TIPAD: MULTIVARIATE ANOMALY DETECTION VIA CRITICAL-TRANSITION THEORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection is essential for applications from industrial monitoring to financial risk management. Progress in anomaly detection has largely come from architectural innovation, while the nature of anomalies has received comparatively little attention. We revisit anomaly detection from a dynamical-systems perspective. A system generating normal data operates near a stable attractor, and detecting an anomaly is essentially detecting a crossing, the process by which the system departs from this stable state. This is precisely what critical-transition theory studies. Motivated by this, we propose **TipAD**, a multivariate anomaly detection method. TipAD combines a dual-path predictor that models fast and slow dynamics separately with a Kalman-filter-based criticality estimator that converts the prediction residual into a stress statistic. On the TSB-AD-M benchmark, which covers 17 datasets and 30 baseline methods, TipAD achieves the second-best average rank under the threshold-independent VUS-PR metric, demonstrating strong competitiveness and consistency across datasets. Beyond the method itself, we further find that critical-transition theory derives a variance equation whose form matches the covariance-prediction equation in Kalman estimation theory, grounding our design in theory. To our knowledge, this correspondence has not been recognized or used for anomaly detection.

1 INTRODUCTION

Anomaly detection plays a crucial role across various scientific fields, including aerospace, industrial control, healthcare, financial risk management, and cybersecurity. Observations of anomalous signals have existed since the 19th century (Edgeworth, 1887), as anomalous phenomena exist in all aspects of the physical world. Methods have evolved from early statistical methods to deep-learning reconstruction-based methods (e.g., autoencoders, VAEs) and prediction-based methods (e.g., LSTMs, Transformers), and more recently to foundation models. The field of anomaly detection has developed a large and diverse body of methods. However, we observe that this progress is largely driven by architectural replacement rather than by a deeper theoretical understanding of anomalies.

These architecture-driven methods have delivered substantial improvements across a wide range of practical applications. However, we find that this paradigm, focused largely on architectural innovation and leaderboard performance, has paid comparatively little attention to the nature of anomalies. Many models that perform well on particular datasets or metrics offer limited insight into why they succeed and often generalize poorly to other datasets or scenarios. This concern is now supported by a recent and rigorous benchmark study. TSB-AD (Liu & Paparrizos, 2024) is a widely recognized leaderboard covering 40 datasets and a broad range of detection algorithms. After a systematic re-evaluation of these methods, the authors found that simpler architectures generally outperformed complex variants, which calls into question the gains reported by many complex works. Based on this, we believe that simply stacking model capacity and architectural complexity cannot replace theoretical research into the nature of anomalies. We therefore revisit anomaly detection from a theoretical perspective, starting from what an anomaly fundamentally is.

Current methods (Ruff et al., 2021) define anomaly detection from a statistical perspective: they first characterize the statistical regularity of normal observations and flag observations in low-density regions. Essentially, this type of anomaly detection learns the characteristics of normal behavior and

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.

scores the degree of anomalousness. However, in the real world, data is generated by dynamical systems (Strogatz, 1994), including systems in various fields such as aerospace, industry, and finance. In these systems, the distribution of normal behavior is a quasi-stationary distribution that the system produces while operating stably near its attractor. Under this distribution, a low-density observation (an anomaly) can indicate that the system has left its stable state, which is closely related to what critical-transition theory studies (Kuehn, 2011). Therefore, we construct an anomaly detection framework from the perspective of critical transitions, detecting a system's departure from its normal (stable) state. We propose **TipAD**, a multivariate anomaly detection method motivated by critical-transition theory, and summarize our contributions as follows:

- We reformulate an anomaly as a *crossing*, the system's departure from its stable state. Inspired by this, we propose **TipAD**, a multivariate anomaly detection method. To our knowledge, TipAD is the first anomaly detection model designed to detect this shared crossing, motivated by the theory of critical transitions.
- We uncover a bridge between critical-transition theory and Kalman estimation theory. The variance equation that critical-transition theory derives for a system's linearized fluctuations around its quasi-equilibrium has the same (Lyapunov) form as the covariance-prediction step of the Kalman filter. This form is classical in estimation theory, but to our knowledge its correspondence to critical transitions has not been recognized or used in anomaly detection. Under linearized Gaussian dynamics, the Kalman filter is provably the minimum-variance estimator of this deviation.
- We evaluate TipAD on 17 datasets and compare it against 30 baseline models. Under the threshold-independent VUS-PR metric, TipAD achieves the second-best average ranking, demonstrating its strong competitiveness and consistency across datasets.

2 Related Work

2.1 Time-series Anomaly Detection

Anomaly detection in time series has received sustained attention for over six decades (Paparrizos et al., 2025), yielding a large and heterogeneous body of methods. These methods fall into three dominant categories (Liu & Paparrizos, 2024): statistical methods, neural network-based methods, and foundation model-based methods. A detailed review of representative methods is provided in Appendix A.

However, previous studies (Chandola et al., 2009; Ruff et al., 2021; Pang et al., 2022) have rarely explored anomaly detection from the perspective of critical-transition theory. Early warning systems (Williamson & Lenton, 2015) borrowed from bifurcation theory, fitting autoregressive models to track the Jacobian's leading eigenvalue and flag critical slowing down. However, critical slowing down is a bifurcation-specific precursor. Early warning systems built on it target the onset of a bifurcation rather than anomaly detection in general, and they have not been systematically evaluated on standard anomaly detection benchmarks. The recent work FOLD (Jhin & Park, 2026) uses the fold-bifurcation equation as a fixed output aggregator for anomaly detection. As a smaller branch of bifurcation theory, fold-bifurcation cannot cover all situations where the system deviates from its steady state. Furthermore, model design and stress signal learning remain engineering choices independent of theory. TipAD instead detects the crossing itself, the crossing that all tipping types share. Since detection no longer depends on a single precursor, TipAD is not limited to the crossing signals that early warning methods, built entirely around that one precursor, are able to detect.

2.2 Critical-transition Theory

In the science of complex systems, critical-transition theory is already well-established. This theory describes how the stability of a dynamical system changes as it approaches a tipping point. Early work (May, 1977) introduced this perspective into ecology, pointing out that ecosystems may have multiple stable states and undergo abrupt changes near a threshold. Subsequently, Scheffer (Scheffer et al., 2001) systematically integrated evidence from various ecosystems, demonstrating the widespread existence of such state transitions. In the subsequent monograph, *Critical Transitions in Nature and Society* (Scheffer, 2009), scientists integrated this into a general theory, the theory

of critical transitions. In the same year, Scheffer and other scientists, based on evidence in numerous fields such as ecosystems, epileptic seizures, and financial markets, proposed early-warning signals (Scheffer et al., 2009). As the scope of theoretical applications expands, researchers have gradually realized that not all critical transitions originate from the same triggering mechanism. Ashwin (Ashwin et al., 2012) systematically categorized the triggering mechanisms of critical transitions into three types: bifurcation-induced tipping (B-tipping), rate-induced tipping (R-tipping), and noise-induced tipping (N-tipping).

[B-tipping] Far from the bifurcation point, the system recovers rapidly after being subjected to small perturbations. However, as the operating point approaches the bifurcation point, the system's resilience gradually weakens, and the recovery time gradually lengthens. This phenomenon is called critical slowing down. Statistically, this critical slowing down phenomenon manifests as an increase in the variance and autocorrelation coefficient of the observed signal. This bifurcation-induced critical transition is also the theoretical basis of early warning signals (Williamson & Lenton, 2015) and the fold-bifurcation formula of FOLD (Jhin & Park, 2026).

[N-tipping] Unlike bifurcation-induced critical transitions, in noise-induced critical transitions, the system does not undergo any gradual approach process. A single, sufficiently large random noise perturbation directly pushes the system past the equilibrium point, causing a critical transition. Since the system exhibits no cumulative or changing trends during the process, the bifurcation-induced observational signals are inapplicable.

[R-tipping] Similarly, unlike B-tipping, R-tipping essentially occurs when the operating point itself moves faster than the system can keep up with it. Because the local stability of the operating point remains unchanged until it falls out of equilibrium, critical slowing down also cannot provide an early warning. In this respect, R-tipping behaves similarly to N-tipping. Currently, there is no closed-form theory like critical slowing down to transform them into statistical indicators that can be directly derived from observational data.

Recent research (Huang et al., 2024) has also confirmed the failure of the classic critical slowing down indicator for R-tipping. However, they also found that a model trained directly on the raw data could still extract sufficient high-order statistical structures for prediction, indicating that R-tipping is not unpredictable in principle, but rather that the underlying theory is still lacking.

Motivated by the above theoretical and empirical support, we observe that relying solely on unique precursors is insufficient to detect all critical transitions. To detect these critical transitions more comprehensively, we turn to their commonalities: regardless of the triggering factor, the system will eventually cross its stable state (Kuehn, 2011). Therefore, we design TipAD to detect this crossing itself (a characteristic shared by all three mechanisms), and evaluate it on the large-scale, multi-domain benchmark (Liu & Paparrizos, 2024), demonstrating the feasibility of a critical-transition-inspired approach to anomaly detection.

3 METHODOLOGY

3.1 PROBLEM DEFINITION

Following the TSB-AD benchmark (Liu & Paparrizos, 2024), we consider multivariate time-series anomaly detection. The input is a d-channel time series $X = (x_1, \ldots, x_n)$ with $x_t \in \mathbb{R}^d$ ($d > 1$). The objective is to produce an anomaly score $s = (s_1, \ldots, s_n) \in \mathbb{R}^n$, with higher s_t indicating a more anomalous step t.

3.2 ARCHITECTURE OVERVIEW

Figure 1 overviews our framework, **TipAD**, which consists of two main parts: the *predictor* and the *criticality estimator*. The predictor (a dual-path forecaster) splits each channel's observation into a slowly-varying quasi-equilibrium component and a fast deviation. It forecasts the two components separately, using a long-range attention head for the slow component and a short-range causal-convolution head for the fast one. The criticality estimator treats the predictor's residual as the observation of a linear-Gaussian state-space model. Theoretically, the system's deviation from the quasi-equilibrium manifold follows linearized Gaussian dynamics (Kuehn, 2011). We later show

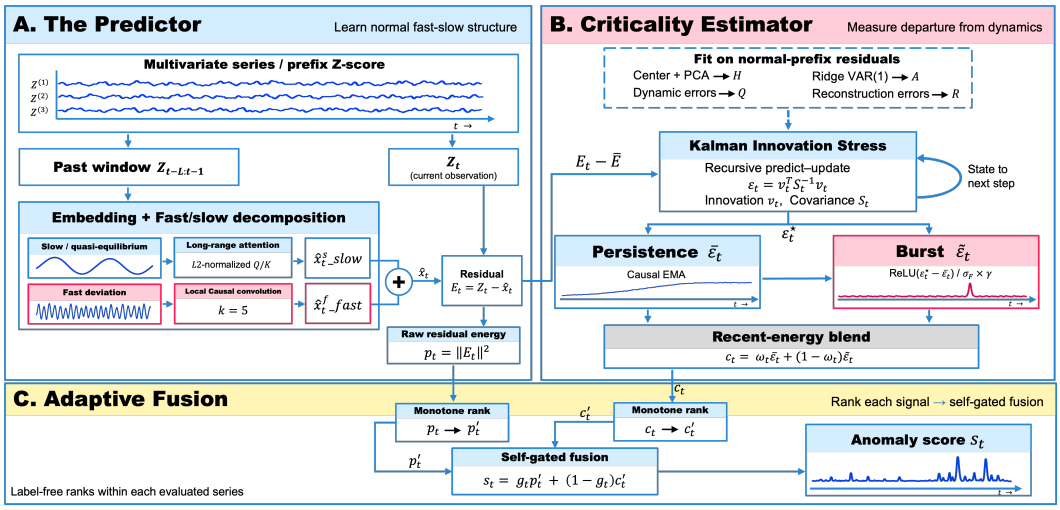


Figure 1: **Architecture of TipAD.**

that its variance equation has the same (Lyapunov) form as the Kalman filter's covariance-prediction step, which is why we use a Kalman filter to obtain a calibrated stress statistic. This statistic is split into a persistence and a burst component, weighted by their relative recent energies into a single score. This score is finally fused with the residual's raw energy to obtain the final anomaly score.

3.3 The Predictor

Each x_t is z-scored on the normal prefix $X_{1:\mathrm{tr}}$ into $Z_t \in \mathbb{R}^d$. Given the most recent L steps, $(Z_{t-L}, \ldots, Z_{t-1})$, the predictor forecasts Z_t one step ahead.

Fast/slow decomposition. Each of the L steps in this window is first embedded into a d_{model}-dimensional representation, $e_i = W_e Z_{t-L-1+i}$ ($W_e \in \mathbb{R}^{d_{\mathrm{model}} \times d}$, $i = 1, \ldots, L$), then split by a learnable causal exponential moving average (EMA):

$$\bar{e}_1 = e_1, \qquad \bar{e}_i = a\bar{e}_{i-1} + (1-a)e_i \ (i \geq 2), \qquad \tilde{e}_i = e_i - \bar{e}_i, \qquad a \in (0,1).$$

$\bar{e}_i$ estimates where the quasi-equilibrium currently sits, and $\tilde{e}_i$ is how far the current step deviates from it. Over the window these collect into slow_in $= (\bar{e}_1, \ldots, \bar{e}_L)$ and fast_in $= (\tilde{e}_1, \ldots, \tilde{e}_L)$, the inputs to the two heads below.

Dual-path predictor. To predict slow and fast components separately, each uses a mechanism suited to its own memory characteristics. The slow head is a long-range attention module with ℓ_2-normalized queries and keys, making its attention scores scale-invariant, and the fast head is a causal convolution with kernel size $k = 5$. Each takes the representation at the most recent step of its own path:

$$\hat{x}_t^{\,s} = W_s \, \mathrm{Attn}_{\ell_2}(\text{slow_in})_L, \qquad \hat{x}_t^{\,f} = W_f \, \mathrm{Conv}_k(\text{fast_in})_L, \qquad W_s, W_f \in \mathbb{R}^{d \times d_{\mathrm{model}}}.$$

Adding the two heads' outputs gives the prediction $\hat{x}_t = \hat{x}_t^{\,s} + \hat{x}_t^{\,f}$, and subtracting it from the observation gives the residual $E_t = Z_t - \hat{x}_t$. When both heads have correctly captured normal behavior at their own timescale, this residual is the deviation the criticality estimator processes next.

Why Fast-Slow Systems and Dual-path Prediction. The mathematical theory of fast-slow systems provides a natural definition of critical transitions (Kuehn, 2011). Based on the theoretical derivation, scientists define a critical transition as a deviation from the normal fast-slow structure. Therefore, clearly characterizing the normal structure is a theoretical prerequisite for the detection of critical transitions, so that a transition can be identified as a deviation from what was learned. We characterize this structure by decomposing the observed series into two parts, mirroring the theory's division into a slow manifold (the equilibrium the fast variable relaxes onto) and the fast variable's deviation from it. The quasi-equilibrium $\bar{e}_i$ is our analogue of the point on this manifold the series currently tracks, and the deviation $\tilde{e}_i$ is our analogue of this same deviation.

These two parts obey different dynamics, so we predict them with two separate paths. The quasi-equilibrium states move on a slow timescale, so their current value depends on the accumulated drift over the entire window. Attention mechanisms can associate the current step with any past steps, which is exactly what is needed to track quasi-equilibrium states. On the other hand, the autocorrelation of deviations decays on a fast timescale, and steps beyond the most recent short window contain virtually no information about their current value. Therefore, convolutions that pool only a small local neighborhood are exactly what the autocorrelation structure of deviations requires.

3.4 The Criticality Estimator

3.4.1 State-Space Identification

The residual $E_t = Z_t - \hat{x}_t$ is used to identify a state-space model on the normal prefix alone, $E_{L:\mathrm{tr}}$. We restrict identification to this segment because it falls within the scope of the critical transition theory (Kuehn, 2011). We remove its mean $\bar{E}$ and use PCA to project the centered residual onto an m-dimensional latent space chosen by an energy threshold,

$$\mu_t = H^\top (E_t - \bar{E}), \qquad H \in \mathbb{R}^{d \times m},$$

keeping $m \ll d$ principal directions and discarding the rest as redundant. μ_t is a compressed summary of the residual's shape at time t, and H is the observation matrix, the fixed map used to translate between this latent space and the original d channels. A ridge-regularized least-squares fit of a VAR(1) model, $\mu_t \approx A\mu_{t-1}$, gives the transition matrix A, capturing how the compressed residual typically moves from one step to the next. This fit is never exact, and the covariance of its residual becomes the process noise $Q = \mathrm{Cov}_t(\mu_t - A\mu_{t-1})$, the amount of that movement A cannot account for. Reconstructing the residual from μ_t back through H, $H\mu_t \approx E_t - \bar{E}$, is also imperfect, since $m < d$ means some detail is discarded during projection. The variance of this reconstruction error, taken separately for each channel, gives the observation noise,

$$R = \mathrm{diag}\big(\mathrm{Var}_t[(E_t - \bar{E}) - H\mu_t]\big).$$

The last latent state in the training window, μ_{tr}, and an initial covariance $P_{\mathrm{tr}}^- = Q$ initialize a recursion that carries this model, (A, H, Q, R), through the data that follows, one step at a time.

3.4.2 Kalman Filter

We use (A, H, Q, R) to run a recursive predict-update over the residual stream E_t, tracking a latent state $\mu_t \in \mathbb{R}^m$. At each step, we predict the state forward using the identified dynamics,

$$\mu_t^- = A\mu_{t-1}, \qquad P_t^- = AP_{t-1}A^\top + Q.$$

μ_t^-, the prior state, is our estimate for what the residual should look like before the new observation arrives. P_t^-, its covariance, tells us how much that estimate could be wrong. We then compare this estimate to what we actually observe,

$$\nu_t = (E_t - \bar{E}) - H\mu_t^-, \qquad S_t = HP_t^-H^\top + R.$$

ν_t, the innovation, measures how far the actual residual is from our estimate. S_t, its covariance, tells us how large this gap should normally be. It reflects both how uncertain our estimate was and how noisy the observation is.

Next, we compute the stress statistic and correct the state,

$$\varepsilon_t = \nu_t^\top S_t^{-1} \nu_t, \qquad K_t = P_t^- H^\top S_t^{-1}, \qquad \mu_t = \mu_t^- + K_t\nu_t, \qquad P_t = (I - K_tH)P_t^-.$$

ε_t is the stress statistic, or NIS (Normalized Innovation Squared). It normalizes ν_t by S_t, turning the departure into a single number that says whether it is normal or not, with S_t^{-1} collapsing all d channels into this one scalar. K_t, the Kalman gain, weighs the prior covariance P_t^- against the total covariance S_t, trusting the new observation more when the prior is less certain. μ_t, the updated state, moves μ_t^- toward the new observation by the amount set by K_t. P_t, the updated covariance, shrinks below P_t^- once the new observation resolves part of the earlier uncertainty (I is the $m \times m$ identity matrix). Only ε_t is the actual output. μ_t and P_t just carry the recursion to the next step.

Why a Kalman Filter. Our choice here follows from the theoretical structure itself. Near the slow (quasi-equilibrium) manifold, critical-transition theory (Kuehn, 2011) linearizes the deviation dynamics into a variance equation. This equation is scalar in its original form, and its multivariate extension is the standard Lyapunov covariance recursion (Särkkä & Svensson, 2023), exactly the form of the Kalman filter's covariance-prediction step $P_t^- = AP_{t-1}A^\top + Q$. In Appendix D we show that, under these linearized Gaussian dynamics, the Kalman filter is the minimum-variance estimator of the compressed latent state μ_t, which justifies our stress statistic $\varepsilon_t = \nu_t^\top S_t^{-1} \nu_t$.

3.4.3 Persistence/Burst Decomposition of the Stress Signal

The input to this stage is ε_t, the stress statistic, which squeezes departure across all d channels into one number, unable to tell a sustained rise in stress from a single burst. We calibrate ε_t to the training prefix, then take a causal exponential average,

$$\varepsilon_t^\star = \varepsilon_t / \mathrm{median}(\varepsilon_{1:\mathrm{tr}}), \qquad \bar{\varepsilon}_t = \mathrm{EMA}_\alpha(\varepsilon_t^\star).$$

$\varepsilon_t^\star$ takes a value near 1 when stress is typical of the normal period. $\bar{\varepsilon}_t$ is the persistence path. It tracks the recent, sustained level of stress. Whatever $\varepsilon_t^\star$ rises above this level is treated separately, as a burst,

$$\tilde{\varepsilon}_t = \gamma \cdot \frac{\mathrm{ReLU}(\varepsilon_t^\star - \bar{\varepsilon}_t)}{\sigma_F}, \qquad \sigma_F = \mathrm{std}\left(\mathrm{ReLU}(\varepsilon_\tau^\star - \bar{\varepsilon}_\tau)\right)_{\tau=1}^{\mathrm{tr}}.$$

$\tilde{\varepsilon}_t$ is the burst path. σ_F puts this excess on the same footing as $\bar{\varepsilon}_t$. γ then applies a separate, tunable weight to it, adjusting how much the burst path can influence the energy gate and the final score below. The two paths are combined by how much recent evidence each carries,

$$w_t = \frac{\mathrm{EMA}_\beta(\bar{\varepsilon}_t^2)}{\mathrm{EMA}_\beta(\bar{\varepsilon}_t^2) + \mathrm{EMA}_\beta(\tilde{\varepsilon}_t^2)}, \qquad c_t = w_t \bar{\varepsilon}_t + (1 - w_t)\tilde{\varepsilon}_t.$$

w_t is an energy ratio, not a learned gate. It is close to 1 when the persistence path carries more recent energy. Then c_t shifts toward $\bar{\varepsilon}_t$. It is close to 0 when the burst path dominates instead. Then c_t shifts toward $\tilde{\varepsilon}_t$. c_t is the output of this stage.

3.5 Adaptive Fusion

The input to this stage is two signals that describe departure from normal in different ways. $\rho_t = \|E_t\|^2$ is the raw energy of the residual E_t, computed without passing through the Kalman filter, unwhitened. c_t is the score produced by the Criticality Estimator, whitened by S_t and split by how sustained or sudden it is.

Different anomalies may leave their clearest signature in one of these two signals rather than the other. A single fixed weight would favor whichever pattern is more common. It would miss the other. We rank each within the evaluated series using a label-free, monotone rank transform,

$$\rho_t' = \mathrm{rank}(\rho_t), \qquad c_t' = \mathrm{rank}(c_t).$$

This puts both signals on the same $[0, 1]$ scale, regardless of their raw magnitudes. We then combine them with a self-gating softmax,

$$g_t = \frac{{\rho_t'}^\kappa}{{\rho_t'}^\kappa + {c_t'}^\kappa}, \qquad s_t = g_t \rho_t' + (1 - g_t) c_t'.$$

g_t is close to 1 when ρ_t' is the larger of the two. Then s_t shifts toward ρ_t'. g_t is close to 0 when c_t' is the larger instead. Then s_t shifts toward c_t'. κ controls how sharply g_t makes this choice: at κ=0 it is an even blend, and as κ grows it moves toward picking the larger signal outright. s_t is the final anomaly score.

4 Experiments

4.1 Datasets and Baselines

We evaluated our method on the widely recognized large-scale **benchmark TSB-AD** (Liu & Paparrizos, 2024). TSB-AD includes 40 different datasets with a total of 1070 rigorously selected time

Table 1: **VUS-PR score (Higher is better) per dataset on the TSB-AD-M benchmark.** Best results are shown in **bold red**, and second-best results in *yellow italic*. Our model is shown in Blue. **Avg. RANK (Lower is better)** is the average of the VUS-PR score rankings in each dataset.

Method	CATSv2	CreditCard	Daphnet	Exathlon	GECCO	GHL	Genesis	LTDB	MITDB	MSL	OPPORTUNITY	PSM	SMAP	SMD	SVDB	SWaT	TAO	Avg. RANK
xLSTMAD	0.23	0.07	0.28	*0.94*	0.19	0.02	*0.67*	0.42	0.14	0.33	*0.31*	**0.40**	0.28	*0.40*	0.34	0.52	0.74	**7.44**
TipAD (ours)	**0.36**	**0.12**	*0.39*	0.91	*0.38*	0.02	0.21	0.36	0.18	0.32	0.15	0.26	0.29	0.35	0.29	0.39	0.88	*7.59*
TipAD (Attention)	*0.31*	*0.10*	0.37	0.91	**0.40**	0.02	0.15	0.39	0.16	0.30	0.15	0.24	0.34	0.35	0.32	0.39	0.84	7.94
TipAD (CNN)	0.22	0.09	0.31	0.89	0.16	0.03	0.13	0.37	0.19	0.30	0.16	0.26	0.33	0.34	0.36	0.37	0.83	8.74
StreamVAE	0.12	0.06	0.38	0.92	0.04	0.02	0.42	0.44	0.16	**0.44**	0.15	0.20	**0.48**	*0.40*	0.48	0.39	0.77	9.03
CHARM	0.06	0.05	**0.48**	0.90	0.07	0.05	0.10	0.46	**0.49**	0.22	0.09	0.24	0.18	0.30	*0.55*	0.49	0.69	10.79
CNN	0.08	0.02	0.21	0.68	0.03	0.02	0.10	0.33	0.14	*0.35*	0.16	0.22	0.19	0.35	0.19	0.41	**1.00**	12.82
KMeansAD	0.12	0.02	0.30	0.37	0.06	0.03	**0.89**	0.41	0.06	**0.44**	0.06	0.21	*0.38*	0.36	0.20	0.16	0.86	13.21
Time-RCD	0.23	0.04	0.15	0.93	0.08	0.02	0.00	0.21	0.24	0.23	0.28	0.22	0.28	**0.46**	0.15	0.17	0.19	13.59
PCA	0.12	*0.10*	0.13	**0.95**	0.20	0.01	0.02	0.24	0.07	0.15	0.30	0.16	0.09	0.36	0.11	0.45	**1.00**	14.32
LSTMAD	0.04	0.02	0.31	0.82	0.02	*0.06*	0.04	0.30	0.09	0.22	0.17	0.24	0.16	0.33	0.15	0.16	*0.99*	15.56
TSPulse (FT)	0.07	0.00	0.36	0.90	0.18	0.01	0.02	*0.57*	0.13	0.20	0.06	0.14	0.31	0.35	0.48	0.13	0.93	15.76
AutoEncoder	0.06	0.03	0.13	0.91	0.05	0.05	0.01	0.21	0.04	0.22	0.14	*0.28*	0.13	0.30	0.06	**0.58**	**1.00**	15.85
Donut	0.07	0.02	0.17	0.66	0.03	0.05	0.18	0.26	0.12	0.30	0.15	0.20	0.18	0.19	0.11	0.44	0.75	16.09
MCD	0.13	0.06	0.14	0.80	0.03	0.01	0.06	0.21	0.04	0.23	0.17	0.26	0.10	0.26	0.07	*0.54*	**1.00**	16.24
OmniAnomaly	0.04	0.02	0.34	0.84	0.02	**0.07**	0.00	0.44	0.11	0.22	0.18	0.16	0.12	0.17	0.35	0.15	0.81	17.47
USAD	0.04	0.02	0.34	0.84	0.02	*0.06*	0.00	0.41	0.12	0.23	0.18	0.19	0.11	0.16	0.32	0.15	0.81	17.47
TSPulse (ZS)	0.05	0.00	0.35	0.89	0.18	0.01	0.01	0.36	0.07	0.20	0.06	0.14	0.30	0.35	0.39	0.13	0.93	17.79
TranAD	0.04	0.02	0.31	0.10	0.02	*0.06*	0.04	0.26	0.07	0.24	0.16	0.23	0.09	0.30	0.12	0.15	0.81	18.68
IForest	0.05	0.03	0.13	0.35	0.04	0.05	0.08	0.21	0.04	0.21	0.18	0.19	0.09	0.26	0.07	0.39	0.93	18.79
COPOD	0.05	0.05	0.11	0.40	0.04	0.03	0.08	0.21	0.04	0.21	0.17	0.20	0.10	0.19	0.07	0.31	*0.99*	19.06
HBOS	0.05	0.04	0.15	0.32	0.04	0.04	0.08	0.21	0.04	0.23	0.17	0.17	0.09	0.25	0.07	0.30	0.83	19.38
OCSVM	0.08	0.02	0.06	0.83	0.04	0.04	0.08	0.20	0.04	0.22	0.12	0.19	0.12	0.28	0.06	0.44	0.81	19.53
MMPAD	0.05	0.01	0.04	0.84	0.03	*0.06*	0.01	**0.60**	*0.37*	0.10	**0.36**	0.11	0.07	0.10	**0.63**	0.27	0.77	19.85
CBLOF	0.06	0.03	0.10	0.86	0.03	0.02	0.02	0.20	0.04	0.21	0.14	0.19	0.14	0.22	0.07	0.29	**1.00**	20.06
EIF	0.06	0.02	0.15	0.41	0.04	0.02	0.06	0.19	0.04	0.18	0.10	0.18	0.13	0.32	0.07	0.32	0.89	20.82
OFA	0.13	0.02	0.31	0.58	0.04	0.01	0.22	0.29	0.06	0.14	0.05	0.17	0.08	0.17	0.12	0.12	0.78	21.32
FITS	0.13	0.02	0.33	0.63	0.03	0.01	0.10	0.23	0.05	0.17	0.05	0.13	0.08	0.17	0.10	0.15	0.78	22.47
TimesNet	0.07	0.02	0.27	0.42	0.03	0.01	0.02	0.27	0.07	0.17	0.06	0.14	0.09	0.14	0.11	0.14	0.79	23.32
KNN	0.07	0.02	0.25	0.33	0.11	0.01	0.04	0.19	0.04	0.18	0.06	0.12	0.12	0.30	0.06	0.11	0.78	24.18
RobustPCA	0.04	0.02	0.06	0.77	0.02	0.03	0.00	0.23	0.04	0.22	0.13	0.12	0.07	0.10	0.08	0.12	**1.00**	24.53
LOF	0.05	0.02	0.11	0.16	0.13	0.01	0.08	0.19	0.04	0.14	0.10	0.15	0.09	0.16	0.06	0.15	0.79	25.26
AnomalyTransformer	0.03	0.02	0.07	0.10	0.02	0.03	0.01	0.21	0.05	0.12	0.07	0.21	0.06	0.07	0.08	0.18	0.77	26.03

series. Furthermore, its continuously updated leaderboard covers detection algorithms ranging from statistical and neural networks to state-of-the-art foundation models, with a unified testing environment ensuring fair and reproducible comparisons. We focused on its multivariate datasets, **TSB-AD-M**, which encompasses 17 real-world multivariate datasets and 200 time series (180 for evaluation and 20 for hyperparameter tuning), covering scenarios such as industrial sensors, spacecraft telemetry, water treatment plants, server clusters, healthcare, finance, and climate. Next, we compared our method with 30 **officially reproducible baseline methods on the TSB-AD-M leaderboard**, as shown in Table 1. For detailed information on datasets, baseline models, and hyperparameters, please refer to Appendix B.

4.2 Evaluation Metrics

To maintain consistency with TSB-AD, we adopt VUS-PR (Volume Under the Surface of the Precision-Recall curve) (Paparrizos et al., 2022) as our evaluation metric. VUS-PR is threshold-independent, effectively eliminating bias that may arise from specific threshold selection. TSB-AD also demonstrates the accuracy and fairness of VUS-PR through experiments with various evaluation metrics. In addition, the average VUS-PR value is susceptible to bias stemming from dataset characteristics. First, there are significant differences in score magnitudes across datasets: scores are mostly above 0.7 for TAO and Exathlon but below 0.1 for GHL, causing datasets with high scores to dominate the average. Second, as the number of series varies across datasets, a series-based average is further skewed by the few largest datasets. Therefore, we rank the VUS-PR scores within each dataset and calculate the average of these ranks (Avg. RANK) to ensure that each dataset carries equal weight. Additionally, we report the average VUS-PR values and win rate in Appendix C. These metrics provide supplementary information from different perspectives.

4.3 Overall Comparison

Table 1 compares TipAD with 30 reproducible baseline methods on the official TSB-AD leaderboard. All baseline models used official leaderboard results, while our model used the same tuning/evaluation split (averaging over 2 seeds). TipAD achieved the second-best average rank (7.59), second only to xLSTMAD (7.44), and outperformed all other baseline models, including the pre-trained time-series foundation models.

Table 2: **Generalization across datasets.** *Rank*: overall Avg. RANK. *Worst*: worst single-dataset rank. *Top-k*: datasets (of 17) ranked top-k.

Method	Rank	Worst	Top-2	Top-15
xLSTMAD	1	31	5	15
TipAD (ours)	**2**	**17**	4	**16**
StreamVAE	5	27	3	15
CHARM	6	32	3	14
TSPulse (FT)	12	32	1	10
MMPAD	24	33	5	6

We further analyzed the results for each dataset and found that TipAD's overall performance was more stable across all datasets. As shown in Table 2, although it only achieved a top-two VUS-PR ranking on 4 out of 17 datasets, it ranked in the top fifteen on 16 datasets. Its worst ranking on a single dataset was only 17th, much lower than other baselines, especially xLSTMAD, which had the highest average ranking but a worst-performing dataset ranking of 31st. This indicates that TipAD's overall second-best ranking is not due to outstanding performance on individual datasets, but rather because it maintained relatively balanced competitiveness across all datasets.

To analyze the predictor's contribution separately, Table 1 also lists two TipAD variants, differing only in the predictor. TipAD (Attention) replaces the causal convolutions in the dual-path predictor with attention modules. TipAD (CNN) replaces the entire dual-path predictor with ordinary convolutional modules, removing the fast-slow decomposition. Their final average rankings are 7.94 and 8.74, respectively. Even the weaker TipAD (CNN) outperformed all baselines except xLSTMAD. Since the criticality estimator remains unchanged in all three TipAD variants, this indicates that the criticality estimator plays a crucial role in the model. To further verify the contribution of the criticality estimator, we conducted a more comprehensive comparative experiment in subsequent ablation experiments.

5 Ablation and Sensitivity Studies

We further ablate the criticality estimator, first by removing it entirely, then by removing each of its components one at a time. Hyperparameter-sensitivity experiments are also provided in Appendix B.4.

5.1 Module Ablation

We ablate TipAD's two modules, the criticality estimator (CE) and the predictor, holding the rest of the pipeline fixed (Table 3). Removing the CE degrades overall performance for all four ablated models. The drop is largest for the variant with the weakest predictor, Linear+CE, whose Avg. RANK worsens from 8.76 to 12.12. Similarly, replacing the predictor with other options results in a decline in model performance. Notably, although Linear+CE uses the weakest predictor (a single-layer linear network), it still ranks just behind xLSTMAD and StreamVAE, outperforming CHARM and all other baseline models (excluding our own variants). Overall, both modules are essential to TipAD: removing either one markedly degrades its performance.

Beyond overall rank, the ablation reveals *why* the two modules are complementary rather than redundant. They specialize in different anomaly types. The predictor's residual is best suited to the *point-anomaly* regime. A one-step spike is already a large residual, so on point-anomaly datasets a raw residual is already a strong signal. However, additional mechanisms such as CE or smoothing predictors (e.g., CNNs) dilute this signal. Therefore, raw-residual variants (Linear) achieve the highest scores on the TAO (1.00) and GECCO (0.58) datasets. The CE handles collective anomalies, such as control faults (CATSv2) and ECG arrhythmias (SVDB), that persist over hundreds of time steps. While raw residuals tend to underestimate these slow, structural deviations, the CE compensates for this limitation, boosting TipAD (ours) performance on CATSv2 from 0.20 to 0.36, the highest score in that column. Thus, both modules are crucial, explaining why the self-gated fusion model remains competitive across the benchmark. Finally, strong performance on a single dataset

Table 3: **Module ablation (VUS-PR, higher is better).** Avg. RANK here is each method's average VUS-PR rank against the 30 official baselines and our ablation variants. Adding these variants changes the ranking pool, so a method's rank can differ between the two tables. The complete ranking is given in Appendix E.

Method	CATSv2	CreditCard	Daphnet	Exathlon	GECCO	GHL	Genesis	LTDB	MITDB	MSL	OPPORTUNITY	PSM	SMAP	SMD	SVDB	SWaT	TAO	Avg. RANK
xLSTMAD	0.23	0.07	0.28	**0.94**	0.19	0.02	**0.67**	0.42	0.14	*0.33*	**0.31**	**0.40**	0.28	*0.40*	0.34	**0.52**	0.74	**6.15**
TipAD (ours)	**0.36**	**0.12**	0.39	0.91	0.38	0.02	0.21	0.36	0.18	0.32	0.15	*0.26*	0.29	0.35	0.29	0.39	0.88	*6.82*
TipAD (Attention)	*0.31*	0.10	0.37	0.91	0.40	0.02	0.15	0.39	0.16	0.30	0.15	0.24	0.34	0.35	0.32	0.39	0.84	7.00
TipAD (CNN)	0.22	0.09	0.31	0.89	0.16	*0.03*	0.13	0.37	*0.19*	0.30	0.16	*0.26*	0.33	0.34	0.36	0.37	0.83	7.44
StreamVAE	0.12	0.06	0.38	*0.92*	0.04	0.02	*0.42*	*0.44*	0.16	**0.44**	0.15	0.20	**0.48**	*0.40*	*0.48*	0.39	0.77	7.65
TipAD (CNN) w/o CE	0.17	0.10	0.25	0.91	0.10	*0.03*	0.06	0.31	0.15	0.26	*0.17*	0.21	0.23	*0.40*	0.21	0.39	0.96	8.26
TipAD (Attention) w/o CE	0.15	*0.11*	0.32	0.91	**0.58**	0.01	0.05	0.34	0.12	0.27	0.16	0.20	0.28	**0.41**	0.21	0.41	0.98	8.59
TipAD (ours) w/o CE	0.20	**0.12**	0.33	0.91	0.50	0.01	0.04	0.29	0.14	0.25	0.16	0.21	0.27	*0.40*	0.18	0.40	*0.99*	8.68
Linear + CE	0.16	0.10	*0.41*	0.89	*0.56*	0.01	0.11	0.32	0.06	0.32	0.12	0.24	*0.37*	0.37	0.17	0.38	0.85	8.76
CHARM	0.06	0.05	**0.48**	0.90	0.07	**0.05**	0.10	**0.46**	**0.49**	0.22	0.09	0.24	0.18	0.30	**0.55**	*0.49*	0.69	8.94
Linear w/o CE	0.07	0.10	0.36	0.83	**0.58**	0.01	0.02	0.25	0.05	0.27	0.11	0.15	0.23	0.37	0.10	0.42	**1.00**	12.12

Table 4: Criticality-estimator component ablation. Sorted by Avg. RANK across datasets, best in **bold**. Columns: Wh, PCA whitening; Dyn, VAR(1)+Kalman; Slow, persistence (causal EMA); Fast, burst; Gate, energy gating; Fuse, fusion with residual energy.

Variant	Wh	Dyn	Slow	Fast	Gate	Fuse	Avg. RANK
TipAD (full)	✓	✓	✓	✓	✓	✓	**3.29**
w/o fusion (crit. only)	✓	✓	✓	✓	✓		3.35
w/o slow/fast (NIS)	✓	✓					3.59
residual energy (base)							3.82
slow path only	✓	✓	✓				3.82
static Mahalanobis	✓						4.29
fast path only	✓	✓		✓			5.82

does not imply strong performance across the whole benchmark. The Linear *w/o CE* variant tops TAO and GECCO yet has a poor overall Avg. RANK (12.12).

5.2 Criticality Estimator Ablation

We freeze the predictor to its full-model setting so that every variant consumes an identical residual E and differs only in criticality-estimator components. Table 4 ranks the variants by per-dataset Avg. RANK under the same protocol as our main comparison. The full TipAD model ranks best at 3.29. Removing only the fusion with residual energy raises the rank slightly to 3.35, because the residual energy it reinjects recovers the large-amplitude anomalies that the criticality score alone underweights. Normalized Innovation Squared (NIS) outranks residual energy (ranking 3.59 vs. 3.82), and the addition of gated slow and fast channels improves its ranking to 3.35. These two channels are most effective when used in combination. Using either channel (fast or slow) in isolation ranks below the NIS. In particular, the fast channel ranks lowest (5.82). Thus, removing any stage degrades the ranking, confirming the importance of the individual components within the estimator.

6 Conclusion

We propose **TipAD**, a multivariate anomaly detection method constructed from the perspective of critical transitions. TipAD consists of two modules. The predictor fits the signal's normal structure with a slow and fast head, leaving a residual. The criticality estimator then maps this residual into a stress statistic with a Kalman filter, refined by a persistence-burst decomposition. On the TSB-AD-M benchmark, which covers 17 multivariate datasets and 30 baselines, TipAD achieves the second-best average rank under the threshold-independent VUS-PR metric, demonstrating strong competitiveness and consistency across datasets. Beyond the method itself, we relate critical-transition theory to Kalman estimation theory, showing that the variance equation of critical-transition theory has the same form as the Kalman filter's covariance-prediction equation. We hope this finding offers a new perspective for future work in both fields.

AI use statement. We used generative AI tools as a writing assistant for polishing language and checking notation. No research ideas or content generation were conducted by LLMs.

Reproducibility Statement. To ensure the reproducibility and completeness of this paper, we make our code available at `https://anonymous.4open.science/r/TipAD-BD39/`.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2485–2494, 2021.

C. C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2nd edition, 2017.

Peter Ashwin, Sebastian Wieczorek, Renato Vitolo, and Peter Cox. Tipping points in open systems: Bifurcation, noise-induced and rate-dependent examples in the climate system. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370 (1962):1166–1184, 2012. doi: 10.1098/rsta.2011.0306.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3395–3404, 2020.

Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Tröster. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, 2010.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *ACM SIGMOD Record*, 29(2):93–104, 2000.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, 2009. doi: 10.1145/1541880.1541882.

Utsav Dutta, Sina Khoshfetrat Pakazad, and Henrik Ohlsson. Time to embed: Unlocking foundation models for time series with channel descriptions. *arXiv preprint arXiv:2505.14543*, 2025.

F. Y. Edgeworth. On discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375, 1887. doi: 10.1080/14786448708628471.

Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley M. Gifford, and Jayant Kalagnanam. TSPulse: Tiny pre-trained models with disentangled representations for rapid time-series analysis. In *International Conference on Learning Representations*, 2026.

Kamil Faber, Marcin Pietron, Dominik Zurek, and Roberto Corizzo. xLSTMAD: A powerful xLSTM-based method for anomaly detection. In *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2025.

Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model. *arXiv preprint arXiv:1612.06676*, 2016.

Patrick Fleith. Controlled anomalies time series (CATS) dataset, 2023. Version 2.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

Scott David Greenwald. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. PhD thesis, Harvard University–MIT Division of Health Sciences and Technology, 1990.

Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1479–1489, 2019.

Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

Yu Huang, Sebastian Bathiany, Peter Ashwin, and Niklas Boers. Deep learning for predicting rate-induced tipping. *Nature Machine Intelligence*, 6:1556–1565, 2024. doi: 10.1038/s42256-024-00937-0.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395, 2018.

Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. Exathlon: A benchmark for explainable anomaly detection over time series. *Proceedings of the VLDB Endowment*, 14(11):2613–2626, 2021.

Sheo Yon Jhin and Noseong Park. Point-wise anomaly detection via fold-bifurcation ODE. In *International Conference on Learning Representations*, 2026.

Christian Kuehn. A mathematical framework for critical transitions: Bifurcations, fast–slow systems and stochastic dynamics. *Physica D: Nonlinear Phenomena*, 240(12):1020–1035, 2011. doi: 10.1016/j.physd.2011.02.012.

Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy. In *International Conference on Machine Learning (ICML)*, 2026.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. COPOD: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1118–1123. IEEE, 2020.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 108231–108261. Curran Associates, Inc., 2024.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short-term memory networks for anomaly detection in time series. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, pp. 89–94, 2015.

Aditya P. Mathur and Nils Ole Tippenhauer. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pp. 31–36. IEEE, 2016.

Robert M. May. Thresholds and breakpoints in ecosystems with a multiplicity of stable states. *Nature*, 269:471–477, 1977. doi: 10.1038/269471a0.

Steffen Moritz, Frederik Rehbach, Sowmya Chandrasekaran, Margarita Rebolledo, and Thomas Bartz-Beielstein. GECCO industrial challenge 2018 dataset: A water quality dataset for the "Internet of Things: Online anomaly detection for drinking water quality" competition, 2018. Genetic and Evolutionary Computation Conference (GECCO) 2018.

M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005, 2018.

NOAA/PMEL. TAO dataset. `https://www.pmel.noaa.gov/gtmba/`, 2019. NOAA Pacific Marine Environmental Laboratory, Tropical Atmosphere Ocean project.

Kadir-Kaan Özer, René Ebeling, and Markus Enzweiler. STREAM-VAE: Dual-path routing for slow and fast dynamics in vehicle telemetry anomaly detection. In *2026 IEEE Intelligent Vehicles Symposium (IV)*, pp. 444–451. IEEE, 2026.

Randy Paffenroth, Kathleen Kay, and Les Servi. Robust PCA for anomaly detection in cyber networks. *arXiv preprint arXiv:1801.01571*, 2018.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):38:1–38:38, 2022. doi: 10.1145/3439950.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022.

John Paparrizos, Paul Boniol, Qinghua Liu, and Themis Palpanas. Advances in time-series anomaly detection: Algorithms, benchmarks, and evaluation measures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*, pp. 6151–6161, Toronto, ON, Canada, 2025. ACM. doi: 10.1145/3711896.3736565.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pp. 233–240. IEEE, 2010.

Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021. doi: 10.1109/JPROC.2021.3052449.

M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014a.

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pp. 4–11, 2014b.

Simo Särkkä and Lennart Svensson. *Bayesian Filtering and Smoothing*, volume 17 of *Institute of Mathematical Statistics Textbooks*. Cambridge University Press, Cambridge, UK, 2 edition, 2023.

Marten Scheffer. *Critical Transitions in Nature and Society*. Princeton Studies in Complexity. Princeton University Press, Princeton, NJ, 2009.

Marten Scheffer, Stephen Carpenter, Jonathan A. Foley, Carl Folke, and Brian Walker. Catastrophic shifts in ecosystems. *Nature*, 413:591–596, 2001. doi: 10.1038/35098000.

Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin, Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461:53–59, 2009. doi: 10.1038/nature08227.

Bernhard Scholkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12, 1999.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, 2018.

Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, Boulder, CO, 1994.

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, 2019.

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.

Alexander von Birgelen and Oliver Niggemann. Anomaly detection and localization for cyber-physical production systems with self-organizing maps. In *IMPROVE-Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency: Intelligent Methods for the Factory of the Future*, pp. 55–71. Springer, 2018.

Mark S. Williamson and Timothy M. Lenton. Detection of bifurcations in noisy coupled systems from multiple time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3): 036407, 2015. doi: 10.1063/1.4908603.

Haixu Wu, Tengge Hu, Yong Liu, Haoran Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, 2018.

J. Xu, H. Wu, J. Wang, and M. Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2021.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *The Twelfth International Conference on Learning Representations*, 2024.

T. Yairi, Y. Kato, and K. Hori. Fault detection by mining association rules from house-keeping data. In *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, volume 18, pp. 21. Citeseer, 2001.

Chin-Chia Michael Yeh, Audrey Der, Uday Singh Saini, Vivian Lai, Yan Zheng, Junpeng Wang, Xin Dai, Zhongfang Zhuang, Yujie Fan, Huiyuan Chen, et al. Matrix profile for anomaly detection on multidimensional time series. In *2024 IEEE International Conference on Data Mining (ICDM)*, pp. 911–916. IEEE, 2024.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43322–43355, 2023.

Appendix

Table of Contents

A Supplement to Related Work 15

B Experimental Environment 15

B.1 Datasets 15

B.2 Baselines 15

B.3 Benchmark Protocol: TSB-AD 16

B.4 Hyperparameters 17

B.5 Settings 18

C Additional Experiments 18

D Optimality of the Kalman Filter 18

E Full Module Ablation 20

A SUPPLEMENT TO RELATED WORK

We expand on the three categories of Section 2.1, describing representative methods.

[Statistical Method] This category utilizes statistical assumptions to detect anomalies that deviate from the expected data distribution. One family scores a point by its relationship to its nearest neighbors. LOF (Breunig et al., 2000) calculates the anomaly score by comparing local density with that of its neighbors. Another family instead summarizes normality as a set of clusters. KMeansAD (Yairi et al., 2001) calculates the anomaly score for each subsequence as the distance to the centroid of its assigned cluster. The third family models normality directly through a low-dimensional projection. PCA (Aggarwal, 2017) projects the data onto a lower-dimensional hyperplane, treating large deviation from this plane as an anomaly.

[Neural Network-based Method] This category models normal patterns by learning from training data without anomalies, and then identifies anomalies in new test data. These methods can be broadly categorized into three types: Reconstruction-based methods reconstruct the input from a compressed latent representation and score anomalies based on reconstruction error. AutoEncoder (Sakurada & Yairi, 2014a) projects the data into a low-dimensional latent space, reconstructs it through encoding and decoding, and indicates anomalies by larger reconstruction error. Prediction-based methods predict the next value based on history and score it based on prediction errors. CNN (Munir et al., 2018) uses a convolutional neural network (CNN) to predict the next time step and compare the predicted value with the observed value. Attention-based methods rely on self-attention mechanisms to associate each point with other points in the sequence. Anomaly-Transformer (Xu et al., 2021) introduces an "anomaly attention" mechanism that scores points based on the differences in their associations.

[Foundation Model-based Method] In recent years, foundation models have developed rapidly, demonstrating strong generalization capabilities. These works in the field of anomaly detection can be broadly divided into two categories. One category applies existing LLMs to time series anomaly detection. OFA (Zhou et al., 2023) adapts a frozen pre-trained GPT backbone to time series data, fine-tuning only its layer normalization and positional embedding layers. The other category utilizes foundation models pre-trained on large-scale time series data for anomaly detection applications. TSPulse (Ekambaram et al., 2026) is an ultra-compact time series foundation model pre-trained through a dual-space (time domain and frequency domain) masked reconstruction objective.

B EXPERIMENTAL ENVIRONMENT

B.1 DATASETS

We evaluate TipAD on **TSB-AD-M**, the multivariate dataset of the TSB-AD benchmark (Liu & Paparrizos, 2024). TSB-AD-M comprises 17 datasets and 200 curated multivariate time series in total, covering industrial and cyber-physical process monitoring (GHL, Genesis), water treatment and quality (SWaT, GECCO), spacecraft and rover telemetry (SMAP, MSL), server/cluster monitoring (Exathlon, SMD, PSM), healthcare signals (ECG: MITDB, SVDB, LTDB; Gait: Daphnet), finance (CreditCard), and ocean/climate monitoring (TAO), among others. Table 5 summarizes the size, dimensionality, and anomaly statistics of each family, reproduced from the official benchmark release.

B.2 BASELINES

We compare TipAD against all 30 baselines on the TSB-AD-M leaderboard (Liu & Paparrizos, 2024), covering classical, deep learning, and recent state-of-the-art methods (Table 6). Its scores are based on official TSB-AD evaluation results, which involved reproducing various baseline methods under a unified protocol. Consequently, TipAD is compared against **reproducible and independently verified** baselines under identical conditions, thereby ensuring a fair comparison.

Table 5: Summary characteristics of each dataset in TSB-AD (Liu & Paparrizos, 2024), # TS: the number of series, Avg. Dim: the mean number of channels, Avg. TS Len: the mean series length, Avg. Anomaly Len: the mean length of an anomaly segment, and Anomaly Ratio: the fraction of anomalous timesteps.

Name	# TS	Avg. Dim	Avg. TS Len	Avg. Anomaly Len	Anomaly Ratio
GHL (Filonov et al., 2016)	25	19	199,001.0	1,035.2	1.1%
Daphnet (Bachlin et al., 2010)	1	9	38,774.0	384.3	5.9%
Exathlon (Jacob et al., 2021)	27	21	60,878.4	1,373.3	9.8%
Genesis (von Birgelen & Niggemann, 2018)	1	18	16,220.0	16.7	0.3%
OPPORTUNITY (Roggen et al., 2010)	8	248	17,426.8	394.3	4.1%
SMD (Su et al., 2019)	22	38	25,466.4	112.8	3.8%
SWaT (Mathur & Tippenhauer, 2016)	2	59	207,457.5	1,093.6	12.7%
PSM (Abdulaal et al., 2021)	1	25	217,624.0	338.6	11.2%
SMAP (Hundman et al., 2018)	27	25	7,855.9	196.3	2.9%
MSL (Hundman et al., 2018)	16	55	3,119.4	111.7	5.1%
CreditCard (Sharafaldin et al., 2018)	1	29	284,807.0	1.1	0.2%
GECCO (Moritz et al., 2018)	1	9	138,521.0	33.8	1.2%
MITDB (Goldberger et al., 2000)	13	2	336,153.8	1,846.8	2.7%
SVDB (Greenwald, 1990)	31	2	207,122.6	268.2	4.8%
LTDB (Goldberger et al., 2000)	5	2	100,000.0	134.4	15.5%
CATSv2 (Fleith, 2023)	6	17	240,000.0	811.6	3.7%
TAO (NOAA/PMEL, 2019)	13	3	10,000.0	1.1	8.7%

Table 6: Baselines with their references, reused from TSB-AD. All are provided and evaluated by TSB-AD.

Family	Methods (with references)
Statistical	MCD (Rousseeuw & Driessen, 1999), OCSVM (Scholkopf et al., 1999), KNN (Ramaswamy et al., 2000), LOF (Breunig et al., 2000), KMeansAD (Yairi et al., 2001), CBLOF (He et al., 2003), IForest (Liu et al., 2008), HBOS (Goldstein & Dengel, 2012), PCA (Aggarwal, 2017), RobustPCA (Paffenroth et al., 2018), EIF (Hariri et al., 2019), COPOD (Li et al., 2020)
Neural network	AutoEncoder (Sakurada & Yairi, 2014b), LSTMAD (Malhotra et al., 2015), Donut (Xu et al., 2018), CNN (Munir et al., 2018), OmniAnomaly (Su et al., 2019), USAD (Audibert et al., 2020), AnomalyTransformer (Xu et al., 2021), TranAD (Tuli et al., 2022), TimesNet (Wu et al., 2023), FITS (Xu et al., 2024)
Foundation model	OFA (Zhou et al., 2023)
Recent	CHARM (Dutta et al., 2025), MMPAD (Yeh et al., 2024), StreamVAE (Özer et al., 2026), TSPulse (Ekambaram et al., 2026), Time-RCD (Lan et al., 2026), xLSTMAD (Faber et al., 2025)

B.3 Benchmark Protocol: TSB-AD

We follow the official TSB-AD evaluation protocol (Liu & Paparrizos, 2024). TSB-AD-M provides 200 curated multivariate series from 17 datasets, partitioned into a 20-series **Tuning** set and a disjoint 180-series **Evaluation** set. Every series additionally carries a short, anomaly-free training prefix.

- **Training**. All fitting is confined to this anomaly-free prefix: we compute the z-score normalization statistics on it, train the dual-path predictor on its windows, and identify the linear-Gaussian state-space model (A, H, Q, R) from its residuals. No anomaly labels are used. At test time the predictor and Kalman filter run over the full series, but nothing beyond the prefix is ever fit.
- **Tuning**. Only the score-fusion rule and its exponent are chosen here. All other hyperparameters are held fixed, with their robustness examined in the sensitivity analysis of Appendix B.4.
- **Evaluation**. All VUS-PR and Avg. RANK numbers in this paper are computed here, with nothing selected on it.

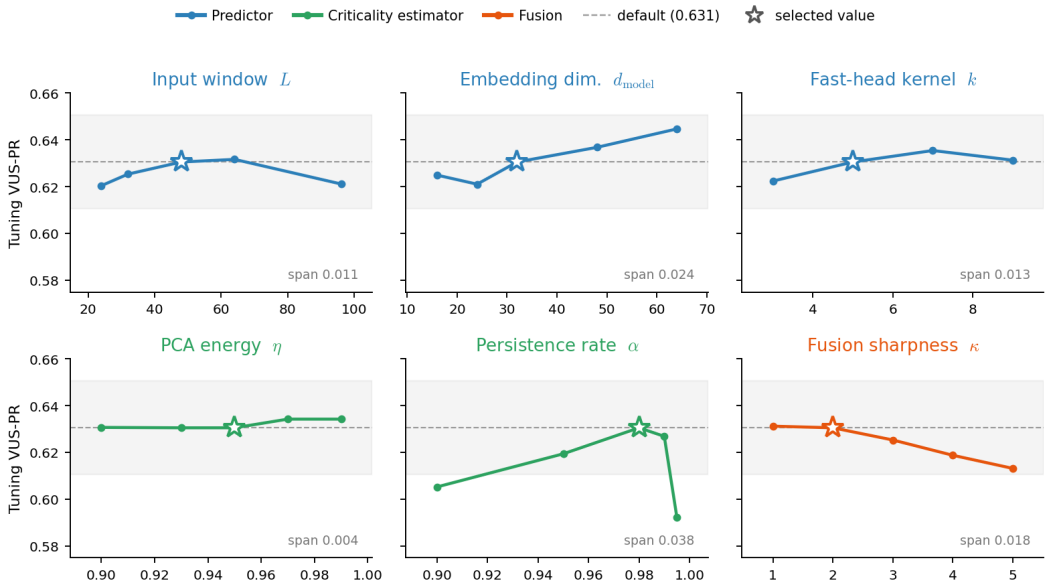


Figure 2: **Hyperparameter sensitivity of TipAD on the Tuning split**. The dashed line marks the default VUS-PR of 0.631, the shaded band spans ± 0.02, and the y-axis is shared across panels. Panels are grouped by module.

The partitioning method and metric (VUS-PR) recommended by TSB-AD were used in our experiments. Therefore, TipAD and each baseline were compared under the same leak-free protocol.

B.4 Hyperparameters

We list the relevant configuration below and then examine TipAD's robustness to its hyperparameters.

Hyperparameter Sensitivity.

- **Predictor:**
  - Window length $L \in \{24, 32, 48, 64, 96\}$
  - Embedding dimension $d_{\text{model}} \in \{16, 24, 32, 48, 64\}$
  - Fast-head kernel size $k \in \{3, 5, 7, 9\}$
- **Criticality Estimator:**
  - PCA energy threshold $\eta \in \{0.90, 0.93, 0.95, 0.97, 0.99\}$
  - Persistence EMA coefficient $\alpha \in \{0.90, 0.95, 0.98, 0.99, 0.995\}$
- **Adaptive Fusion:**
  - Sharpness $\kappa \in \{1.0, 2.0, 3.0, 4.0, 5.0\}$

We vary six of them one at a time around their chosen value, keeping the rest fixed, and measure VUS-PR on the Tuning split only, never touching Eval. They cover the predictor (input window L, model width d_{model}, fast-head kernel k), the criticality estimator (PCA energy threshold η, slow-channel persistence rate α), and the fusion (sharpness κ).

Determined hyperparameters. The same configuration was applied consistently across all datasets:

- **Predictor:**
  - Window length $L = 48$
  - Embedding dimension $d_{\text{model}} = 32$

  - Fast-head kernel size $k = 5$
- **Criticality Estimator:**
  - PCA energy threshold $\eta = 0.95$
  - Persistence EMA coefficient $\alpha = 0.98$
- **Adaptive Fusion:**
  - Sharpness $\kappa = 2.0$

TipAD demonstrates strong robustness (Fig. 2). Across the various parameter sweeps, no sweep spans more than 0.038 VUS-PR (default: 0.631), and four of the six vary by less than 0.02. The tuned sharpness κ and most fixed choices lie within 0.005 of their sweep optimum, and the persistence rate α is exactly optimal at 0.98. The only clear pattern is that a wider model scores slightly higher, with Tuning VUS-PR rising steadily as d_{model} grows (0.645 at 64 versus 0.631 at 32). We still keep the smaller d_{model}=32 for a lighter and faster model, giving up only 0.014 VUS-PR. In short, TipAD does not need careful hyperparameter tuning to work well.

B.5 Settings

All experiments were conducted on an Apple MacBook equipped with an Apple M3 chip (8-core CPU) and 24 GB of unified memory, running macOS 15.4, Python 3.13.2, and PyTorch 2.12.1. All TipAD components were executed on **the CPU, without GPU**.

C Additional Experiments

Table 7 provides further insight. Several methods with high Avg. VUS-PR scores exhibit relatively poor Avg. RANK standings. StreamVAE ranks first in Avg. VUS-PR (0.435) but only fifth in Avg. RANK (9.03). CHARM ranks third in Avg. VUS-PR (0.391) and sixth in Avg. RANK (10.79). This is primarily attributed to the imbalance in the number of series across datasets and the vast differences in score magnitudes among them. First, there is a significant disparity in the number of series per dataset (see Table 5): the largest datasets contribute far more series than others, while several others contain only one. Consequently, averages are heavily influenced by a few large datasets and fail to effectively reflect the benchmark's overall performance. Second, the score magnitudes vary drastically across datasets (e.g., TAO ≈ 0.9 versus GHL ≈ 0.02), which further causes a small number of datasets to dominate the raw average calculation. **The Avg. RANK metric eliminates these biases by ranking within each dataset and assigning equal weight to all datasets.** This is why we adopt it as our primary evaluation metric. Therefore, TipAD's lower raw average score is skewed by the aforementioned biases and does not accurately represent the model's overall performance. Avg. RANK provides a fair reflection of TipAD's overall competitiveness and its relatively stable ranking across domains.

Furthermore, the win rate against xLSTMAD offers another perspective on performance. Across the 17 dataset categories, TipAD won 7, tied 1, and lost the remaining 9. It trailed by a wide margin on the Genesis dataset (0.21 vs. 0.67). Consequently, the fact that TipAD and xLSTMAD achieved comparable average rankings (7.59 vs. 7.44) is not due to TipAD directly outperforming xLSTMAD in the majority of categories, but rather stems from its overall strong competitiveness against the full set of 30 baseline models.

D Optimality of the Kalman Filter

Under the linear-Gaussian model, the Kalman filter computes the exact posterior of the latent state. It is therefore the minimum-variance estimator, and its innovations are calibrated, giving ε_t a known null distribution. The minimum-variance estimator is the posterior mean (Lemma 1), and the Kalman filter computes it exactly (Theorem 1).

Setup. We use the linear-Gaussian model

$$\mu_t = A\mu_{t-1} + w_t, \qquad E_t = H\mu_t + v_t, \tag{1}$$

Table 7: **Per-dataset VUS-PR alongside Avg. RANK, Avg. VUS-PR, and win-rate against xLSTMAD.** Avg. VUS-PR is the mean over all 180 evaluation series. Win-rate is reported as Win–Tie–Loss over the 17 datasets. Sorted by Avg. RANK, matching Table 1.

Method	CATSv2	CreditCard	Daphnet	Exathlon	GECCO	GHL	Genesis	LTDB	MITDB	MSL	OPPORTUNITY	PSM	SMAP	SMD	SVDB	SWaT	TAO	Avg. RANK	Avg. VUS-PR	Win–Tie–Loss
xLSTMAD	0.23	0.07	0.28	*0.94*	0.19	0.02	*0.67*	0.42	0.14	0.33	*0.31*	**0.40**	0.28	*0.40*	0.34	0.52	0.74	7.44	*0.392*	–
TipAD (ours)	**0.36**	**0.12**	*0.39*	0.91	*0.38*	0.02	0.21	0.36	0.18	0.32	0.15	0.26	0.29	0.35	0.29	0.39	0.88	7.59	0.378	7–1–9
TipAD (Attention)	*0.31*	*0.10*	0.37	0.91	**0.40**	0.02	0.15	0.39	0.16	0.30	0.15	0.24	0.34	0.35	0.32	0.39	0.84	7.94	0.385	7–1–9
TipAD (CNN)	0.22	0.09	0.31	0.89	0.16	0.03	0.13	0.37	0.19	0.30	0.16	0.26	0.33	0.34	0.36	0.37	0.83	8.74	0.383	7–0–10
StreamVAE	0.12	0.06	0.38	0.92	0.04	0.02	0.42	0.44	0.16	**0.44**	0.15	0.20	**0.48**	*0.40*	0.48	0.39	0.77	9.03	**0.435**	7–2–8
CHARM	0.06	0.05	**0.48**	0.90	0.07	0.05	0.10	0.46	**0.49**	0.22	0.09	0.24	0.18	0.30	*0.55*	0.49	0.69	10.79	0.391	5–0–12
CNN	0.08	0.02	0.21	0.68	0.03	0.02	0.10	0.33	0.14	*0.35*	0.16	0.22	0.19	0.35	0.19	0.41	**1.00**	12.82	0.313	2–2–13
KMeansAD	0.12	0.02	0.30	0.37	0.06	0.03	**0.89**	0.41	0.06	**0.44**	0.06	0.21	*0.38*	0.36	0.20	0.16	0.86	13.21	0.295	6–0–11
Time-RCD	0.23	0.04	0.15	0.93	0.08	0.02	0.00	0.21	0.24	0.23	0.28	0.22	0.28	**0.46**	0.15	0.17	0.19	13.59	0.317	2–3–12
PCA	0.12	*0.10*	0.13	**0.95**	0.20	0.01	0.02	0.24	0.07	0.15	0.30	0.16	0.09	0.36	0.11	0.45	**1.00**	14.32	0.310	4–0–13
LSTMAD	0.04	0.02	0.31	0.82	0.02	*0.06*	0.04	0.30	0.09	0.22	0.17	0.24	0.16	0.33	0.15	0.16	*0.99*	15.56	0.307	3–0–14
TSPulse (FT)	0.07	0.00	0.36	0.90	0.18	0.01	0.02	*0.57*	0.13	0.20	0.06	0.14	0.31	0.35	0.48	0.13	0.93	15.76	0.387	5–0–12
AutoEncoder	0.06	0.03	0.13	0.91	0.05	0.05	0.01	0.21	0.04	0.22	0.14	*0.28*	0.13	0.30	0.06	**0.58**	**1.00**	15.85	0.295	3–0–14
Donut	0.07	0.02	0.17	0.66	0.03	0.05	0.18	0.26	0.12	0.30	0.15	0.20	0.18	0.19	0.11	0.44	0.75	16.09	0.260	2–0–15
MCD	0.13	0.06	0.14	0.80	0.03	0.01	0.06	0.21	0.04	0.23	0.17	0.26	0.10	0.26	0.07	*0.54*	**1.00**	16.24	0.271	2–0–15
OmniAnomaly	0.04	0.02	0.34	0.84	0.02	**0.07**	0.00	0.44	0.11	0.22	0.18	0.16	0.12	0.17	0.35	0.15	0.81	17.47	0.312	5–0–12
USAD	0.04	0.02	0.34	0.84	0.02	*0.06*	0.00	0.41	0.12	0.23	0.18	0.19	0.11	0.16	0.32	0.15	0.81	17.47	0.304	3–0–14
TSPulse (ZS)	0.05	0.00	0.35	0.89	0.18	0.01	0.01	0.36	0.07	0.20	0.06	0.14	0.30	0.35	0.39	0.13	0.93	17.79	0.361	4–0–13
TranAD	0.04	0.02	0.31	0.10	0.02	*0.06*	0.04	0.26	0.07	0.24	0.16	0.23	0.09	0.30	0.12	0.15	0.81	18.68	0.176	3–0–14
IForest	0.05	0.03	0.13	0.35	0.04	0.05	0.08	0.21	0.04	0.21	0.18	0.19	0.09	0.26	0.07	0.39	0.93	18.79	0.203	2–0–15
COPOD	0.05	0.05	0.11	0.40	0.04	0.03	0.08	0.21	0.04	0.21	0.17	0.20	0.10	0.19	0.07	0.31	*0.99*	19.06	0.203	2–0–15
HBOS	0.05	0.04	0.15	0.32	0.04	0.04	0.08	0.21	0.04	0.23	0.17	0.17	0.09	0.25	0.07	0.30	0.83	19.38	0.190	2–0–15
OCSVM	0.08	0.02	0.06	0.83	0.04	0.04	0.08	0.20	0.04	0.22	0.12	0.19	0.12	0.28	0.06	0.44	0.81	19.53	0.265	2–0–15
MMPAD	0.05	0.01	0.04	0.84	0.03	*0.06*	0.01	**0.60**	*0.37*	0.10	**0.36**	0.11	0.07	0.10	**0.63**	0.27	0.77	19.85	0.354	6–0–11
CBLOF	0.06	0.03	0.10	0.86	0.03	0.02	0.02	0.20	0.04	0.21	0.14	0.19	0.14	0.22	0.07	0.29	**1.00**	20.06	0.273	1–1–15
EIF	0.06	0.02	0.15	0.41	0.04	0.02	0.06	0.19	0.04	0.18	0.10	0.18	0.13	0.32	0.07	0.32	0.89	20.82	0.210	1–1–15
OFA	0.13	0.02	0.31	0.58	0.04	0.01	0.22	0.29	0.06	0.14	0.05	0.17	0.08	0.17	0.12	0.12	0.78	21.32	0.211	2–0–15
FITS	0.13	0.02	0.33	0.63	0.03	0.01	0.10	0.23	0.05	0.17	0.05	0.13	0.08	0.17	0.10	0.15	0.78	22.47	0.215	2–0–15
TimesNet	0.07	0.02	0.27	0.42	0.03	0.01	0.02	0.27	0.07	0.17	0.06	0.14	0.09	0.14	0.11	0.14	0.79	23.32	0.185	1–0–16
KNN	0.07	0.02	0.25	0.33	0.11	0.01	0.04	0.19	0.04	0.18	0.06	0.12	0.12	0.30	0.06	0.11	0.78	24.18	0.184	1–0–16
RobustPCA	0.04	0.02	0.06	0.77	0.02	0.03	0.00	0.23	0.04	0.22	0.13	0.12	0.07	0.10	0.08	0.12	**1.00**	24.53	0.239	2–0–15
LOF	0.05	0.02	0.11	0.16	0.13	0.01	0.08	0.19	0.04	0.14	0.10	0.15	0.09	0.16	0.06	0.15	0.79	25.26	0.139	1–0–16
AnomalyTransformer	0.03	0.02	0.07	0.10	0.02	0.03	0.01	0.21	0.05	0.12	0.07	0.21	0.06	0.07	0.08	0.18	0.77	26.03	0.118	2–0–15

with i.i.d. Gaussian noise $w_t \sim \mathcal{N}(0, Q)$, $v_t \sim \mathcal{N}(0, R)$, and a Gaussian initial state, all mutually independent. The matrices A, H, Q, and R are estimated from the normal data, and E_t denotes the mean-centered residual (the main text's $E_t - \bar{E}$), so that identification and filtering share the same centered signal. Running the Kalman filter then gives, at each step, a prior estimate $\hat{\mu}_t^-$ and covariance P_t^-, a posterior estimate $\hat{\mu}_t$ and covariance P_t, the innovation $\nu_t = E_t - H\hat{\mu}_t^-$, and its covariance S_t. These estimates are written without hats in the main text (as μ_t^- and μ_t). Here we add hats and reserve the unhatted μ_t for the true latent state that the theorem concerns.

Lemma 1 (The minimum-variance estimator is the posterior mean)**.** *Let μ satisfy $\mathbb{E}\|\mu\|^2 < \infty$. Among all estimators $\phi(E)$ of μ from the data E, the posterior mean $\mathbb{E}[\mu \mid E]$, the average of μ given the data, attains the smallest mean squared error:*

$$\mathbb{E}\|\mu - \phi(E)\|^2 \;\geq\; \mathbb{E}\big\|\mu - \mathbb{E}[\mu \mid E]\big\|^2,$$

with equality if and only if $\phi(E) = \mathbb{E}[\mu \mid E]$.

Proof. Let $g = \mathbb{E}[\mu \mid E]$ and split the error as $\mu - \phi = (\mu - g) + (g - \phi)$. The term $\mu - g$ has zero mean given E, and $g - \phi$ is a function of E, so the two are uncorrelated and the cross term vanishes. Hence

$$\mathbb{E}\|\mu - \phi\|^2 = \mathbb{E}\|\mu - g\|^2 + \mathbb{E}\|g - \phi\|^2 \;\geq\; \mathbb{E}\|\mu - g\|^2,$$

with equality exactly when $\phi = g = \mathbb{E}[\mu \mid E]$. □

By this lemma, proving the Kalman filter optimal reduces to showing that its estimate equals the posterior mean, which the next theorem does.

Theorem 1 (Kalman optimality)**.** *For the model above, the posterior of the state given the data so far is Gaussian, $p(\mu_t \mid E_{1:t}) = \mathcal{N}(\hat{\mu}_t, P_t)$, with mean and covariance given exactly by the Kalman recursion. Hence the Kalman estimate is the posterior mean, $\hat{\mu}_t = \mathbb{E}[\mu_t \mid E_{1:t}]$, and by Lemma 1 it is the minimum-variance estimator of the latent state μ_t, with error covariance P_t.*

Proof. We prove by induction that the posterior stays Gaussian with the Kalman mean and covariance. The base case holds because the state starts Gaussian. Now suppose $p(\mu_{t-1} \mid E_{1:t-1}) =$

$\mathcal{N}(\hat{\mu}_{t-1}, P_{t-1})$. The predict step preserves Gaussianity. Since $\mu_t = A\mu_{t-1} + w_t$ is a linear map of a Gaussian plus independent Gaussian noise, we have $p(\mu_t \mid E_{1:t-1}) = \mathcal{N}(\hat{\mu}_t^-, P_t^-)$. The update step preserves it as well. The pair (μ_t, E_t) is jointly Gaussian given $E_{1:t-1}$, and conditioning on the new observation E_t yields $p(\mu_t \mid E_{1:t}) = \mathcal{N}(\hat{\mu}_t, P_t)$, which is exactly the Kalman update. The mean parameter of this Gaussian posterior is $\mathbb{E}[\mu_t \mid E_{1:t}]$, so $\hat{\mu}_t = \mathbb{E}[\mu_t \mid E_{1:t}]$, and Lemma 1 gives the optimality. □

Remark. The same argument applies one step earlier. Since $\hat{\mu}_t^- = A\hat{\mu}_{t-1} = \mathbb{E}[\mu_t \mid E_{1:t-1}]$ is likewise a posterior mean (conditioned on one fewer observation), Lemma 1 shows it is the minimum-variance estimator of μ_t given only $E_{1:t-1}$, with error covariance P_t^-. This is the estimate the stress statistic ε_t is evaluated against, before the new observation E_t refines it into $\hat{\mu}_t$.

A calibrated stress statistic. Because $\hat{\mu}_t^-$ is the conditional mean, $E_t \mid E_{1:t-1} \sim \mathcal{N}(H\hat{\mu}_t^-, S_t)$, so the innovation satisfies $\nu_t \sim \mathcal{N}(0, S_t)$ and $\varepsilon_t = \nu_t^\top S_t^{-1} \nu_t \sim \chi^2_d$. This calibration thus gives ε_t a known null distribution, and departures from the fitted model inflate it beyond its χ^2_d expectation, which is the detection signal.

Scope. Under the linear-Gaussian assumptions with exact (A, H, Q, R), the estimator is globally minimum-variance. If only Gaussianity is relaxed, the Kalman filter remains the best linear unbiased estimator (BLUE); under nonlinearity, colored noise, or parameter misspecification, it is generally no longer BLUE.

E Full Module Ablation

Table 8: **Full module ablation on TSB-AD-M (VUS-PR, higher is better).** All 30 baselines are listed together with the ablation variants so that every Avg. RANK in Table 3 is directly verifiable. Our models and ablation variants are blue. **Avg. RANK (lower is better)**.

Method	CATSv2	CreditCard	Daphnet	Exathlon	GECCO	GHL	Genesis	LTDB	MITDB	MSL	OPPORTUNITY	PSM	SMAP	SMD	SVDB	SWaT	TAO	Avg. RANK
xLSTMAD	0.23	0.07	0.28	*0.94*	0.19	0.02	*0.67*	0.42	0.14	0.33	*0.31*	**0.40**	0.28	0.40	0.34	0.52	0.74	**6.15**
TipAD (ours)	**0.36**	**0.12**	0.39	0.91	0.38	0.02	0.21	0.36	0.18	0.32	0.15	0.26	0.29	0.35	0.29	0.39	0.88	*6.82*
TipAD (Attention)	*0.31*	0.10	0.37	0.91	0.40	0.02	0.15	0.39	0.16	0.30	0.15	0.24	0.34	0.35	0.32	0.39	0.84	7.00
TipAD (CNN)	0.22	0.09	0.31	0.89	0.16	0.03	0.13	0.37	0.19	0.30	0.16	0.26	0.33	0.34	0.36	0.37	0.83	7.44
StreamVAE	0.12	0.06	0.38	0.92	0.04	0.02	0.42	0.44	0.16	**0.44**	0.15	0.20	**0.48**	0.40	0.48	0.39	0.77	7.65
TipAD (CNN) w/o CE	0.17	0.10	0.25	0.91	0.10	0.03	0.06	0.31	0.15	0.26	0.17	0.21	0.23	0.40	0.21	0.39	0.96	8.26
TipAD (Attention) w/o CE	0.15	*0.11*	0.32	0.91	**0.58**	0.01	0.05	0.34	0.12	0.27	0.16	0.20	0.28	*0.41*	0.21	0.41	0.98	8.59
TipAD (ours) w/o CE	0.20	**0.12**	0.33	0.91	0.50	0.01	0.04	0.29	0.14	0.25	0.16	0.21	0.27	0.40	0.18	0.40	*0.99*	8.68
Linear + CE	0.16	0.10	*0.41*	0.89	*0.56*	0.01	0.11	0.32	0.06	0.32	0.12	0.24	0.37	0.37	0.17	0.38	0.85	8.76
CHARM	0.06	0.05	**0.48**	0.90	0.07	0.05	0.10	0.46	**0.49**	0.22	0.09	0.24	0.18	0.30	*0.55*	0.49	0.69	8.94
CNN	0.08	0.02	0.21	0.68	0.03	0.02	0.10	0.33	0.14	*0.35*	0.16	0.22	0.19	0.35	0.19	0.41	**1.00**	10.68
KMeansAD	0.12	0.02	0.30	0.37	0.06	0.03	**0.89**	0.41	0.06	**0.44**	0.06	0.21	*0.38*	0.36	0.20	0.16	0.86	11.35
Time-RCD	0.23	0.04	0.15	0.93	0.08	0.02	0.00	0.21	0.24	0.23	0.28	0.22	0.28	**0.46**	0.15	0.17	0.19	11.41
Linear w/o CE	0.07	0.10	0.36	0.83	**0.58**	0.01	0.02	0.25	0.05	0.27	0.11	0.15	0.23	0.37	0.10	0.42	**1.00**	12.12
PCA	0.12	0.10	0.13	**0.95**	0.20	0.01	0.02	0.24	0.07	0.15	0.30	0.16	0.09	0.36	0.11	0.45	**1.00**	12.35
LSTMAD	0.04	0.02	0.31	0.82	0.02	*0.06*	0.04	0.30	0.09	0.22	0.17	0.24	0.16	0.33	0.15	0.16	*0.99*	13.15
TSPulse (FT)	0.07	0.00	0.36	0.90	0.18	0.01	0.02	*0.57*	0.13	0.20	0.06	0.14	0.31	0.35	0.48	0.13	0.93	13.65
AutoEncoder	0.06	0.03	0.13	0.91	0.05	0.05	0.01	0.21	0.04	0.22	0.14	*0.28*	0.13	0.30	0.06	**0.58**	**1.00**	13.68
Donut	0.07	0.02	0.17	0.66	0.03	0.05	0.18	0.26	0.12	0.30	0.15	0.20	0.18	0.19	0.11	0.44	0.75	13.68
MCD	0.13	0.06	0.14	0.80	0.03	0.01	0.06	0.21	0.04	0.23	0.17	0.26	0.10	0.26	0.07	*0.54*	**1.00**	13.88
USAD	0.04	0.02	0.34	0.84	0.02	*0.06*	0.00	0.41	0.12	0.23	0.18	0.19	0.11	0.16	0.32	0.15	0.81	15.15
OmniAnomaly	0.04	0.02	0.34	0.84	0.02	**0.07**	0.00	0.44	0.11	0.22	0.18	0.16	0.12	0.17	0.35	0.15	0.81	15.18
TSPulse (ZS)	0.05	0.00	0.35	0.89	0.18	0.01	0.01	0.36	0.07	0.20	0.06	0.14	0.30	0.35	0.39	0.13	0.93	15.50
TranAD	0.04	0.02	0.31	0.10	0.02	*0.06*	0.04	0.26	0.07	0.24	0.16	0.23	0.09	0.30	0.12	0.15	0.81	16.03
IForest	0.05	0.03	0.13	0.35	0.04	0.05	0.08	0.21	0.04	0.21	0.18	0.19	0.09	0.26	0.07	0.39	0.93	16.44
COPOD	0.05	0.05	0.11	0.40	0.04	0.03	0.08	0.21	0.04	0.21	0.17	0.20	0.10	0.19	0.07	0.31	*0.99*	16.56
HBOS	0.05	0.04	0.15	0.32	0.04	0.04	0.08	0.21	0.04	0.23	0.17	0.17	0.09	0.25	0.07	0.30	0.83	16.76
OCSVM	0.08	0.02	0.06	0.83	0.04	0.04	0.08	0.20	0.04	0.22	0.12	0.19	0.12	0.28	0.06	0.44	0.81	16.88
CBLOF	0.06	0.03	0.10	0.86	0.03	0.02	0.02	0.20	0.04	0.21	0.14	0.19	0.14	0.22	0.07	0.29	**1.00**	17.29
MMPAD	0.05	0.01	0.04	0.84	0.03	*0.06*	0.01	**0.60**	*0.37*	0.10	**0.36**	0.11	0.07	0.10	**0.63**	0.27	0.77	17.74
EIF	0.06	0.02	0.15	0.41	0.04	0.02	0.06	0.19	0.04	0.18	0.10	0.18	0.13	0.32	0.07	0.32	0.89	18.06
OFA	0.13	0.02	0.31	0.58	0.04	0.01	0.22	0.29	0.06	0.14	0.05	0.17	0.08	0.17	0.12	0.12	0.78	18.53
FITS	0.13	0.02	0.33	0.63	0.03	0.01	0.10	0.23	0.05	0.17	0.05	0.13	0.08	0.17	0.10	0.15	0.78	19.53
TimesNet	0.07	0.02	0.27	0.42	0.03	0.01	0.02	0.27	0.07	0.17	0.06	0.14	0.09	0.14	0.11	0.14	0.79	20.32
KNN	0.07	0.02	0.25	0.33	0.11	0.01	0.04	0.19	0.04	0.18	0.06	0.12	0.12	0.30	0.06	0.11	0.78	21.18
RobustPCA	0.04	0.02	0.06	0.77	0.02	0.03	0.00	0.23	0.04	0.22	0.13	0.12	0.07	0.10	0.08	0.12	**1.00**	21.85
LOF	0.05	0.02	0.11	0.16	0.13	0.01	0.08	0.19	0.04	0.14	0.10	0.15	0.09	0.16	0.06	0.15	0.79	22.26
AnomalyTransformer	0.03	0.02	0.07	0.10	0.02	0.03	0.01	0.21	0.05	0.12	0.07	0.21	0.06	0.07	0.08	0.18	0.77	23.18

The module-ablation table in the main text (Table 3) is only a compact excerpt of the full comparison. To save space, it shows just our model, its ablation variants, and the three top-ranked baselines (xLSTMAD, StreamVAE, and CHARM) for reference. Table 8 below reports the same comparison

in full: all 30 official TSB-AD-M baselines together with every ablation variant, computed over the same pool as Table 3. Every Avg. RANK and VUS-PR value shown in Table 3 is therefore identical to its entry here. Table 8 shows the remaining baselines, so readers can directly verify each number in Table 3 against the full comparison.

At the same time, compared to Table 1, the comparison pool in the table above includes a large number of ablation variants, which causes the Avg. RANK results to change. This is because a method's ranking depends on the number of competitors involved in the comparison, rather than solely on its own score. Therefore, although the VUS-PR score for each baseline remains identical across both tables, the overall rankings shift due to the different comparison pool. However, this does not affect the experimental results or the demonstration of the model's performance.