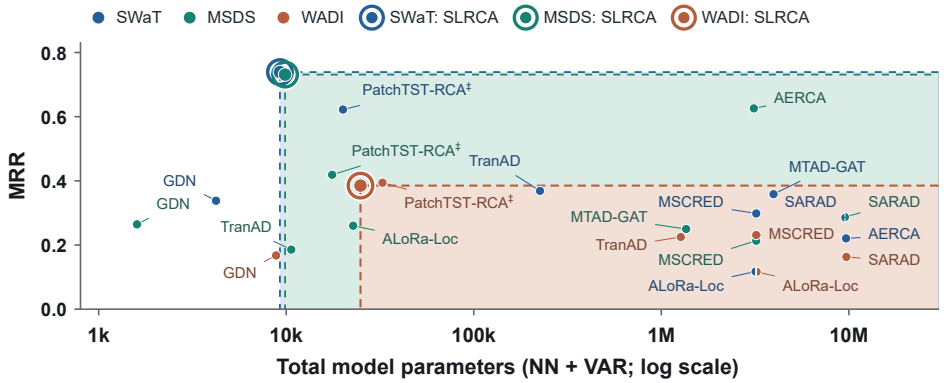



Figure 3: Total model parameters (including VAR) versus root cause localization performance. Each shaded region denotes more parameters and lower MRR than SLRCA on the same dataset; ‡ marks an architectural control.

On an RTX 5070, the mean of per-model median latencies for full event scoring is 0.858 ms for MSDS and 0.913 ms for WADI; Appendix C.2 provides parameter breakdowns and timing settings. By substantially reducing model size while retaining strong localization performance and millisecond-level event scoring, SLRCA lowers the computational requirements for root cause diagnosis in practical monitoring systems.

## 5 Conclusion

We presented SLRCA, a lightweight root cause localization framework that combines linear modeling with shared temporal prediction to achieve competitive localization with a compact model. Our findings show that a shared temporal network can model variable-specific normal behavior. Separating variable-specific modeling from temporal prediction retains differences between variables while reducing neural parameters. Our method offers a new direction for tracing root causes in systems with more complex variable dependencies, while its lightweight design facilitates practical deployment.

## Reproducibility Statement

Appendix A provides the method analysis and proofs, Appendix B describes the experimental settings, and Appendix C reports complete architecture comparisons, cost measurements, and additional results. Each table specifies the number of runs and aggregation method. Detailed configurations, evaluation scripts, and individual run records are retained in the experimental materials.

## AI Use Statement

We used generative AI to assist with manuscript formatting and literature search and organization.

## 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. IEEE, 2023. doi: 10.1109/IJCNN54540.2023.10192004. URL https://doi.org/10.1109/IJCNN54540.2023.10192004.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

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*, volume 37, 2024. doi: 10.52202/079017-1533. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/56ad264ac7448239145606cf4106042f-Abstract-Conference.html.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4027–4035, 2021. doi: 10.1609/aaai.v35i5.16523. URL https://ojs.aaai.org/index.php/AAAI/article/view/16523.

Xiao Han, Saima Absar, Lu Zhang, and Shuhan Yuan. Root cause analysis of anomalies in multivariate time series through granger causal discovery. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6fde96479648d71e4fd9724374bf76eb-Paper-Conference.pdf.

Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures in microservices through causal discovery. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2259. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c9fcd02e6445c7dfbad6986abee53d0d-Abstract-Conference.html.

Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. Causal inference-based root cause analysis for online service systems with intervention recognition, 2022. URL https://arxiv.org/abs/2206.05871.

Jinchang Liu and Dechang Pi. Root cause analysis framework for industrial faults based on mixed causal discovery and dual residual disentanglement. *Measurement*, 2026. doi: 10.1016/j.measurement.2026.122929. URL https://doi.org/10.1016/j.measurement.2026.122929. Available online 18 August 2026; in press.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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 Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*, 2016. URL https://tippenhauer.de/publication/mathur-16-swat/.

Pablo Montero-Manso and Rob J Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality, 2020. URL https://arxiv.org/abs/2008.00444. First preprint 2020; version 3, 26 March 2021.

Chuadhary Mujeeb, Palleti Venkata Reddy, and Aditya Mathur. Wadi: A water distribution testbed for research in the design of secure cyber physical systems. Author presentation at CySWater 2017, 2017. URL https://projects.ics.forth.gr/spl/cyswater2017/_docs/CySWater_Paper6.pdf. Presentation source; not the proceedings paper record.

Sasho Nedelkoski, Jasmin Bogatinovski, Ajay Kumar Mandapati, Soeren Becker, Jorge Cardoso, and Odej Kao. Multi-source distributed system data for ai-powered analytics. In *European Conference on Service-Oriented and Cloud Computing*, pp. 161–176. Springer, 2020. URL https://github.com/snedelkoski/multi-source-observability-dataset.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

Rajat Sen, Hsiang-Fu Yu, and Inderjit Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1905.03806.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios M. Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization, 2026. URL https://arxiv.org/abs/2602.08467.

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6): 1201–1214, 2022. doi: 10.14778/3514061.3514067. URL https://vldb.org/pvldb/vol15/p1201-tuli.pdf.

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, 2018. URL https://arxiv.org/abs/1811.08055.

Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *IEEE International Conference on Data Mining*, 2020. URL https://arxiv.org/abs/2009.02040.

# A Method Analysis and Proofs

## A.1 Risk Analysis of Shared Prediction

We prove Proposition 1 while holding the VAR coefficients and residual scales $A, d_i$, along with the history inputs $H$, fixed. The notation and history boundaries follow Section 3.2; the sampling weights over variables and valid training times match the training objective in Eq. (11). Write $\pi_i(h) = \Pr(I = i \mid H = h)$. The derivative of the Smooth-L1 loss is $\psi(v) = \mathrm{clip}(v, -1, 1)$.

Using the conditional risks $L_i$ and optimal predictions $m_i$ from Eq. (7), let $m(h)$ be a measurable optimal prediction for the mixture conditional risk $\sum_i \pi_i(h) L_i(a \mid h)$. Assume $\mathbb{E}|U| < \infty$ and that the conditional risks admit measurable minimizers. Then $R^*_{\text{sh}}$ and $R^*_{\text{id}}$ are attained by $m(H)$ and $m_I(H)$, respectively. The variance bound additionally requires $\mathbb{E}[m_I(H)^2] < \infty$.

*Proof of Proposition 1.* Taking conditional expectations given the history input yields

$$R^*_{\text{sh}} - R^*_{\text{id}} = \mathbb{E}_H \sum_i \pi_i(H) \big[ L_i(m(H) \mid H) - L_i(m_i(H) \mid H) \big].$$

Each bracketed difference is nonnegative, giving an exact expression for the risk gap and the lower bound in Eq. (8).

For the upper bound, $\psi$ is 1-Lipschitz and satisfies $|\psi| \le 1$. We can therefore differentiate the conditional expectation to obtain

$$L_i'(a \mid h) = -\mathbb{E}[\psi(U - a) \mid H = h, I = i], \qquad |L_i'(a \mid h) - L_i'(b \mid h)| \le |a - b|.$$

The optimal prediction satisfies $L_i'(m_i(h) \mid h) = 0$. Integrating from $m_i(h)$ to $a$ gives

$$L_i(a \mid h) - L_i(m_i(h) \mid h) \le \tfrac{1}{2}(a - m_i(h))^2.$$

Define $\bar{m}(h) = \sum_i \pi_i(h) m_i(h)$. By optimality under the mixture risk, we may compare against $\bar{m}(h)$, giving

$$\begin{aligned} &\sum_i \pi_i(h) \big[ L_i(m(h) \mid h) - L_i(m_i(h) \mid h) \big] \\ &\quad \le \sum_i \pi_i(h) \big[ L_i(\bar{m}(h) \mid h) - L_i(m_i(h) \mid h) \big] \\ &\quad \le \tfrac{1}{2} \sum_i \pi_i(h) (\bar{m}(h) - m_i(h))^2 = \tfrac{1}{2} \operatorname{Var}_{I \mid H = h}(m_I(h)). \end{aligned}$$

Taking the expectation over $H$ yields the variance bound.

Finally, the risk gap is zero if and only if these nonnegative differences vanish almost everywhere for every variable with positive weight, equivalently, if a common optimal prediction exists:

$$\bigcap_{i:\, \pi_i(h) > 0} \arg\min_a L_i(a \mid h) \neq \varnothing \quad \text{for almost every } h.$$

If a common optimal prediction exists, every minimizer of the mixture risk also minimizes each component risk. Conversely, if the risk gap is zero, an optimal shared prediction must belong to this intersection. The Smooth-L1 conditional risk can have multiple optimal predictions. Whether the risk gap is zero therefore depends on the existence of a common optimum, independently of which optimum we select for each variable. □

When optimal predictions are not unique, the bound holds for every measurable selection $\boldsymbol{m} = (m_1, \ldots, m_D)$ satisfying the above conditions. Denoting the set of these selections by $\mathcal{M}$ and taking the infimum gives

$$\Delta_{\text{sh}} \le \frac{1}{2} \inf_{\boldsymbol{m} \in \mathcal{M}} \mathbb{E}_H \big[ \operatorname{Var}_{I \mid H} \big( m_I(H) \big) \big] .$$

### A.2 Derivation of Shift Scores and Normal Calibration

Under the Gaussian model in Section 3.4, consider a persistent shift of amplitude $\mu$ throughout the window and a single-point shift at position $\tau \in E$ with amplitude $a$. Their log-likelihood ratios are

$$\Lambda_C(\mu) = \mu \sum_{t \in E} e_{i,t} - \tfrac{w\mu^2}{2}, \qquad \Lambda_P(a, \tau) = a e_{i,\tau} - \tfrac{a^2}{2}. \tag{16}$$

Maximizing $\Lambda_C$ gives $\widehat{\mu} = w^{-1} \sum_{t \in E} e_{i,t}$. For fixed position $\tau$, the optimal pulse amplitude is $\widehat{a} = e_{i,\tau}$. Substituting these values and maximizing over the pulse position $\tau \in E$ yields $C_i$ and $P_i$ in Eq. (13).

For normal calibration in Eq. (14), sort the window increments as $g_{(1)} \le \cdots \le g_{(N)}$. Under the higher-quantile convention, let $k = 1 + \lceil 0.95(N-1) \rceil$, so that $\lambda = \max(0, g_{(k)})$. At least $k$ windows satisfy $g_n \le \lambda$; hence, every candidate in these windows satisfies

$$P_i^{(n)} - C_i^{(n)} \le g_n \le \lambda \quad \Longrightarrow \quad \max\{C_i^{(n)}, P_i^{(n)} - \lambda\} = C_i^{(n)} \quad \text{for all } i.$$

Since $k/N \ge 0.95$, every candidate's final score equals its persistent-shift score in at least 95% of the normal windows used for calibration.

### A.3 An Analytical Example of VAR and Parameter Sharing

We construct a normal process with population standardization and exact linear coefficients to compare the minimum risks of a shared network predicting observations directly and predicting VAR residuals.

Let $\xi_t$ be independent and identically distributed, with

$$\Pr(\xi_t = 0) = \tfrac{1}{2}, \qquad \Pr(\xi_t = \sqrt{2}) = \Pr(\xi_t = -\sqrt{2}) = \tfrac{1}{4}, \qquad z_t = (\xi_{t-1}, -\xi_{t-1}, \xi_t)^\top.$$

All three variables have zero normal mean and unit variance. For finite history length $R$, replace the VAR residuals in Eq. (6) with raw observations. The inputs for direct sharing are

$$\widetilde{c}_{i,s} = \left[ z_{i,s}, \operatorname{mean}_j z_{j,s}, \operatorname{std}_j z_{j,s}, \max_j |z_{j,s}| \right], \qquad \widetilde{H}_{i,t} = (\widetilde{c}_{i,t-R}, \ldots, \widetilde{c}_{i,t-1}).$$

At a time with sufficient history, consider the event

$$\mathcal{B}_t = \{\xi_{t-R-1} = \cdots = \xi_{t-2} = 0,\ \xi_{t-1} = \sqrt{2}\}, \qquad \Pr(\mathcal{B}_t) = 2^{-R}/4 > 0.$$

On $\mathcal{B}_t$, the first two variables have all-zero observation histories and identical summary channels, so $\widetilde{H}_{1,t} = \widetilde{H}_{2,t}$. Their next-step targets, however, are $\sqrt{2}$ and $-\sqrt{2}$, respectively. A shared predictor without the variable index must produce the same output $a$. Its average loss over the two variables satisfies

$$\tfrac{1}{2}\left[\ell(\sqrt{2} - a) + \ell(-\sqrt{2} - a)\right] \ge \ell(\sqrt{2}) = \sqrt{2} - \tfrac{1}{2} > 0.$$

Convexity and even symmetry of Smooth-L1 give the inequality, with equality for $a \in [1-\sqrt{2}, \sqrt{2}-1]$. A prediction function with access to the variable index can distinguish the two responses: since $\operatorname{mean}_j z_{j,t-1} = \xi_{t-1}/3$, it can output three times this mean for the first variable and minus three times this mean for the second. The third variable is independent of the available history and has an optimal conditional prediction of zero. Under this input representation, a shared predictor using only history therefore has a strictly higher minimum risk than one with additional access to the variable index.

The coefficient matrix for this process is

$$A_\star = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad A_\star^2 = 0, \qquad z_t - A_\star z_{t-1} = (0, 0, \xi_t)^\top.$$

Its spectral radius is zero. For any fixed positive scales $d_i$, the first two standardized residuals remain zero. The third residual, $\xi_t/d_3$, is independent of history and symmetrically distributed. Although the residual distributions differ across variables, their Smooth-L1 conditional risks all attain their minima at zero. Thus, the same $f \equiv 0$ attains the minimum residual prediction risk achievable with access to the variable index, and a shared TCN can represent this zero function. In this construction, variable-specific linear coefficients express the opposing responses, leaving a common optimum for residual prediction.

# B Experimental Settings

## B.1 Data, Training, and Calibration

The synthetic systems follow the generation settings of the public benchmark (Han et al., 2025). Each generation seed provides 10 normal trajectories and 100 faulty trajectories. We split the normal trajectories into training, validation, and calibration sets in an 8/1/1 ratio; the normal counterparts of faulty trajectories do not enter fitting. SWaT uses a 10-second sampling interval and, in chronological order, 34, 6, and 7 normal segments for training, validation, and calibration, with one 1000-point segment separating adjacent groups. MSDS and WADI use chronological 70%/15%/15% splits, leave a 128-point gap after each of the two boundaries, and preserve the original segment boundaries. We exclude synthetic trajectories only when the pre-onset history or the final scoring window is too short; the 23 MSDS events require 14 normal points before fault onset.

We estimate the standardization parameters, VAR coefficients, and input scales $d_i$ from normal training data and keep them fixed throughout training, validation, calibration, and inference. We construct the TCN according to Eq. (9), use AdamW (Loshchilov & Hutter, 2019) to optimize the Smooth-L1 loss, and select the model by normal validation loss. Table 3 gives the settings. Each synthetic training window contains 17 history points and 64 target points. Windows do not cross trajectory boundaries, and the loss applies only to the target interval.

Table 3: Training and calibration settings for SLRCA.

| Setting | Four synthetic systems | SWaT | MSDS | WADI |
|---|---|---|---|---|
| TCN layers/width/kernel | 3/32/3 | 3/32/3 | 4/32/3 | 4/32/3 |
| Learning rate | 0.001 | 0.001 | 0.01 | 0.01 |
| Weight decay | $10^{-4}$ | $10^{-4}$ | $10^{-5}$ | $10^{-3}$ |
| Batch size | 32 | 64 | 32 | 64 |
| Maximum epochs | 500 | 2000 | 150 | 60 |
| Patience (epochs) | 30 | 20 | 10 | 5 |
| Calib. history/score length | 17/7 | 17/17 | 14/7 | 34/17 |
| VAR ridge coefficient | 0.05 | 0.05 | 0.05 | 1.0 |
| VAR residual scale $d_i$ | Std. dev. | Std. dev. | Std. dev. | RMS |
| Standardized-input clipping | $[-10, 10]$ | $[-10, 10]$ | None | $[-10, 10]$ |

The VAR spectral-radius limit is 0.90. MSDS fits VAR using all adjacent sample pairs within normal training segments; lagged terms do not cross segment boundaries. Write the normal training residual as $v_{i,t} = z_{i,t} - (Az_{t-1})_i$, with an overbar denoting an average over the samples used to estimate the scale. The population-standard-deviation and root-mean-square estimates of the input scale are

$$d_i^{\text{std}} = \sqrt{\overline{(v_i - \bar{v}_i)^2}}, \qquad d_i^{\text{RMS}} = \sqrt{\overline{v_i^2}}.$$

Table 3 specifies the estimator for each dataset. We replace standard deviations below $10^{-3}$ with 1 and floor root mean squares at $10^{-3}$. We scale the inputs as $u_{i,t} = v_{i,t}/d_i$ without additionally subtracting the residual mean.

After model selection, we estimate the sample standard deviation $s_i$ of the final residuals on separate normal calibration segments, with a floor of 0.10. We determine the pulse-score penalty $\lambda$ from 64 normal windows using Eq. (14).

## B.2 Comparison Methods and Event Evaluation

Each comparison method retains its own modeling objective and variable scores. We select models by normal validation loss and evaluate them on the same fault events and target windows. GDN, SARAD, TranAD, and MTAD-GAT use an 8/1/1 normal-data split for synthetic experiments. These four methods, MSCRED, and ALoRa-Loc use chronological 80%/20% training/validation splits for real-data experiments. We estimate their input standardization parameters using only the training split; MSCRED and ALoRa-Loc also restrict variable-pair selection to this split. AERCA retains the authors' normal-data splits: 8/2 for synthetic systems, 39/10 for SWaT, and 1/1 for MSDS, with validation data also used to estimate normal scales. Table 4 gives the method configurations.

Table 4: Main configurations and variable scores of the comparison methods.

| Method | Input and training | Variable score |
|---|---|---|
| AERCA | History order 5 for Nonlinear and 1 otherwise; four hidden layers for MSDS | Signed encoder scores calibrated by normal scales |
| GDN | 5-point history, 64-dimensional embeddings, top-5 adjacency; up to 30 epochs | Absolute prediction residuals, median/IQR calibration, and 4-point smoothing |
| SARAD | 100-point window, 512-dimensional representation, 3 layers, 8 heads; up to 3 epochs | Per-variable reconstruction error defined in the paper |
| TranAD | 10-point window, two-stage self-conditioning objective; up to 5 epochs; SWaT learning rate 0.0008 | Second-stage per-variable reconstruction error |
| MTAD-GAT | 100-point window, 300-dimensional GRU and latent variables; up to 100 epochs | Prediction error and reconstruction density, with weight $\gamma = 0.8$ |
| MSCRED | Relation-matrix scales: 6/11/17 for SWaT/WADI, 2/5/7 for MSDS; up to 20 epochs, 4,000/800 training/validation windows | Variable scores from reconstructed relation matrices |
| ALoRa-Loc | Up to 20 epochs and 20,000/2,000 training/validation windows | The original model's variable localization scores |
| PatchTST-RCA | Up to 400/800/600 updates for SWaT/MSDS/WADI | The same full score $S$ as our method |

The seven existing methods in the table take the maximum score per variable within the target window. Longer inputs extend before fault onset; insufficient history is padded with the first point. GDN estimates the median and IQR from normal data, using validation segments for real-data experiments and calibration trajectories for synthetic experiments.

As an architectural control, PatchTST-RCA retains the VAR frontend, normal-data split, and full score $S$ of SLRCA, with population standard deviations for its normal residual scales. We implement MTAD-GAT from the paper's equations and use Min–Max standardization fitted on the training split. We omit its WADI result because of abnormal reconstruction loss on normal validation data.

AC@$k$ divides the number of root causes among the top $k$ candidates by $\min(k, |\mathcal{G}_E|)$ and averages this ratio over events. Avg@10 averages AC@1 through AC@10, and MRR averages the reciprocal rank of the first root cause. We report means and sample standard deviations over three runs: synthetic experiments vary generation seeds in paired comparisons, while real-data experiments fix the events and vary training seeds.

# C Supplementary Experimental Results

## C.1 Architectural Ablations and VAR Order

The six architectures in Table 5 examine the effects of VAR linear prediction, parameter sharing, and variable embeddings on localization. Independent networks copy the initial parameters of the corresponding shared network, retain the same width, depth, convolutional kernels, and four-channel inputs, and then train separately. To examine the effect of explicitly distinguishing variables, we add four-dimensional learnable variable embeddings to the shared TCN that directly predicts raw observations. VAR-only scores the VAR residuals directly to test the role of TCN correction.

Within each system, all architectures use the same normal-data split, standardization, training target intervals, and model-selection rule. Each estimates its own residual scales and pulse-score penalty on the same normal windows. We report both the persistent-shift score $C$ and the full score $S$ to examine whether architectural comparisons depend on the scoring rule. The full-score results of VAR+shared match the main table.

Table 5: Architectural comparisons (mean and standard deviation over three seeds; real-data VAR uses one run). Neural counts exclude $D^2$ VAR coefficients; ID denotes a four-dimensional learnable variable-identity embedding.

| Dataset | Architecture | Neural params. | Persistent score $C$ | | Full score $S$ | |
|---|---|---|---|---|---|---|
| | | | AC@1 | MRR | AC@1 | MRR |
| Linear | VAR | 0 | $0.943 \pm 0.023$ | $0.968 \pm 0.013$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Shared | 6,657 | $0.900 \pm 0.030$ | $0.942 \pm 0.018$ | $0.993 \pm 0.012$ | $0.997 \pm 0.006$ |
| | Raw+Shared+ID | 7,057 | $0.880 \pm 0.061$ | $0.928 \pm 0.040$ | $0.993 \pm 0.012$ | $0.997 \pm 0.006$ |
| | Raw+Independent | 26,628 | $0.910 \pm 0.066$ | $0.947 \pm 0.039$ | $0.997 \pm 0.006$ | $0.998 \pm 0.003$ |
| | VAR+Independent | 26,628 | $0.940 \pm 0.017$ | $0.966 \pm 0.011$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Shared | 6,657 | $0.940 \pm 0.017$ | $0.965 \pm 0.010$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Nonlinear | VAR | 0 | $0.837 \pm 0.075$ | $0.898 \pm 0.043$ | $0.993 \pm 0.006$ | $0.996 \pm 0.004$ |
| | Raw+Shared | 6,657 | $0.863 \pm 0.006$ | $0.915 \pm 0.003$ | $0.997 \pm 0.006$ | $0.997 \pm 0.005$ |
| | Raw+Shared+ID | 7,065 | $0.723 \pm 0.047$ | $0.835 \pm 0.021$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Independent | 39,942 | $0.837 \pm 0.031$ | $0.896 \pm 0.015$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Independent | 39,942 | $0.833 \pm 0.042$ | $0.899 \pm 0.025$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Shared | 6,657 | $0.823 \pm 0.045$ | $0.887 \pm 0.025$ | $0.997 \pm 0.006$ | $0.998 \pm 0.004$ |
| Lotka–Volterra | VAR | 0 | $0.943 \pm 0.021$ | $0.968 \pm 0.010$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Shared | 6,657 | $0.973 \pm 0.031$ | $0.986 \pm 0.017$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Shared+ID | 7,201 | $0.923 \pm 0.115$ | $0.959 \pm 0.063$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Independent | 266,280 | $0.820 \pm 0.130$ | $0.891 \pm 0.095$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Independent | 266,280 | $0.842 \pm 0.143$ | $0.908 \pm 0.091$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Shared | 6,657 | $0.846 \pm 0.136$ | $0.900 \pm 0.091$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Lorenz96 | VAR | 0 | $0.397 \pm 0.021$ | $0.597 \pm 0.017$ | $0.780 \pm 0.082$ | $0.879 \pm 0.044$ |
| | Raw+Shared | 6,657 | $0.673 \pm 0.169$ | $0.785 \pm 0.126$ | $0.940 \pm 0.010$ | $0.970 \pm 0.005$ |
| | Raw+Shared+ID | 7,121 | $0.763 \pm 0.049$ | $0.848 \pm 0.035$ | $0.940 \pm 0.010$ | $0.969 \pm 0.005$ |
| | Raw+Independent | 133,140 | $0.733 \pm 0.071$ | $0.824 \pm 0.062$ | $0.897 \pm 0.025$ | $0.947 \pm 0.013$ |
| | VAR+Independent | 133,140 | $0.857 \pm 0.015$ | $0.918 \pm 0.010$ | $0.937 \pm 0.006$ | $0.967 \pm 0.003$ |
| | VAR+Shared | 6,657 | $0.830 \pm 0.154$ | $0.903 \pm 0.093$ | $0.887 \pm 0.046$ | $0.942 \pm 0.023$ |
| SWaT | VAR | 0 | 0.300 | 0.396 | 0.250 | 0.383 |
| | Raw+Shared | 6,657 | $0.333 \pm 0.076$ | $0.492 \pm 0.049$ | $0.367 \pm 0.153$ | $0.519 \pm 0.091$ |
| | Raw+Shared+ID | 7,245 | $0.400 \pm 0.132$ | $0.514 \pm 0.093$ | $0.500 \pm 0.050$ | $0.603 \pm 0.048$ |
| | Raw+Independent | 339,507 | $0.350 \pm 0.050$ | $0.506 \pm 0.027$ | $0.400 \pm 0.087$ | $0.562 \pm 0.054$ |
| | VAR+Independent | 339,507 | $0.200 \pm 0.050$ | $0.423 \pm 0.037$ | $0.417 \pm 0.058$ | $0.572 \pm 0.040$ |
| | VAR+Shared | 6,657 | $0.617 \pm 0.058$ | $0.699 \pm 0.057$ | $0.667 \pm 0.058$ | $0.739 \pm 0.062$ |
| MSDS | VAR | 0 | 0.000 | 0.413 | 0.000 | 0.413 |
| | Raw+Shared | 9,761 | $0.420 \pm 0.365$ | $0.628 \pm 0.309$ | $0.420 \pm 0.365$ | $0.628 \pm 0.309$ |
| | Raw+Shared+ID | 10,185 | $0.304 \pm 0.328$ | $0.565 \pm 0.246$ | $0.304 \pm 0.328$ | $0.565 \pm 0.246$ |
| | Raw+Independent | 97,610 | $0.000 \pm 0.000$ | $0.386 \pm 0.048$ | $0.000 \pm 0.000$ | $0.386 \pm 0.048$ |
| | VAR+Independent | 97,610 | $0.000 \pm 0.000$ | $0.380 \pm 0.010$ | $0.000 \pm 0.000$ | $0.380 \pm 0.010$ |
| | VAR+Shared | 9,761 | $0.652 \pm 0.000$ | $0.731 \pm 0.021$ | $0.652 \pm 0.000$ | $0.731 \pm 0.021$ |
| WADI | VAR | 0 | 0.364 | 0.408 | 0.364 | 0.408 |
| | Raw+Shared | 9,761 | $0.303 \pm 0.052$ | $0.391 \pm 0.041$ | $0.303 \pm 0.052$ | $0.391 \pm 0.041$ |
| | Raw+Shared+ID | 10,637 | $0.242 \pm 0.052$ | $0.320 \pm 0.029$ | $0.242 \pm 0.052$ | $0.320 \pm 0.029$ |
| | Raw+Independent | 1,200,603 | $0.333 \pm 0.052$ | $0.404 \pm 0.034$ | $0.333 \pm 0.052$ | $0.398 \pm 0.028$ |
| | VAR+Independent | 1,200,603 | $0.364 \pm 0.000$ | $0.397 \pm 0.005$ | $0.364 \pm 0.000$ | $0.396 \pm 0.005$ |
| | VAR+Shared | 9,761 | $0.364 \pm 0.000$ | $0.386 \pm 0.004$ | $0.364 \pm 0.000$ | $0.385 \pm 0.004$ |

We further compare VAR(1), VAR(2), and VAR(4), keeping the shared TCN, normal-data split, optimizer, model-selection rule, and scoring formulas fixed. For each order, we refit the frontend, retrain the network, and recalibrate the scores. VAR fitting and network inputs both use time points after an initial four-point history within each window; the companion matrix's spectral radius controls higher-order stability. In this comparison, MSDS uses a ridge coefficient of 1.0, root-mean-square scaling, and $[-10, 10]$ clipping. We also retrain the first-order model in Table 6 under these settings; its initial history and MSDS frontend differ from those in the main table. Increasing the VAR order does not consistently improve localization in this comparison.

Table 6: Matched comparison of VAR orders (mean and sample standard deviation over three seeds under the full score $S$). Total parameters include the shared TCN and all VAR coefficients.

| Dataset | Order $p$ | AC@1 | AC@3 | MRR | Total params. |
|---|---|---|---|---|---|
| SWaT | 1 | $0.700 \pm 0.050$ | $0.767 \pm 0.058$ | $0.763 \pm 0.046$ | 9,258 |
| SWaT | 2 | $0.600 \pm 0.050$ | $0.700 \pm 0.087$ | $0.682 \pm 0.040$ | 11,859 |
| SWaT | 4 | $0.683 \pm 0.029$ | $0.708 \pm 0.038$ | $0.743 \pm 0.022$ | 17,061 |
| MSDS | 1 | $0.507 \pm 0.153$ | $0.725 \pm 0.201$ | $0.653 \pm 0.114$ | 9,861 |
| MSDS | 2 | $0.420 \pm 0.365$ | $0.609 \pm 0.000$ | $0.587 \pm 0.183$ | 9,961 |
| MSDS | 4 | $0.145 \pm 0.251$ | $0.406 \pm 0.214$ | $0.369 \pm 0.219$ | 10,161 |
| WADI | 1 | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.391 \pm 0.009$ | 24,890 |
| WADI | 2 | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.386 \pm 0.005$ | 40,019 |
| WADI | 4 | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.386 \pm 0.004$ | 70,277 |

### C.2 Parameter Counts and Inference Cost

For four-channel inputs, hidden width $h$, kernel size $k$, and $L$ convolutional layers, the parameter count of a single temporal network is

$$M = (4hk + h) + (L - 1)(h^2k + h) + (h + 1). \tag{17}$$

The three terms correspond to the input convolution, hidden convolutions, and output layer, all including biases. Shared and independent networks use $M$ and $DM$ neural parameters, respectively; SLRCA additionally has $D^2$ VAR coefficients (Table 8). With fixed history length, the computational costs of VAR linear prediction, TCN prediction across all variables, and window scoring scale with $D^2$, $D$, and $wD$, respectively.

Table 7 compares the parameter counts of the neural networks used at inference, counting only AERCA's scoring encoder. Table 8 lists the VAR coefficients separately and gives the total parameter count of SLRCA, comprising the shared TCN parameters and VAR coefficients.

Table 7: Neural parameter counts of deep learning models, using the configurations in Table 2. AERCA counts only its inference encoder.

| **Method** | **SWaT** | **MSDS** | **WADI** |
|---|---|---|---|
| AERCA | 9,662,601 | 3,114,100 | — |
| GDN | 4,225 | 1,601 | 8,833 |
| SARAD | 9,588,746 | 9,525,442 | 9,699,914 |
| TranAD | 225,519 | 10,638 | 1,270,023 |
| MTAD-GAT | 3,965,111 | 1,356,240 | — |
| MSCRED | 3,206,243 | 3,206,243 | 3,206,243 |
| ALoRa-Loc | 3,183,563 | 22,754 | 3,220,703 |
| PatchTST-RCA | 17,473 | 17,473 | 17,473 |
| SLRCA | 6,657 | 9,761 | 9,761 |

Table 8: Parameter counts of shared and independent temporal networks. The last column sums shared neural parameters and VAR coefficients.

| **Dataset** | **D** | **Shared neural params.** | **Independent neural params.** | **VAR coefficients** | **SLRCA total params.** |
|---|---|---|---|---|---|
| SWaT | 51 | 6,657 | 339,507 | 2,601 | 9,258 |
| MSDS | 10 | 9,761 | 97,610 | 100 | 9,861 |
| WADI | 123 | 9,761 | 1,200,603 | 15,129 | 24,890 |

We measure inference cost on an NVIDIA RTX 5070 (12 GB), using PyTorch 2.7.0, CUDA 12.8, driver 595.79, and 8 CPU threads. Each model receives 5 warm-up rounds and 30 measured rounds, with CUDA synchronization before and after timing. Timing starts from raw events in memory and ends with variable scores on the CPU. It includes preprocessing, transfers, prediction, and scoring, and excludes file reading, model loading, offline calibration, and final ranking.

In Table 9, we compute p50 and p95 from each model's single-event latencies. We compute amortized latency per event by dividing the total time of a single call that sequentially processes all events by the number of events. Peak memory is the GPU memory allocated by PyTorch during single-event scoring, including resident models and scales.

Table 9: Event-scoring inference cost. Entries are means over three models; latency also includes sample standard deviations.

| Dataset | Score | p50 (ms) | p95 (ms) | Amortized per event (ms) | Peak VRAM (MiB) |
|---|---|---|---|---|---|
| MSDS | $C$ | $0.782 \pm 0.016$ | $1.469 \pm 0.023$ | $0.654 \pm 0.128$ | 0.134 |
| MSDS | $S$ | $0.858 \pm 0.022$ | $1.584 \pm 0.009$ | $0.628 \pm 0.070$ | 0.134 |
| WADI | $C$ | $0.832 \pm 0.016$ | $1.367 \pm 0.022$ | $0.646 \pm 0.005$ | 2.604 |
| WADI | $S$ | $0.913 \pm 0.013$ | $1.447 \pm 0.019$ | $0.661 \pm 0.004$ | 2.604 |

The two scores use the same models and residuals, with measurements alternating by event. Adding the pulse score requires only window aggregation, with no additional network forward pass.

### C.3 Role of the Score Components

To distinguish the contributions of scoring and prediction architecture, we hold the predictors, normal residual scales, and evaluation events fixed in Table 10. For both SLRCA and PatchTST-RCA, we compare the persistent-shift score $C$ with the full score $S$. Adding the pulse score raises SLRCA's AC@1 from 0.823 to 0.997 on Nonlinear and improves its AC@1 and MRR on SWaT by 5.0 and 4.0 percentage points, respectively. This comparison isolates the effect of scoring the same prediction residuals differently, complementing the architectural ablations with fixed scoring in Table 5.

Table 10: Scoring comparison with fixed predictors (mean and standard deviation over three seeds). $C$ is the persistent score and $S$ the full score.

| Dataset | Method | Score | AC@1 | AC@3 | MRR | Avg@10 |
|---|---|---|---|---|---|---|
| Linear | SLRCA | $C$ | $0.940 \pm 0.017$ | $0.974 \pm 0.023$ | $0.965 \pm 0.010$ | $0.987 \pm 0.003$ |
| Linear | SLRCA | $S$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Nonlinear | SLRCA | $C$ | $0.823 \pm 0.045$ | $0.781 \pm 0.022$ | $0.887 \pm 0.025$ | $0.913 \pm 0.014$ |
| Nonlinear | SLRCA | $S$ | $0.997 \pm 0.006$ | $0.982 \pm 0.014$ | $0.998 \pm 0.004$ | $0.994 \pm 0.004$ |
| Lotka–Volterra | SLRCA | $C$ | $0.846 \pm 0.136$ | $0.802 \pm 0.143$ | $0.900 \pm 0.091$ | $0.851 \pm 0.120$ |
| Lotka–Volterra | SLRCA | $S$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Lorenz96 | SLRCA | $C$ | $0.830 \pm 0.154$ | $0.789 \pm 0.131$ | $0.903 \pm 0.093$ | $0.797 \pm 0.096$ |
| Lorenz96 | SLRCA | $S$ | $0.887 \pm 0.046$ | $0.885 \pm 0.050$ | $0.942 \pm 0.023$ | $0.897 \pm 0.036$ |
| SWaT | SLRCA | $C$ | $0.617 \pm 0.058$ | $0.733 \pm 0.115$ | $0.699 \pm 0.057$ | $0.774 \pm 0.036$ |
| SWaT | SLRCA | $S$ | $0.667 \pm 0.058$ | $0.750 \pm 0.087$ | $0.739 \pm 0.062$ | $0.805 \pm 0.046$ |
| SWaT | PatchTST-RCA | $C$ | $0.550 \pm 0.000$ | $0.683 \pm 0.076$ | $0.632 \pm 0.008$ | $0.725 \pm 0.022$ |
| SWaT | PatchTST-RCA | $S$ | $0.550 \pm 0.000$ | $0.600 \pm 0.050$ | $0.622 \pm 0.008$ | $0.710 \pm 0.022$ |
| MSDS | SLRCA | $C$ | $0.652 \pm 0.000$ | $0.667 \pm 0.100$ | $0.731 \pm 0.021$ | $0.857 \pm 0.053$ |
| MSDS | SLRCA | $S$ | $0.652 \pm 0.000$ | $0.667 \pm 0.100$ | $0.731 \pm 0.021$ | $0.857 \pm 0.053$ |
| MSDS | PatchTST-RCA | $C$ | $0.217 \pm 0.000$ | $0.464 \pm 0.025$ | $0.419 \pm 0.006$ | $0.601 \pm 0.010$ |
| MSDS | PatchTST-RCA | $S$ | $0.217 \pm 0.000$ | $0.464 \pm 0.025$ | $0.419 \pm 0.006$ | $0.601 \pm 0.010$ |
| WADI | SLRCA | $C$ | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.386 \pm 0.004$ | $0.364 \pm 0.001$ |
| WADI | SLRCA | $S$ | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.385 \pm 0.004$ | $0.364 \pm 0.000$ |
| WADI | PatchTST-RCA | $C$ | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.396 \pm 0.007$ | $0.385 \pm 0.009$ |
| WADI | PatchTST-RCA | $S$ | $0.364 \pm 0.000$ | $0.364 \pm 0.000$ | $0.394 \pm 0.004$ | $0.383 \pm 0.007$ |

The benefit of adding the pulse score varies across systems and predictors. The performance of SLRCA remains largely unchanged on MSDS and WADI, while its gains on Lorenz96 differ across seeds. For PatchTST-RCA, the full score leaves AC@1 unchanged on all three systems and slightly lowers MRR on SWaT and WADI. These results show that the full score captures residual variations that the persistent-shift score does not fully reflect, while its benefit depends on the residuals produced by the predictor.

### C.4 Fixed-Model Evaluation on New Events

To test localization on new fault events, we keep each synthetic system's dynamics fixed and regenerate faulty trajectories, obtaining 1,194 valid events under the established window-length requirements. The four architectures use the same new events. Models, frontends, and calibration parameters remain fixed, with no further fitting or calibration.

Table 11: Localization on original and new events with fixed models (mean and sample standard deviation over three seeds under the full score $S$).

| | | AC@1 | | MRR | |
|---|---|---|---|---|---|
| Dataset | Architecture | Original | New | Original | New |
| Linear | Raw+Independent | $0.997 \pm 0.006$ | $0.997 \pm 0.006$ | $0.998 \pm 0.003$ | $0.998 \pm 0.003$ |
| | Raw+Shared | $0.993 \pm 0.012$ | $0.997 \pm 0.006$ | $0.997 \pm 0.006$ | $0.998 \pm 0.003$ |
| | VAR+Independent | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Shared | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Nonlinear | Raw+Independent | $1.000 \pm 0.000$ | $0.993 \pm 0.006$ | $1.000 \pm 0.000$ | $0.996 \pm 0.004$ |
| | Raw+Shared | $0.997 \pm 0.006$ | $0.990 \pm 0.010$ | $0.997 \pm 0.005$ | $0.994 \pm 0.007$ |
| | VAR+Independent | $1.000 \pm 0.000$ | $0.993 \pm 0.006$ | $1.000 \pm 0.000$ | $0.996 \pm 0.004$ |
| | VAR+Shared | $0.997 \pm 0.006$ | $0.997 \pm 0.006$ | $0.998 \pm 0.004$ | $0.998 \pm 0.004$ |
| Lotka–Volterra | Raw+Independent | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | Raw+Shared | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Independent | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| | VAR+Shared | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ | $1.000 \pm 0.000$ |
| Lorenz96 | Raw+Independent | $0.897 \pm 0.025$ | $0.920 \pm 0.017$ | $0.947 \pm 0.013$ | $0.958 \pm 0.011$ |
| | Raw+Shared | $0.940 \pm 0.010$ | $0.933 \pm 0.012$ | $0.970 \pm 0.005$ | $0.966 \pm 0.008$ |
| | VAR+Independent | $0.937 \pm 0.006$ | $0.920 \pm 0.046$ | $0.967 \pm 0.003$ | $0.958 \pm 0.025$ |
| | VAR+Shared | $0.887 \pm 0.046$ | $0.893 \pm 0.058$ | $0.942 \pm 0.023$ | $0.944 \pm 0.031$ |

Table 11 shows that SLRCA retains its original AC@1 on Linear, Nonlinear, and Lotka–Volterra; on Lorenz96, its AC@1 and MRR stay close to their original levels. The shared model that directly predicts raw observations remains better on Lorenz96. Overall, the fixed normal prediction model and scoring rule maintain localization performance close to the original level on new faulty trajectories, without adjustments for the new events.