RAFLOW: GRAPH-CONDITIONED CONTINUOUS NORMALIZING FLOWS FOR RELATIONAL TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time-series anomaly detection requires modelling both temporal dynamics and evolving dependencies among correlated entities. Existing graph-based normalizing flow methods incorporate learned relational structures into density estimation, but condition the graph only before the flow transformation, limiting the ability of the likelihood model to adapt continuously to dynamic relational structure. We propose RAFlow, a graph-conditioned continuous normalizing flow framework for relational time-series anomaly detection, where the velocity field of a neural ordinary differential equation (ODE) is jointly conditioned on learned temporal and relational embeddings at every integration step, ensuring inter-entity dependencies continuously shape the density transformation throughout the entire trajectory. Terminal likelihood alone cannot distinguish anomalies that reach similar density regions through irregular latent trajectories; we introduce a path-integral anomaly score that combines terminal log-likelihood with the kinetic energy of the ODE trajectory as complementary detection signals. Theoretically, we prove that RAFlow yields a consistent anomaly score that is stable, robust to measurement noise, and resistant to wild fluctuations under small input variations. Extensive experiments on five benchmark datasets against nineteen baselines show that RAFlow consistently outperforms existing graph-based flow methods and achieves the strongest performance on densely correlated benchmarks, highlighting the effectiveness of continuous graph-conditioned density estimation.

1 INTRODUCTION

Multivariate time-series anomaly detection (MTAD) plays a critical role in reliability monitoring for industrial, medical, and infrastructure systems, where sensors interact through complex and often unobserved dependencies. In these environments, anomalies rarely arise from isolated sensor deviations alone; instead, the most critical failures emerge when the joint behaviour of correlated entities departs from normal relational patterns. As modern sensor networks become increasingly high-dimensional and interconnected, effective anomaly detection requires modelling not only individual temporal dynamics but also evolving inter-entity dependencies.

Early MTAD approaches, such as reconstruction-based methods (Su et al., 2019; Zong et al., 2018), primarily model global temporal patterns while treating sensors independently. More recent graph-based methods (Deng & Hooi, 2021; Dai & Chen, 2022; Zhou et al., 2023) incorporate graph neural networks (GNNs) to capture inter-sensor dependencies and achieve substantially improved performance. However, these methods rely on finite-depth likelihood transformations, where relational structure acts primarily as static conditioning before density evolution. As a result, relational dependencies influence latent representations only indirectly, while the transformation dynamics themselves remain decoupled from graph interactions throughout the likelihood estimation process. This limits the ability of the model to adapt continuously to evolving relational dynamics, particularly for correlated anomalies that emerge through abnormal joint behaviour despite individually normal sensor patterns.

To overcome these limitations, relational dependencies must participate directly in the density evolution process itself rather than acting only as static conditioning. Motivated by this perspective, we propose **Relational Adaptive Flow (RAFlow)**, a graph-conditioned continuous flow framework

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.

for multivariate time-series anomaly detection. RAFlow models density evolution as continuous-time dynamics governed by a neural ODE, where the velocity field is jointly parameterised by temporal representations and dynamically learned graph-relational embeddings at every integration step. Unlike finite-depth likelihood transformations, this continuous-time formulation enables evolving inter-entity dependencies to continuously shape latent density dynamics throughout the full transformation trajectory. Furthermore, unlike existing CNF-based sequential models (Rubanova et al., 2019) and graph neural ODE methods (Poli et al., 2019; Chamberlain et al., 2021), which primarily focus on latent sequence modelling or feature propagation objectives, RAFlow directly enables likelihood-based anomaly detection through continuous relational density estimation. We further introduce a trajectory-aware anomaly score combining terminal negative log-likelihood with ODE trajectory kinetic energy, capturing both low-density deviations and irregular latent transport behaviour.

Contributions. The novelties and main contributions of this work are as follows:

1. We propose RAFlow, a graph-conditioned continuous flow framework for multivariate time-series anomaly detection. RAFlow formulates anomaly detection as a continuous relational density evolution process, enabling evolving inter-entity dependencies to continuously shape latent density dynamics throughout the full transformation trajectory.
2. We introduce a trajectory-aware anomaly scoring framework that combines terminal likelihood estimation with latent trajectory complexity, improving sensitivity to correlated anomalies that may exhibit similar terminal densities but irregular transformation dynamics.
3. We present a theoretical proof showing that RAFlow produces a consistent anomaly score—one that remains stable and robust in the presence of measurement noise, ensuring it does not fluctuate under minor input perturbations.
4. Extensive experiments on five benchmark datasets against nineteen baseline methods demonstrate that RAFlow consistently outperforms existing graph-based anomaly detection approaches, with particularly strong improvements on benchmarks exhibiting dense relational coupling.

2 Related Work

Reconstruction and prediction-based MTAD. Early MTAD approaches primarily rely on reconstruction or prediction objectives to model temporal dynamics. Representative methods such as LSTM-VAE (Park et al., 2018), OmniAnomaly (Su et al., 2019), and DAGMM (Zong et al., 2018) learn latent temporal representations for anomaly detection, but largely treat sensors independently and therefore struggle to capture relational anomalies arising from inter-sensor dependencies. Transformer-based approaches, including the Anomaly Transformer (Xu et al., 2022) and DCdetector (Yang et al., 2023), further improve long-range temporal modelling through self-attention mechanisms. More recent methods such as SARAD (Dai et al., 2024) and CATCH (Wu et al., 2025) achieve strong empirical performance using spatial association modelling and frequency-domain patching strategies respectively.

Graph-based multivariate anomaly detection. To explicitly model inter-sensor dependencies, graph-based MTAD methods incorporate graph neural networks into anomaly detection frameworks. GDN (Deng & Hooi, 2021) employs graph attention mechanisms for prediction-based scoring, while GANF (Dai & Chen, 2022) and MTGFlow (Zhou et al., 2023) introduce graph-conditioned likelihood estimation through flow-based density modelling. Despite strong empirical performance, existing graph-based likelihood models primarily treat relational structure as conditioning prior to finite-depth density transformations, limiting the ability of the density evolution process itself to adapt continuously to evolving relational interactions.

Flow-based density estimation and continuous-time models. Normalizing flows (Kobyzev et al., 2020; Papamakarios et al., 2021) enable exact likelihood estimation through invertible transformations, with architectures such as NICE (Dinh et al., 2014), RealNVP (Dinh et al., 2017), MAF (Papamakarios et al., 2017), and Glow (Kingma & Dhariwal, 2018) forming the foundation of likelihood-based anomaly detection. Continuous normalizing flows and Neural ODEs (Chen et al., 2018; Grathwohl et al., 2019) extend this framework by parameterising density evolution as continuous-time dynamics governed by differential equations, enabling adaptive latent transformation trajectories. Latent ODE

models (Rubanova et al., 2019) capture temporal sequence dynamics through continuous latent-state evolution, while Graph Neural ODEs (Poli et al., 2019) and GRAND (Chamberlain et al., 2021) continuously propagate node representations over graph structures for representation learning and prediction tasks. More recently, score-based diffusion models have also been explored for time-series generation and anomaly detection through stochastic denoising dynamics, although they typically rely on iterative sampling procedures rather than exact likelihood estimation. Existing continuous-time approaches, however, do not explicitly couple adaptive relational structure with continuous likelihood evolution for graph-based anomaly detection.

3 Proposed Method

Let $\mathbb{X} = \{X_1, \ldots, X_K\}$ denote a multivariate time series with K entities, where $X_k = (x_{k,1}, \ldots, x_{k,L})$ represents the observations of entity k over L time steps. We segment the series into overlapping windows of length T and stride s, giving the c-th window $X^c = [X_1^c, \ldots, X_K^c] \in \mathbb{R}^{K \times T}$, where $X_k^c \in \mathbb{R}^T$ is the subsequence of entity k. Following the low-density assumption, we define the anomaly score as the negative log-likelihood $\phi(X^c) = -\log p(X^c)$, so that windows in low-probability regions receive higher scores. Since entities in multivariate time series are mutually dependent, we learn a context-conditioned invertible transformation $Z_k^c = f_\theta(X_k^c; C_k^c)$ that maps each X_k^c to a latent variable Z_k^c, where the context C_k^c is a deterministic function of X^c defined in Section 3.1. By the change-of-variables formula applied per-entity with C_k^c held fixed, the conditional log-likelihood for entity k is:

$$\log p_\theta(X_k^c \mid C_k^c) = \log p_Z(Z_k^c) + \log \left| \det \frac{\partial Z_k^c}{\partial X_k^c} \right|_{C_k^c \text{ fixed}}, \tag{1}$$

where $p_Z(Z_k^c)$ is an entity-specific Gaussian prior defined in Section 3.3, and the Jacobian is computed holding the realised context C_k^c constant. The window-level training objective is the factorized conditional log-likelihood given in Eq. 2.

$$\widetilde{\mathcal{L}}(\theta) = -\frac{1}{NK} \sum_{c=1}^{N} \sum_{k=1}^{K} \log p_\theta(X_k^c \mid C_k^c(X^c)). \tag{2}$$

Because C_k^c is a deterministic function of the full window X^c, the product $\prod_k p_\theta(X_k^c \mid C_k^c(X^c))$ does not integrate to 1 over X^c. Consequently, $\widetilde{\mathcal{L}}$ is a composite conditional scoring function optimized via empirical risk minimization, not the log-likelihood of a fully coupled joint density. The gradient $\partial \widetilde{\mathcal{L}} / \partial \theta$ nevertheless flows through C_k^c to train the temporal encoder, graph learner, and flow parameters jointly. Figure 1 illustrates the overall view of the RAFlow framework.

3.1 Temporal-relational context construction

For each entity k in window $\mathbf{X}^c$, we compute a context vector $C_k^c \in \mathbb{R}^d$ that captures both temporal dynamics and inter-entity dependencies to condition the subsequent density transformation. We encode the temporal behaviour of each entity using a shared LSTM: $h_{k,\tau}^c = \mathrm{LSTM}(x_{k,\tau}^c, h_{k,\tau-1}^c)$, for $\tau = 1, \ldots, T$, retaining the final hidden state $h_k^c = h_{k,T}^c \in \mathbb{R}^d$ as a fixed-length temporal summary. Stacking all entities gives $H^c = [h_1^c, \ldots, h_K^c]^\top \in \mathbb{R}^{K \times d}$. To capture inter-entity dependencies, we learn a window-specific adjacency matrix via scaled dot-product attention. Query and key projections $Q_i^c = h_i^c W_Q$ and $K_j^c = h_j^c W_K$, with $W_Q, W_K \in \mathbb{R}^{d \times d_a}$, give pairwise interaction scores $e_{ij}^c = \frac{(Q_i^c)^\top K_j^c}{\sqrt{d_a}}$, normalised row-wise via softmax to obtain the dynamic adjacency matrix $A^c = [a_{ij}^c] \in \mathbb{R}^{K \times K}$, recomputed for each window. Temporal and relational information are fused as

$$C^c = \mathrm{ReLU}(A^c H^c W_1 + H^c W_2)\, W_3, \tag{3}$$

where $W_1, W_2, W_3 \in \mathbb{R}^{d \times d}$ are learnable parameters and the ReLU is applied element-wise. The k-th row of C^c gives the context vector $C_k^c \in \mathbb{R}^d$ for entity k.

3.2 Context-conditioned continuous normalising flow

We model the density transformation as a continuous-time process governed by a neural ODE (Chen et al., 2018), following the CNF framework (Grathwohl et al., 2019). The key novelty of RAFlow is

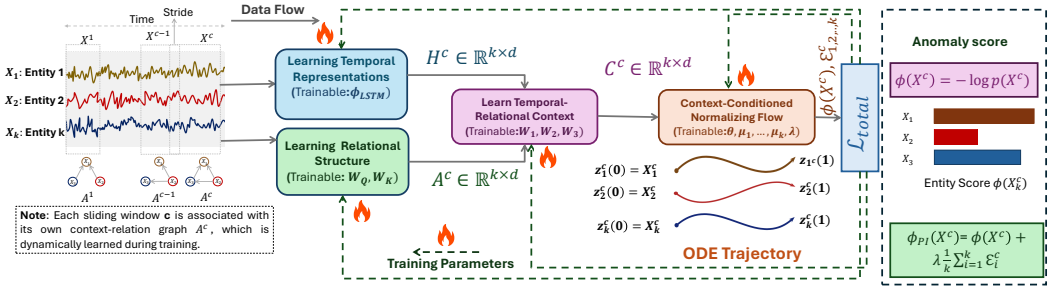


Figure 1: Overview of RAFlow. Temporal representations H^c and relational graph A^c are fused into context C^c that conditions the ODE velocity field at every integration step, ensuring graph topology continuously shapes the density transformation throughout the full trajectory.

that the ODE velocity field is jointly conditioned on both temporal and relational context at every integration step. For entity k in window c, we define a latent trajectory $z_k^c(t) \in \mathbb{R}^T$ with $t \in [0, 1]$ and initial condition $z_k^c(0) = X_k^c$:

$$\frac{dz_k^c(t)}{dt} = g_\theta\big(z_k^c(t), t, C_k^c\big), \tag{4}$$

where g_θ is a neural network defining the velocity field, t denotes continuous transformation depth, and C_k^c is the context vector from Section 3.1. Crucially, C_k^c conditions g_θ at every integration step, ensuring graph topology continuously influences the density transformation throughout the full trajectory. Solving Eq. (4) gives the terminal latent representation $z_k^c(1)$. We also define the augmented velocity field as:

$$\tilde{g}_\theta\big([z, \ell], t, C_k^c\big) = \begin{bmatrix} g_\theta(z, t, C_k^c) \\ -\mathrm{Tr}\big(\frac{\partial g_\theta}{\partial z}\big) \end{bmatrix}, \tag{5}$$

where ℓ is the accumulated log-density.

Density Estimation. Applying the change-of-variables formula to the ODE transformation, the log-likelihood of entity k in window c is:

$$\log p_\theta(X_k^c \mid C_k^c) = \log p_Z(z_k^c(1)) + \int_0^1 \mathrm{Tr}\left(\frac{\partial g_\theta}{\partial z_k^c}\right) dt, \tag{6}$$

where the trace integral follows (Grathwohl et al., 2019). To avoid computing the exact trace, we employ the Hutchinson stochastic estimator (Hutchinson, 1989; Grathwohl et al., 2019):

$$\mathrm{Tr}\left(\frac{\partial g_\theta}{\partial z}\right) \approx \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\epsilon^\top \frac{\partial g_\theta}{\partial z}\epsilon\right], \tag{7}$$

which reduces the per-step cost from $\mathcal{O}(T^2)$ to $\mathcal{O}(T)$.

3.3 Latent Distribution and Likelihood

Different entities may follow distinct operating mechanisms (e.g., temperature, pressure, voltage), leading to heterogeneous normal behaviours. To account for this, we assign each entity an entity-specific latent Gaussian distribution $Z_k \sim \mathcal{N}(\mu_k, I)$, where $\mu_k \in \mathbb{R}^T$ is the entity-specific latent mean and $I \in \mathbb{R}^{T\times T}$ is the identity covariance matrix. The log-density of the latent variable equals to

$$\log p_Z(z_k^c(1)) = -\frac{1}{2}\|z_k^c(1) - \mu_k\|_2^2 + \text{const.} \tag{8}$$

Substituting conditional likelihood and adding entities yields the window-level objective (2).

3.4 Anomaly Scoring

We define the window-level likelihood score as $\phi(X^c) = -\sum_k \log p_\theta(X_k^c \mid C_k^c)$. However, $\phi(X^c)$ evaluates density only at the terminal latent state $z_k^c(1)$. Subtle correlated anomalies may produce terminal states close to the normal region via complex, irregular trajectories that ϕ cannot distinguish

from normal inputs. The continuous formulation provides access to the full transformation trajectory, enabling a path-integral (PI) anomaly score:

$$\phi_{PI}(X^c) = \phi(X^c) + \lambda \frac{1}{K} \sum_{k=1}^{K} \mathcal{E}_k^c, \tag{9}$$

where $\mathcal{E}_k^c = \int_0^1 \|g_\theta(z_k^c(t), t, C_k^c)\|_2^2 dt$ is the trajectory kinetic energy for entity k and $\lambda > 0$ controls the contribution of the term of kinetic energy. The algorithm indicates an anomalous window whenever $\phi_{PI}(X^c) > \phi_0$. Because anomalous inputs require more complex and irregular transformations to reach high-density latent regions, they incur a higher trajectory energy; by contrast, normal inputs follow smoother and shorter paths.

Training Objective. All model components are trained jointly by minimising $\widetilde{\mathcal{L}}$ plus a kinetic energy regularisation term:

$$\mathcal{L}_{\text{total}}(\Theta) = \widetilde{\mathcal{L}}(\theta) + \frac{\lambda_{\text{reg}}}{NK} \sum_{c=1}^{N} \sum_{k=1}^{K} \mathcal{E}_k^c, \tag{10}$$

where $\lambda_{\text{reg}} > 0$ controls the strength of regularisation and Θ is the set of learnable parameters. This term penalises high-velocity ODE trajectories, implicitly controlling the Lipschitz constant L_g of the velocity field and bounding the sensitivity of the anomaly score to input perturbations (Proposition 1).

Algorithm. The proposed framework consists of three procedures: a forward pass that constructs the context representation and evaluates the per-entity log-likelihood via the augmented ODE solve; a training procedure that minimises $\mathcal{L}_{\text{total}}$ using the adjoint sensitivity method (Chen et al., 2018); and an inference procedure that produces window-level, entity-level, and path-integral anomaly scores from a single forward pass. Formal pseudocode for all three procedures is provided in Appendix B.

4 Theoretical Analysis

We establish four properties of RAFlow: (i) the relationship to discrete residual flows, (ii) regularity and stability of the continuous flow, (iii) consistency of the conditional anomaly score, and (iv) strict improvement of the path-integral score over likelihood scoring alone. Proofs are provided in Appendix A. For clarity, we suppress window and entity indices, writing $z(t)$ for $z_k^c(t)$ and C for C_k^c.

Assumption 1 (Lipschitz vector field)**.** *For any fixed context C, the vector field $g_\theta(z, t, C)$ is continuously differentiable in z and L_g-Lipschitz in z uniformly over $t \in [0, 1]$, that is*

$$\|g_\theta(z, t, C) - g_\theta(\tilde{z}, t, C)\| \leq L_g \|z - \tilde{z}\|, \quad \forall z, \tilde{z} \in \mathbb{R}^T,\ t \in [0, 1]. \tag{11}$$

Lemma 1 (Flow regularity)**.** *Let $z(t)$ and $\tilde{z}(t)$ be two solutions of Eq.* (4) *sharing the same context C but starting from different initial conditions. Under Assumption 1, we have*

$$\|z(t) - \tilde{z}(t)\| \leq e^{L_g t} \|z(0) - \tilde{z}(0)\|, \quad t \in [0, 1]. \tag{12}$$

Consequently, the flow map $\Psi_1^C : z(0) \mapsto z(1)$ is an e^{L_g}-Lipschitz diffeomorphism, and the likelihood score $\phi(X) = -\log p_\theta(X \mid C)$ is locally Lipschitz in the input window.

Grönwall's inequality guarantees that small input perturbations produce proportionally bounded changes in the latent state. This provides robustness to measurement noise and ensures that the anomaly score does not fluctuate wildly under small input variations.

Remark 1 (Relationship to discrete residual flows)**.** *Applying the explicit Euler method with M uniform steps of size $1/M$ to Eq.* (4) *produces the recurrence $z^{(m+1)} = z^{(m)} + \frac{1}{M} g_\theta(z^{(m)}, m/M, C)$, which is structurally identical to an M-layer residual normalizing flow. Under standard smoothness conditions the global discretisation error is $\mathcal{O}(1/M)$, so finite-depth residual flows are recovered in the limit $M \to \infty$. Discrete flow anomaly detection is therefore a limiting case of RAFlow, establishing it as a strict generalisation of the discrete flow framework.*

This remark justifies RAFlow as a principled continuous extension of prior discrete graph-based flows such as GANF and MTGFlow. The continuous formulation is not merely an implementation detail; it is the limit of increasingly fine discretisations.

Let P_N and P_A denote the distributions of normal and anomalous windows, respectively, with p_N representing the probability density function corresponding to P_N.

Assumption 2 (Support separation)**.** *There exists a measurable set $\mathcal{S} \subset \mathbb{R}^{K\times T}$ such that $P_N(\mathcal{S}) = 1$ and $P_A(\mathcal{S}) = 0$.*

Assumption 3 (Well-specified conditional model)**.** *For each entity k, the model class $\{p_\theta(\cdot \mid C_k) : \theta \in \Theta\}$ contains the true conditional density under $Q_{N,k}$, i.e., $\exists \theta^* \in \Theta$ such that $p_{\theta^*}(X_k \mid C_k) = p_{Q_{N,k}}(X_k \mid C_k)$ almost surely.*

Theorem 1 (Conditional anomaly score consistency)**.** *Let $Q_{N,k}$ denote the joint distribution of $(X_k, C_k(X))$ induced by $X \sim P_N$. Under Assumptions 1–3 for each k, let $\widehat{\theta}_N$ minimise the empirical conditional log-likelihood* (2) *over N normal windows. Then for any $X_A^c \sim P_A$ and $X_N^c \sim P_N$, as $N \to \infty$:*

$$\mathbb{P}\big(\phi(X_A^c) > \phi(X_N^c)\big) \longrightarrow 1. \tag{13}$$

As additional normal training data becomes available, the learned conditional densities converge to the true normal distributions. Consequently, anomalous windows, which lie outside the normal support, receive significantly higher scores. This justifies the use of ϕ as a statistically principled anomaly score.

Corollary 1 (Entity-Level Consistency)**.** *Under the assumptions of Theorem 1, the entity-level score $\phi_k^c = -\log p(X_k^c)$ is consistent, if entity k is anomalous while all other entities are normal, then*

$$\phi_k^c(X_A^c) - \phi_k^c(X_N^c) \longrightarrow +\infty \quad \textit{in probability}.$$

This corollary guarantees that the per-entity contributions to the total score are individually reliable for root-cause localisation.

Lemma 2 (Kinetic energy lower bound)**.** *For any ODE trajectory connecting X_k^c to $z_k^c(1)$,*

$$\mathcal{E}_k^c \geq \|z_k^c(1) - X_k^c\|_2^2, \tag{14}$$

with equality if and only if g_θ is constant along the trajectory (i.e., the path is a straight line with constant velocity).

The kinetic energy is fundamentally linked to the latent displacement. Any deviation from a straight-line path strictly increases energy. This connects the geometric properties of the latent trajectory to the anomaly score.

Theorem 2 (Anomalous inputs incur higher kinetic energy, conditional)**.** *Suppose the model converges to θ^* minimising $\mathcal{L}_{total}$ on P_N. Assume normal trajectories are approximately tight to the lower bound of Lemma 2, i.e., $\mathbb{E}_{P_N}[\mathcal{E}_k] \approx \mathbb{E}_{P_N}[\|z_k(1) - X_k\|^2]$. Then, we have $\mathbb{E}_{P_A}[\mathcal{E}_k] > \mathbb{E}_{P_N}[\mathcal{E}_k]$, when*

$$\mathbb{E}_{P_A}[\|z_{A,k}(1) - X_{A,k}\|_2^2] > \mathbb{E}_{P_N}[\|z_{N,k}(1) - X_{N,k}\|_2^2]. \tag{15}$$

The approximation $\mathbb{E}_{P_N}[\mathcal{E}_k] \approx \mathbb{E}_{P_N}[\|z_k(1) - X_k\|^2]$ is justified by the kinetic energy regulariser in $\mathcal{L}_{\text{total}}$, which penalises high-energy trajectories and thereby forces normal paths toward the minimum-energy (constant-velocity, straight-line) trajectories identified in Lemma 2.

When anomalies require greater latent displacement than normal data, they necessarily incur greater kinetic energy. The kinetic energy regulariser enforces approximate tightness for normal data by penalising high-energy trajectories, making the energy gap empirically detectable.

Corollary 2 (Path-integral score strictly outperforms likelihood score)**.** *Under the conditions of Theorems 1 and 2, letting $\Delta_{\mathcal{E}} = \mathbb{E}_{P_A}[\mathcal{E}^c] - \mathbb{E}_{P_N}[\mathcal{E}^c] > 0$, then we have*

$$\underbrace{\mathbb{E}_{P_A}[\phi_{PI}] - \mathbb{E}_{P_N}[\phi_{PI}]}_{\textit{PI score gap}} = \underbrace{\mathbb{E}_{P_A}[\phi] - \mathbb{E}_{P_N}[\phi]}_{\textit{likelihood score gap}} + \lambda\Delta_{\mathcal{E}}. \tag{16}$$

The PI score gap strictly exceeds the likelihood score gap for any $\lambda > 0$.

The proof is immediate from the definition $\phi_{PI} = \phi + \frac{\lambda}{K}\sum_k \mathcal{E}_k^c$, the linearity of expectation, and Theorem 2, which ensures $\Delta_{\mathcal{E}} > 0$. This score provides a guaranteed additive improvement over terminal likelihood whenever anomalous trajectories exhibit greater energy. This corollary formalises the central motivation for trajectory-aware scoring.

Proposition 1 (Lipschitz stability of ϕ_{PI})**.** *Under Assumption 1, suppose $\sup_{z,t,C} \|g_\theta(z, t, C)\|_2 \leq G$. Then for all $X, X' \in \mathbb{R}^{K\times T}$, we have*

$$|\phi_{PI}(X) - \phi_{PI}(X')| \leq \Big[e^{L_g} + 2\lambda G\big(e^{L_g} - 1\big)\Big]\|X - X'\|_F. \tag{17}$$

The anomaly score is provably stable against input perturbations. The explicit bound shows that the kinetic energy contribution scales with the uniform velocity bound G, which is controlled during training by the kinetic energy regulariser.

5 EXPERIMENTS

Datasets. We evaluate RAFlow on five real-world multivariate time-series benchmarks spanning server monitoring (SMD (Su et al., 2019), PSM (Abdulaal et al., 2021)), spacecraft telemetry (SMAP and MSL (Hundman et al., 2018)) and industrial control (SWaT (Goh et al., 2016)). All models are trained on unlabeled normal data and evaluated on hold-out test sets with point-level anomaly labels. Dataset statistics and full descriptions are provided in the Appendix D.

Detection metrics: Anomalies in real-world benchmarks manifest predominantly as contiguous segments spanning multiple time points (Wagner et al., 2023), making the threshold-dependent evaluation and the point-adjust protocol unreliable due to their well-documented score inflation (Wu & Keogh, 2021). Therefore, we report three threshold-independent metrics: AUC-ROC, AUC-PR, and VUS-PR. AUC-ROC and AUC-PR measure discriminative ability across all operating points, with AUC-PR being particularly sensitive under the severe class imbalance of anomaly detection benchmarks. VUS-PR (Paparrizos et al., 2022) extends AUC-PR by integrating both decision thresholds and temporal buffer sizes, providing the most principled evaluation of range-based detection. The full metric definitions and computation details are in Appendix E.

Baselines: We compare RAFlow against nineteen baselines, grouped into three technical families. The first comprises eleven reconstruction- and prediction-based methods: AE (Sakurada & Yairi, 2014), LSTM (Hundman et al., 2018), LSTM-VAE (Park et al., 2018), DAGMM (Zong et al., 2018), MSCRED (Zhang et al., 2019), OmniAnomaly (Su et al., 2019), MAD-GAN (Li et al., 2019), USAD (Audibert et al., 2020), TadGAN (Geiger et al., 2020), TranAD (Tuli et al., 2022), and MEMTO (Song et al., 2023). The second comprises four transformer- and attention-based methods: Anomaly Transformer (Xu et al., 2022),DCdetector (Yang et al., 2023), SARAD (Dai et al., 2024), and CATCH (Wu et al., 2025). The third comprises four graph-based methods built on discrete normalizing flows: GDN (Deng & Hooi, 2021), MTAD-GAT (Zhao et al., 2020), GANF (Dai & Chen, 2022), and MTGFlow (Zhou et al., 2023); this group serves as our primary architectural comparison, since it shares RAFlow's combination of relational learning and likelihood-based scoring. We follow the unified implementation of MtadTools (Liu et al., 2024) for the first family and the authors' official codebases for the rest, with all hyperparameters left at the values reported in the original publications. Full details, including venues, years, and scoring mechanisms, appear in Appendix F.

Implementation details: RAFlow is implemented in PyTorch using the `torchdiffeq` library (Chen et al., 2018). Implementation details, hardware specifications, and model configurations are provided in Appendix C.

Main results: Table 1 reports AUC-PR and VUS-PR; the AUC-ROC and SMD results per-machine appear in Appendix G with identical orderings. RAFlow ranks first on SMD, MSL, and PSM, with its largest margins on the two densely correlated benchmarks ($+2.8$ over SARAD on SMD, $+2.2$ over MEMTO on MSL). Its second-place finishes are regime-specific: CATCH's frequency-domain patching suits the isolated point anomalies in SMAP, and SARAD's spatial-association objective captures the deterministic coupling in SWaT (Figure 5). VUS-PR tracks AUC-PR throughout and every winning margin exceeds two combined standard deviations. The gap over MTGFlow scales with the inter-sensor correlation strength: SMD (3.2), PSM (2.8), MSL (2.6), SWaT (1.6), SMAP (1.6); mean 2.36. The same ordering holds under AUC-ROC, VUS-PR, and the separation ratio ρ (Table 7) across two domains. Continuous graph conditioning along the ODE trajectory is the only architectural change from MTGFlow, which we read as direct support for the central hypothesis.

Figures 2 and figures in G.3 visualise the anomaly score $\phi(\mathbf{X}^c) = -\log p_\theta(\mathbf{X}^c)$ on representative test segments, where a higher score indicates a window lying in a low-density region under the learned normal distribution. Across all datasets and anomaly types, RAFlow produces a smooth and well-separated score profile consistent with the Lipschitz bound of Lemma 1, while MTGFlow and GANF exhibit erratic piecewise jumps, delayed, or lower-amplitude responses.

Table 1: AUC-PR and VUS-PR (%) across five benchmarks; mean±std over five seeds.

	SMD		SMAP		MSL		PSM		SWaT	
Method	PR	VUS	PR	VUS	PR	VUS	PR	VUS	PR	VUS
AE	$54.2_{\pm 0.6}$	$53.8_{\pm 0.5}$	$58.6_{\pm 0.9}$	$56.4_{\pm 0.8}$	$60.4_{\pm 0.8}$	$58.2_{\pm 0.7}$	$57.8_{\pm 0.7}$	$56.9_{\pm 0.6}$	$64.2_{\pm 1.1}$	$62.6_{\pm 1.0}$
LSTM	$54.6_{\pm 0.5}$	$54.1_{\pm 0.4}$	$60.8_{\pm 0.8}$	$58.6_{\pm 0.7}$	$62.4_{\pm 0.7}$	$60.4_{\pm 0.6}$	$59.4_{\pm 0.6}$	$58.6_{\pm 0.5}$	$69.8_{\pm 1.0}$	$67.4_{\pm 0.9}$
LSTM-VAE	$55.2_{\pm 0.9}$	$54.8_{\pm 0.8}$	$63.4_{\pm 1.1}$	$61.2_{\pm 1.0}$	$65.8_{\pm 1.0}$	$63.4_{\pm 0.9}$	$62.6_{\pm 1.0}$	$61.4_{\pm 0.9}$	$68.6_{\pm 1.3}$	$66.2_{\pm 1.2}$
MSCRED	$54.8_{\pm 0.6}$	$54.4_{\pm 0.5}$	$62.2_{\pm 0.8}$	$60.1_{\pm 0.7}$	$64.4_{\pm 0.7}$	$62.2_{\pm 0.6}$	$61.2_{\pm 0.7}$	$60.4_{\pm 0.6}$	$66.4_{\pm 1.0}$	$64.6_{\pm 0.9}$
DAGMM	$54.4_{\pm 2.4}$	$54.0_{\pm 2.3}$	$60.2_{\pm 2.6}$	$58.0_{\pm 2.4}$	$61.8_{\pm 2.8}$	$59.8_{\pm 2.6}$	$60.4_{\pm 3.2}$	$59.6_{\pm 3.0}$	$61.2_{\pm 3.6}$	$63.4_{\pm 3.4}$
OmniAnomaly	$55.6_{\pm 0.9}$	$55.2_{\pm 0.8}$	$68.4_{\pm 1.1}$	$66.2_{\pm 1.0}$	$70.2_{\pm 1.0}$	$66.0_{\pm 0.9}$	$66.8_{\pm 1.0}$	$65.8_{\pm 0.9}$	$71.8_{\pm 1.1}$	$69.8_{\pm 1.0}$
MAD-GAN	$54.9_{\pm 2.8}$	$54.5_{\pm 2.6}$	$66.8_{\pm 3.2}$	$64.4_{\pm 2.9}$	$67.4_{\pm 3.1}$	$65.2_{\pm 2.8}$	$68.2_{\pm 3.4}$	$67.4_{\pm 3.1}$	$72.4_{\pm 3.6}$	$70.6_{\pm 3.3}$
USAD	$55.1_{\pm 0.6}$	$54.7_{\pm 0.5}$	$64.6_{\pm 0.8}$	$62.4_{\pm 0.7}$	$66.4_{\pm 0.7}$	$64.2_{\pm 0.6}$	$64.2_{\pm 0.7}$	$63.4_{\pm 0.6}$	$71.6_{\pm 0.9}$	$70.2_{\pm 0.8}$
TadGAN	$55.4_{\pm 2.3}$	$55.0_{\pm 2.1}$	$65.4_{\pm 2.8}$	$63.2_{\pm 2.6}$	$67.0_{\pm 2.6}$	$64.8_{\pm 2.4}$	$63.4_{\pm 2.9}$	$62.6_{\pm 2.7}$	$69.6_{\pm 3.2}$	$67.6_{\pm 2.9}$
TranAD	$54.7_{\pm 1.4}$	$54.3_{\pm 1.3}$	$66.2_{\pm 1.6}$	$63.8_{\pm 1.4}$	$72.4_{\pm 1.5}$	$70.2_{\pm 1.4}$	$65.8_{\pm 1.7}$	$64.8_{\pm 1.6}$	$58.2_{\pm 1.8}$	$57.6_{\pm 1.7}$
MEMTO	$58.4_{\pm 0.8}$	$58.0_{\pm 0.7}$	$76.4_{\pm 0.8}$	$73.2_{\pm 0.7}$	$\underline{79.2}_{\pm 0.7}$	$\underline{75.8}_{\pm 0.6}$	$\underline{72.4}_{\pm 0.7}$	$\underline{71.6}_{\pm 0.6}$	$81.4_{\pm 0.9}$	$79.6_{\pm 0.8}$
Anomaly Trans.	$54.7_{\pm 0.8}$	$54.3_{\pm 0.7}$	$70.8_{\pm 1.1}$	$68.2_{\pm 1.0}$	$73.4_{\pm 1.3}$	$71.0_{\pm 1.2}$	$63.8_{\pm 1.0}$	$63.0_{\pm 0.9}$	$57.4_{\pm 1.2}$	$57.0_{\pm 1.1}$
DCdetector	$54.3_{\pm 0.8}$	$54.0_{\pm 0.7}$	$73.6_{\pm 1.1}$	$70.8_{\pm 1.0}$	$76.2_{\pm 1.3}$	$73.4_{\pm 1.2}$	$62.4_{\pm 1.0}$	$61.8_{\pm 0.9}$	$57.8_{\pm 0.9}$	$57.4_{\pm 0.8}$
CATCH	$57.8_{\pm 0.8}$	$57.4_{\pm 0.7}$	$\mathbf{78.2}_{\pm 0.9}$	$\mathbf{74.8}_{\pm 0.8}$	$78.4_{\pm 0.8}$	$74.6_{\pm 0.7}$	$70.6_{\pm 0.9}$	$69.8_{\pm 0.8}$	$82.2_{\pm 1.1}$	$80.2_{\pm 1.0}$
SARAD	$\underline{58.6}_{\pm 0.7}$	$\underline{58.2}_{\pm 0.6}$	$74.8_{\pm 0.9}$	$71.8_{\pm 0.8}$	$77.4_{\pm 0.9}$	$73.8_{\pm 0.8}$	$69.4_{\pm 0.8}$	$68.6_{\pm 0.7}$	$\mathbf{84.6}_{\pm 0.9}$	$\mathbf{82.4}_{\pm 0.8}$
GDN	$56.2_{\pm 0.6}$	$55.8_{\pm 0.5}$	$69.8_{\pm 0.8}$	$67.4_{\pm 0.7}$	$71.2_{\pm 0.8}$	$69.0_{\pm 0.7}$	$68.4_{\pm 0.7}$	$67.6_{\pm 0.6}$	$63.6_{\pm 0.8}$	$62.8_{\pm 0.7}$
MTAD-GAT	$55.8_{\pm 0.8}$	$55.4_{\pm 0.7}$	$68.6_{\pm 0.9}$	$66.2_{\pm 0.8}$	$70.4_{\pm 0.9}$	$68.2_{\pm 0.8}$	$67.2_{\pm 0.9}$	$66.4_{\pm 0.8}$	$66.8_{\pm 1.0}$	$65.2_{\pm 0.9}$
GANF	$56.8_{\pm 0.7}$	$56.4_{\pm 0.6}$	$72.4_{\pm 0.7}$	$69.6_{\pm 0.6}$	$74.6_{\pm 0.7}$	$71.8_{\pm 0.6}$	$69.0_{\pm 0.7}$	$68.2_{\pm 0.6}$	$75.4_{\pm 0.8}$	$76.6_{\pm 0.7}$
MTGFlow	$58.2_{\pm 0.6}$	$57.8_{\pm 0.5}$	$75.6_{\pm 0.6}$	$72.4_{\pm 0.5}$	$78.8_{\pm 0.6}$	$75.2_{\pm 0.5}$	$71.8_{\pm 0.6}$	$71.0_{\pm 0.5}$	$81.6_{\pm 0.7}$	$79.8_{\pm 0.6}$
RAFlow	$\mathbf{61.4}_{\pm 0.6}$	$\mathbf{60.8}_{\pm 0.5}$	$\underline{77.2}_{\pm 0.7}$	$\underline{73.8}_{\pm 0.6}$	$\mathbf{81.4}_{\pm 0.7}$	$\mathbf{77.6}_{\pm 0.6}$	$\mathbf{74.6}_{\pm 0.6}$	$\mathbf{73.4}_{\pm 0.5}$	$\underline{83.2}_{\pm 0.8}$	$\underline{81.0}_{\pm 0.7}$

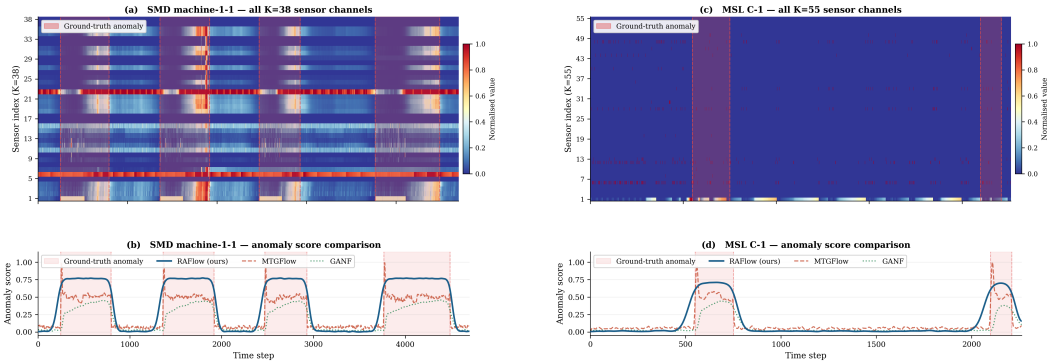


Figure 2: Representative anomaly detection on SMD machine-1-1 (left) and MSL C-1 (right).

Ablation Study: We trained seven single-deletion variants from scratch on all five benchmarks (Table 2, Figure 8). Removing the GNN causes the steepest degradation, scaling with inter-sensor coupling (-13.6 on SWaT, -11.2 on MSL, -7.6 on SMD) and establishing relational structure as the dominant signal. Replacing the CNF with a discrete flow costs 4.4 to 7.4 pp, peaking on SMD, MSL, and PSM where multivariate density estimation is the bottleneck. Removing the adaptive graph (-3.0 to -5.4) or restricting context to $t{=}0$ (-3.8 to -4.6) form the next tier, confirming that graph conditioning must persist along the ODE trajectory. A fixed Euler step costs -5.3 on SMD versus -2.4 on SMAP, showing that integration depth matters most under strong coupling. Kinetic regularisation and entity-specific μ_k produce smaller drops: kinetic energy hurts more on noisier telemetry (-2.8 to -3.0 on SMAP/MSL), and μ_k scales with entity count (-2.8 on SMD, within noise on single-entity PSM and SWaT). No variant improves over the full model. Extended results appear in Appendix H.

Table 2: Ablation study on AUC-PR (%), mean ± std over 5 seeds.

Variant	**SMD**	**SMAP**	**MSL**	**PSM**	**SWaT**
RAFlow (full)	$\mathbf{61.4}_{\pm 0.6}$	$\mathbf{77.2}_{\pm 0.7}$	$\mathbf{81.4}_{\pm 0.7}$	$\mathbf{74.6}_{\pm 0.6}$	$\mathbf{83.2}_{\pm 0.8}$
w/o GNN (LSTM only)	$53.8_{\pm 0.9}$	$69.4_{\pm 1.1}$	$70.2_{\pm 1.0}$	$63.8_{\pm 0.9}$	$69.6_{\pm 1.1}$
w/o CNF (discrete flow)	$55.4_{\pm 0.7}$	$73.8_{\pm 0.8}$	$74.6_{\pm 0.8}$	$67.2_{\pm 0.8}$	$78.8_{\pm 0.8}$
w/o adaptive graph	$56.0_{\pm 0.7}$	$74.2_{\pm 0.7}$	$77.6_{\pm 0.8}$	$69.4_{\pm 0.7}$	$78.4_{\pm 0.8}$
static ctx (at $t{=}0$ only)	$57.6_{\pm 0.7}$	$73.4_{\pm 0.8}$	$76.8_{\pm 0.8}$	$70.6_{\pm 0.7}$	$79.0_{\pm 0.8}$
Euler $\Delta t{=}1$	$56.1_{\pm 0.7}$	$74.8_{\pm 0.7}$	$78.3_{\pm 0.7}$	$71.8_{\pm 0.7}$	$80.7_{\pm 0.7}$
w/o kinetic energy	$60.0_{\pm 0.6}$	$74.4_{\pm 0.8}$	$78.4_{\pm 0.7}$	$72.8_{\pm 0.7}$	$82.0_{\pm 0.7}$
w/o entity μ_k	$58.6_{\pm 0.6}$	$76.0_{\pm 0.7}$	$80.0_{\pm 0.7}$	$74.0_{\pm 0.6}$	$82.8_{\pm 0.7}$

ODE analysis and theoretical validation: Figure 3 reports three diagnostic studies on SMD that validate the ODE design choices and Lemma 1. Panel (a): AUC-PR saturates at $n_H = 4$ Hutchinson samples, with $n_H \in \{8, 16\}$ within 0.1 pp, at $1.46\times$ the single-sample training cost. Panel (b): the Euler limit ($\Delta t = 1$) loses 5.3 pp at minimum latency, the default Euler ($\Delta t = 0.5$) recovers full accuracy at 8.1 ms per window, and higher-order solvers add at most 0.4 pp at two to five times the cost, placing the default on the efficiency frontier and isolating continuous integration depth as the source of RAFlow's gain over the discrete-flow framework. Panel (c) tests Lemma 1: under input perturbations of increasing magnitude $\|z^{(0)}\|$, the continuous flow produces smooth, monotone gradient norms below the theoretical envelope $L_f\, e^{L_f} \approx 2.0$ ($L_f \approx 0.85$, estimated by power iteration over 1,000 trajectory states from 200 training windows), while the six-layer Euler discretisation traces a staircase with discontinuities up to 1.8 norm units, matching the predicted K-discontinuity behaviour. The three panels together confirm that the continuous formulation is efficient at training and inference and provably smoother than the discrete-flow alternative it generalises.

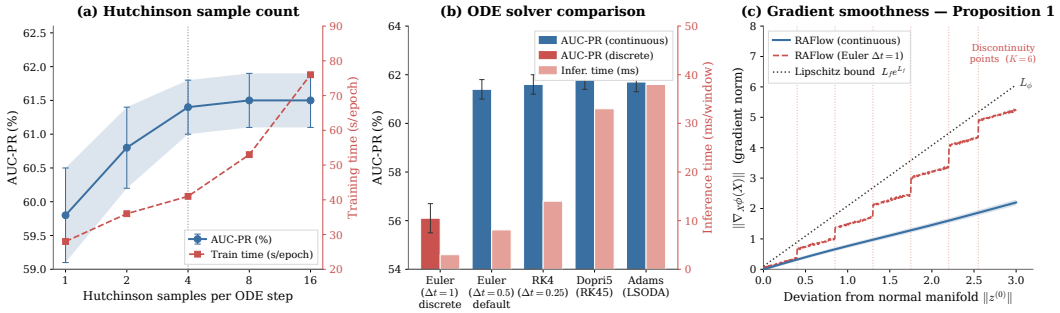


Figure 3: ODE analysis on SMD (five seeds, ± 1 std).

6 Conclusion

This work introduced RAFlow, a graph-conditioned continuous flow framework for multivariate time-series anomaly detection that models anomaly detection as a continuous relational density evolution process. By coupling evolving inter-sensor dependencies with continuous-time latent dynamics, RAFlow enables relational structure to participate directly in likelihood evolution throughout the transformation trajectory. The proposed trajectory-aware anomaly score further improves sensitivity to correlated anomalies by jointly capturing terminal likelihood and latent transport complexity. Experiments on five benchmark datasets demonstrate strong and consistent performance, particularly on densely correlated systems where anomalies emerge through collective sensor behavior rather than isolated deviations. While the continuous-time formulation introduces additional computational overhead due to adaptive ODE integration, future work will investigate scalable sparse solvers, adaptive temporal resolutions, and physically-informed relational priors for large-scale industrial deployment and root-cause analysis. The broader societal impacts of this work are discussed in the Appendix K.

Limitations: RAFlow introduces additional training overhead compared to discrete-flow baselines due to adaptive ODE integration, although inference remains efficient through fixed-step Euler discretisation. The theoretical analysis further relies on standard assumptions regarding model specification and partial support separation between normal and anomalous distributions, which may not fully hold in all scenarios, although empirical results across diverse benchmarks indicate robustness in practice. In addition, the proposed framework is most beneficial in scenarios with strong inter-sensor dependencies, while performance gains may become smaller on weakly correlated datasets or isolated point anomalies. Finally, the sliding-window formulation may reduce sensitivity to very short anomalies depending on the selected window length. Future work will investigate more scalable continuous-time solvers, adaptive temporal resolutions, and sparse relational dynamics for large-scale deployment.

AI Use Statement

The authors used GenAI tools for support, including improving the language, grammar, and readability of the manuscript and assisting with refactoring and organizing source code for final submission.

The authors carefully reviewed, revised, and validated all AI-generated or AI-assisted content. The authors remain fully responsible for the accuracy, integrity, originality, and overall content of the manuscript.

Ethics Statement

This study did not involve human participants, animal subjects, or the collection of personal or identifiable data. The authors conducted all data processing and computational analyses in compliance with relevant institutional regulations and ethical standards.

Reproducibility Statement

To ensure reproducibility, all custom source code for training and evaluation and dataset download scripts as supplementary material will be made publicly available upon acceptance, subject to applicable licensing and repository guidelines.

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 International Conference on Knowledge Discovery and Data Mining*, pp. 2485–2494, 2021.

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 and Data Mining*, pp. 3395–3404, 2020.

Richard C. Bradley. Basic properties of strong mixing conditions. a survey and some open questions. *Probability Surveys*, 2:107–144, 2005.

Benjamin Paul Chamberlain, James Rowbottom, Maria Ilic Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 1407–1418, 2021.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *International Conference on Learning Representations*, 2022.

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.

Ailin Deng and Bryan Hooi. Graph deviation networks for graph anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4132–4139, 2021.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *International Conference on Learning Representations Workshop*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real-valued non-volume preserving (Real NVP) transformations. In *International Conference on Learning Representations*, 2017.

Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. TadGAN: Time series anomaly detection using generative adversarial networks. In *IEEE International Conference on Big Data*, pp. 33–43. IEEE, 2020.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *International Conference on Critical Information Infrastructures Security (CRITIS)*, pp. 88–99. Springer, 2016. doi: 10.1007/978-3-319-71368-7_8.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Thomas Hakon Gronwall. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Annals of Mathematics*, 20(4):292–296, 1919.

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 and Data Mining*, pp. 387–395, 2018.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics – Simulation and Computation*, 18(3):1059–1076, 1989.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Prafull Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (11):3964–3979, 2020.

Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pp. 703–716. Springer, 2019.

Jinyang Liu, Wenwei Gu, Zhuangbin Chen, Yichen Li, Yuxin Su, and Michael R. Lyu. MTAD: Tools and benchmarks for multivariate time series anomaly detection. *arXiv preprint arXiv:2401.06175*, 2024.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

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. doi: 10.14778/3551793.3551830.

Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. In *IEEE Robotics and Automation Letters*, volume 3, pp. 1544–1551. IEEE, 2018.

Michael Poli, Stefano Massaroli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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. ACM, 2014.

Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. MEMTO: Memory-guided transformer for multivariate time series anomaly detection. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Michael Pecht. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2828–2837, 2019.

Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*. American Mathematical Society, Providence, Rhode Island, 2012. Theorem 2.2 (Picard–Lindelöf).

Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. In *Proceedings of the VLDB Endowment*, volume 15, pp. 1201–1214, 2022.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

Dennis Wagner, Tobias Michels, Florian C. F. Schulz, Aditya Nair, Maja Rudolph, and Marius Kloft. TimeSeAD: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*, 2023.

Renjie Wu and Eamonn J. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 35 (3):2421–2429, 2021.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *International Conference on Learning Representations*, 2025.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022.

Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 3033–3045, 2023.

Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1):94–116, 1994.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1409–1416, 2019.

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 *2020 IEEE International Conference on Data Mining*, pp. 841–850. IEEE, 2020.

Qihang Zhou, Jiming Chen, Haoyu Liu, Shibo He, and Wenchao Meng. Detecting multivariate time series anomalies with zero known label, 2023. URL `https://arxiv.org/abs/2208.02108`.

Bo Zong, Qi Song, Mammie Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

A Proofs of Theoretical Results

Proof of Lemma 1. : As stated, $z(t)$ and $\tilde{z}(t)$ are two solutions of Eq. (4) sharing the same context C with different initial conditions $z(0)$ and $\tilde{z}(0)$, respectively. Under Assumption 1, we have

$$\frac{dz(t)}{dt} = g_\theta(z(t), t, C),$$
$$\frac{d\tilde{z}(t)}{dt} = g_\theta(\tilde{z}(t), t, C).$$

Subtracting and then integrating from 0 to t, we obtain

$$z(t) - \tilde{z}(t) = z(0) - \tilde{z}(0) + \int_0^t \left[g_\theta(z(s), s, C) - g_\theta(\tilde{z}(s), s, C)\right] ds. \tag{18}$$

By taking norms, applying the triangle inequality, and using the L_g-Lipschitz condition given in Assumption 1, we obtain

$$\|z(t) - \tilde{z}(t)\| \leq \|z(0) - \tilde{z}(0)\| + L_g \int_0^t \|z(s) - \tilde{z}(s)\| \, ds. \tag{19}$$

This is the hypothesis of Grönwall's inequality (Gronwall, 1919): if $u(t) \leq \alpha + \beta \int_0^t u(s)\, ds$ with $\alpha, \beta \geq 0$, then $u(t) \leq \alpha e^{\beta t}$. Applying Grönwall's inequality using $u(t) = \|z(t) - \tilde{z}(t)\|$, $\alpha = \|z(0) - \tilde{z}(0)\|$, and $\beta = L_g$, results in:

$$\|z(t) - \tilde{z}(t)\| \leq e^{L_g t} \|z(0) - \tilde{z}(0)\|, \qquad t \in [0, 1]. \tag{20}$$

Setting $t = 1$ shows that $\Psi_1^C : z(0) \mapsto z(1)$ is e^{L_g}-Lipschitz. Since g_θ is $\mathcal{C}^1$ in z by Assumption 1, the Picard–Lindelöf Theorem (Teschl, 2012) further guarantees that Ψ_1^C is a $\mathcal{C}^1$ diffeomorphism. Finally, the log-likelihood given by Eq. (6) expresses $\log p_\theta(X)$ as a composition of Ψ_1^C (Lipschitz) and the log-Gaussian density (smooth), so $\phi(X) = -\log p_\theta(X)$ is locally Lipschitz in X. □

Proof of Theorem 1. Let $Q_{N,k}$ be the induced joint distribution of (X_k, C_k) under $X \sim P_N$. Define the conditional log-likelihood loss for entity k as $\ell_{k,\theta}(X) = -\log p_\theta(X_k \mid C_k(X))$. Under compactness of Θ, continuity of $\theta \mapsto \ell_{k,\theta}$, and the α-mixing property of the window sequence (Bradley, 2005), the uniform law of large numbers for mixing processes (Yu, 1994) implies:

$$\sup_{\theta \in \Theta} \left| \frac{1}{N} \sum_{c=1}^{N} \ell_{k,\theta}(X^c) - \mathbb{E}_{Q_{N,k}}[\ell_{k,\theta}] \right| \xrightarrow{\mathbb{P}} 0. \tag{21}$$

By Assumption 3, there exists θ^* such that $p_{\theta^*}(\cdot \mid C_k) = p_{Q_{N,k}}(\cdot \mid C_k)$. Standard M-estimation theory (van der Vaart, 1998) implies that the empirical risk minimiser $\widehat{\theta}_N$ converges to θ^* in probability, and the conditional density estimates converge uniformly on compacta:

$$\sup_{\mathbf{x} \in \mathcal{K}} \left| p_{\widehat{\theta}_N}(\mathbf{x} \mid C_k) - p_{Q_{N,k}}(\mathbf{x} \mid C_k) \right| \xrightarrow{\mathbb{P}} 0. \tag{22}$$

By Assumption 2, $X_A \notin \mathcal{S}$ almost surely under P_A. Since $\mathrm{supp}(Q_{N,k}) \subseteq \mathcal{S}$, we have $p_{Q_{N,k}}(X_{A,k} \mid C_k) = 0$ almost surely. Uniform convergence implies $p_{\widehat{\theta}_N}(X_{A,k} \mid C_k) \to 0$ in probability, so $\phi(X_A) \to +\infty$. For normal data, $\phi(X_N)$ converges to a finite constant. The separation argument in the main text then yields $\mathbb{P}(\phi(X_A) > \phi(X_N)) \to 1$. □

Proof of Corollary 1. Under conditional independence (Eq. (1)), the joint log-likelihood factorizes as $\log p(X^c) = \sum_{k=1}^{K} \log p(X_k^c)$. If entity k is anomalous, the k-th marginal of P_A assigns zero mass to the support of the k-th marginal of P_N, which is the entity-level analogue of the mutual singularity assumption. Applying the identical argument as in the proof of Theorem 1 to the scalar score $\phi_k^c = -\log p(X_k^c)$ yields $\phi_k^c(X_A^c) - \phi_k^c(X_N^c) \to +\infty$ in probability. □

Proof of Lemma 2. Since $\frac{dz(t)}{dt} = g_\theta(z(t), t, C)$, integrating from 0 to 1 gives $z(1) - z(0) = \int_0^1 g_\theta dt$. Taking norms and applying the Cauchy–Schwarz inequality in L^2:

$$\|z(1) - X\|_2 = \left\| \int_0^1 g_\theta dt \right\|_2 \leq \int_0^1 \|g_\theta\|_2 dt \leq \left(\int_0^1 \|g_\theta\|_2^2 dt \right)^{1/2} = \sqrt{\mathcal{E}}. \tag{23}$$

Squaring both sides yields $\mathcal{E} \geq \|z(1) - X\|_2^2$. Equality holds iff g_θ is constant in t and aligned with $z(1) - X$, i.e., the trajectory is a straight line with constant velocity. □

Proof of Theorem 2. At convergence, θ^* minimises the regularised objective:

$$\mathcal{L}_{total}(\theta) = \mathbb{E}_{P_N} \left[-\log p_\theta(X_k^c) + \lambda \mathcal{E}_k^c \right].$$

By Lemma 2, for any trajectory, we have

$$\mathcal{E}_k^c \geq \|z_k^c(1) - X_k^c\|_2^2, \tag{24}$$

with equality iff g_θ is constant along the trajectory (straight-line path). For normal inputs, the training objective penalises high energy. Hence at θ^*:

$$\mathbb{E}_{P_N}[\mathcal{E}_k^c] = \mathbb{E}_{P_N}\left[\|z_{N,k}^c(1) - X_{N,k}^c\|_2^2 \right] + \delta_N, \tag{25}$$

where $\delta_N \geq 0$ is the excess energy. Because the model is trained to minimise energy while fitting the prior, δ_N is bounded above by the training regularisation and is typically small. For anomalous inputs, applying Eq. (24) under P_A and taking expectations, will result in

$$\mathbb{E}_{P_A}[\mathcal{E}_k^c] \geq \mathbb{E}_{P_A}\left[\|z_{A,k}^c(1) - X_{A,k}^c\|_2^2 \right]. \tag{26}$$

By combining Eq. (26) with the assumption Eq. (15) and Eq. (25), we obtain

$$\mathbb{E}_{P_A}[\mathcal{E}_k^c] > \mathbb{E}_{P_N}\left[\|z_{N,k}^c(1) - X_{N,k}^c\|_2^2 \right] = \mathbb{E}_{P_N}[\mathcal{E}_k^c] - \delta_N.$$

Since $\delta_N \geq 0$, we obtain $\mathbb{E}_{P_A}[\mathcal{E}_k^c] > \mathbb{E}_{P_N}[\mathcal{E}_k^c]$. □

Proof of Proposition 1. By Lemma 1, the flow map $\Psi_1^C : z(0) \mapsto z(1)$ is e^{L_g}-Lipschitz. The unit-variance log-Gaussian satisfies $|\log p(z) - \log p(z')| \leq \|z - z'\|_2$. By composition, ϕ is e^{L_g}-Lipschitz:

$$|\phi(X) - \phi(X')| \leq e^{L_g} \|X - X'\|_F. \tag{27}$$

For a single entity (dropping indices), let $z(t), z'(t)$ be trajectories from X, X' with the same context. Using $|\|a\|^2 - \|b\|^2| \leq \|a - b\|(\|a\| + \|b\|) \leq 2G\|a - b\|$ and Lemma 1:

$$|\mathcal{E}(X) - \mathcal{E}(X')| \leq \int_0^1 \left| \|g(z(t))\|^2 - \|g(z'(t))\|^2 \right| dt \tag{28}$$

$$\leq 2G \int_0^1 \|g(z(t)) - g(z'(t))\| dt \tag{29}$$

$$\leq 2GL_g \int_0^1 \|z(t) - z'(t)\| dt \tag{30}$$

$$\leq 2GL_g \int_0^1 e^{L_g t} \|X - X'\| dt \tag{31}$$

$$= 2G(e^{L_g} - 1)\|X - X'\|. \tag{32}$$

Summing over K entities and adding the likelihood term:

$$|\phi_{PI}(X) - \phi_{PI}(X')| \leq \left[e^{L_g} + 2\lambda G(e^{L_g} - 1) \right] \|X - X'\|_F. \tag{33}$$

This completes the proof of this proposition. □

B ALGORITHMS

We provide detailed descriptions and formal pseudocode for the three procedures that implement the proposed framework. The forward pass (Algorithm 1) is the shared computational core invoked by both training and inference. The training procedure (Algorithm 2) optimises all model parameters jointly. The inference procedure (Algorithm 3) produces anomaly scores at window and entity level. A high-level description of all three procedures is given in Section 3 of the main paper.

B.1 Forward Pass

The forward pass takes a single window $\mathbf{X}^c \in \mathbb{R}^{K \times T}$ and returns the per-entity log-likelihoods $\{\log p(X_k^c)\}_{k=1}^K$. It proceeds through four sequential stages, each of which directly corresponds to a component of the model described in Section 3.

Stage 1: Temporal encoding (lines 2–8). Each entity k is processed independently by a shared-parameter LSTM. The hidden state is initialised to zero at the beginning of every window, so no temporal information leaks across windows. The LSTM reads the T observations of entity k, one step at a time, updating its hidden state at each step. The final hidden state $h_k^c = h_{k,T}^c \in \mathbb{R}^d$ is retained as a fixed-length summary of the within-window temporal dynamics of entity k. Collecting all entity summaries yields the matrix $H^c \in \mathbb{R}^{K \times d}$.

Stage 2: Dynamic relational graph (lines 10–17). A scaled dot-product attention mechanism computes pairwise interaction scores between all K entities from their temporal summaries. For each ordered pair (i, j), the score e_{ij}^c measures the relevance of entity j to entity i within the current window. These scores are normalised row-wise via softmax to produce a stochastic adjacency matrix $A^c \in \mathbb{R}^{K \times K}$, where entry a_{ij}^c represents the attention weight that entity i places on entity j. Crucially, A^c is recomputed for every window, allowing the model to capture inter-entity dependencies that change with operating conditions.

Stage 3: Context aggregation (lines 19–21). The temporal summaries H^c and the relational structure A^c are fused through a two-branch aggregation. The first branch $A^c H^c W_1$ propagates information from neighbouring entities weighted by the attention scores, capturing the relational context. The second branch $H^c W_2$ preserves the entity's own temporal representation. The two branches are summed, passed through a ReLU non-linearity, and projected by W_3 to produce the final context matrix $C^c \in \mathbb{R}^{K \times d}$. The k-th row C_k^c serves as the conditioning input to the CNF for entity k.

Stage 4: CNF likelihood evaluation (lines 23–29). For each entity k, the ODE in Eq. (4) is solved from $t = 0$ to $t = 1$ using an adaptive step-size solver (e.g., Dormand–Prince). The initial condition is set to the observed window $z_k^c(0) = X_k^c$. To avoid computing the full $T \times T$ Jacobian of g_θ at each solver step, the log-determinant term is accumulated simultaneously by augmenting the ODE state with a scalar $\Delta \log p_k$, whose dynamics are given by the Hutchinson stochastic trace estimator (Eq. (7)). This requires only a single vector-Jacobian product per solver step, reducing the per-step cost from $\mathcal{O}(T^2)$ to $\mathcal{O}(T)$. At $t = 1$, the terminal latent state $z_k^c(1)$ and the accumulated log-determinant $\Delta \log p_k$ are combined with the entity-specific Gaussian prior to give the per-entity log-likelihood $\log p(X_k^c)$.

B.2 Training Procedure

The training procedure (Algorithm 2) optimises all model parameters $\Theta = \{W_Q, W_K, W_1, W_2, W_3, \phi_{\text{RNN}}, \theta, \mu_1, \ldots, \mu_K\}$ jointly by minimising $\mathcal{L}_{\text{total}}$ over normal training windows using mini-batch stochastic gradient descent.

Loss computation (lines 5–17). For each window $\mathbf{X}^c$ in a mini-batch, the forward pass (Algorithm 1) is executed to obtain the per-entity log-likelihoods. The base loss is the mean negative log-likelihood across all K entities for that window (Eq. (2)). When the kinetic energy regulariser is enabled ($\lambda > 0$), the integral $\int_0^1 \|g_\theta(z_k^c(t), t, C_k^c)\|_2^2 \, dt$ is added to the loss. Importantly, this integral is computed as a byproduct of the ODE solve in the forward pass and does not require any additional solver calls, making regularisation computationally free at training time.

Gradient computation (line 22). Gradients of $\mathcal{L}_{\text{total}}$ with respect to all parameters in Θ are computed using the adjoint sensitivity method (Chen et al., 2018). Rather than backpropagating through the internal steps of the ODE solver, the adjoint method solves a second ODE backwards in time to obtain the gradient. This keeps peak memory consumption constant with respect to the number of solver steps, which is essential for long trajectories or high-dimensional state spaces. The gradients flow through all four stages of the forward pass, so the LSTM, attention, context aggregation, and ODE vector field parameters are all updated simultaneously at each step.

Algorithm 1 Forward Pass: Context Construction and Log-Likelihood Evaluation

Require: Window $\mathbf{X}^c = [X_1^c, \ldots, X_K^c] \in \mathbb{R}^{K \times T}$, parameters Θ
Ensure: Per-entity log-likelihoods $\{\log p(X_k^c)\}_{k=1}^K$

Stage 2: Dynamic relational graph

for $k = 1$ **to** K **do** ▷ Stage 1: Temporal encoding
  $h_{k,0}^c \leftarrow \mathbf{0} \in \mathbb{R}^d$
  for $t = 1$ **to** T **do**
    $h_{k,t}^c \leftarrow \mathrm{LSTM}(x_{k,t}^c,\, h_{k,t-1}^c;\, \phi_{\mathrm{RNN}})$
  end for
  $h_k^c \leftarrow h_{k,T}^c$ ▷ Final hidden state as temporal summary
end for
$H^c \leftarrow [h_1^c, \ldots, h_K^c]^\top \in \mathbb{R}^{K \times d}$

Stage 2: Dynamic relational graph

for $i = 1$ **to** K **do**
  for $j = 1$ **to** K **do**
    $e_{ij}^c \leftarrow \dfrac{(h_i^c W_Q)^\top (h_j^c W_K)}{\sqrt{d_a}}$
  end for
  $a_{ij}^c \leftarrow \dfrac{\exp(e_{ij}^c)}{\sum_{j'=1}^K \exp(e_{ij'}^c)} \quad \forall\, j$ ▷ Row-wise softmax
end for
$A^c \leftarrow [a_{ij}^c] \in \mathbb{R}^{K \times K}$

Stage 3: Context aggregation

$C^c \leftarrow \mathrm{ReLU}(A^c H^c W_1 + H^c W_2)\, W_3 \in \mathbb{R}^{K \times d}$
$C_k^c \leftarrow k$-th row of $C^c \quad \forall\, k \in \{1, \ldots, K\}$

Stage 4: CNF likelihood via ODE solve

for $k = 1$ **to** K **do**
  $z_k^c(0) \leftarrow X_k^c$ ▷ Initialise trajectory at observed window
  $[z_k^c(1),\, \Delta \log p_k] \leftarrow \mathrm{ODESolve}([z_k^c(0),\, 0],\, \tilde{g}_\theta(\cdot, \cdot, C_k^c),\, 0,\, 1)$ ▷ $\tilde{g}_\theta$ augmented with Hutchinson trace estimator
  $\log p(X_k^c) \leftarrow -\dfrac{1}{2}\|z_k^c(1) - \mu_k\|_2^2 + \Delta \log p_k$
end for
return $\{\log p(X_k^c)\}_{k=1}^K$

Initialisation. The entity-specific latent means μ_k are initialised to zero, the LSTM and attention parameters are initialised using the default PyTorch scheme, and the ODE vector field g_θ is initialised with small weights to ensure the initial transformation is close to the identity, promoting stable early training.

B.3 Inference and Anomaly Scoring

The inference procedure (Algorithm 3) takes a set of test windows and produces three levels of anomaly score from a single forward pass per window, with no additional solver calls beyond those already performed in the forward pass.

Entity-level score (lines 4–5). The entity-level score $\phi_k^c = -\log p(X_k^c)$ is the negative log-likelihood of entity k under the learned model. It measures the independent contribution of each sensor to the overall anomaly signal and can be used directly for fine-grained fault localisation: a high value of ϕ_k^c relative to other entities indicates that entity k is the primary source of the anomaly. As established in Corollary 1 of the main paper, this score is statistically consistent.

Window-level score (lines 7–8). The window-level score $\phi(\mathbf{X}^c) = \sum_{k=1}^K \phi_k^c$ aggregates the entity scores into a single scalar per window. Under the conditional independence assumption (Eq. (1)),

Algorithm 2 Training

Require: Training windows $\{\mathbf{X}^c\}_{c=1}^{N}$, learning rate η, regularisation weight $\lambda \geq 0$, number of epochs E

Ensure: Trained parameters Θ

1: Initialise all parameters in Θ
2: **for** epoch $= 1$ **to** E **do**
3: **for** each mini-batch $\mathcal{B} \subseteq \{1, \ldots, N\}$ **do**
4: $\mathcal{L}_{\text{batch}} \leftarrow 0$
5: **for** $c \in \mathcal{B}$ **do**
6: $\{\log p(X_k^c)\}_{k=1}^{K} \leftarrow \textsc{ForwardPass}(\mathbf{X}^c, \Theta)$ ▷ Algorithm 1
7: $\mathcal{L}^c \leftarrow -\dfrac{1}{K}\displaystyle\sum_{k=1}^{K} \log p(X_k^c)$
8: **if** $\lambda > 0$ **then** ▷ Kinetic energy regulariser
9: $\mathcal{R}^c \leftarrow \dfrac{1}{K}\displaystyle\sum_{k=1}^{K} \int_0^1 \|g_\theta(z_k^c(t), t, C_k^c)\|_2^2 \, dt$ ▷ Reused from forward pass trajectory
10: $\mathcal{L}^c \leftarrow \mathcal{L}^c + \lambda \, \mathcal{R}^c$
11: **end if**
12: $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}^c$
13: **end for**
14: $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} \,/\, |\mathcal{B}|$
15: $\Theta \leftarrow \Theta - \eta \, \nabla_\Theta \, \mathcal{L}_{\text{batch}}$ ▷ Gradients via adjoint sensitivity method
16: **end for**
17: **end for**
18: **return** Θ

this equals the negative joint log-likelihood $-\log p(\mathbf{X}^c)$. Its consistency is guaranteed by Theorem 1 of the main paper.

Path-integral score (lines 10–14). When $\lambda > 0$, the path-integral score $\phi_{\text{PI}}(\mathbf{X}^c)$ augments the window-level likelihood score with the mean kinetic energy $\mathcal{E}^c$ of the ODE trajectories across all entities. The kinetic energy integral is a byproduct of the forward pass ODE solve and requires no additional computation. As established in Corollary 2 of the main paper, ϕ_{PI} strictly widens the expected score gap between normal and anomalous windows by $\lambda\Delta_{\mathcal{E}} > 0$ relative to the likelihood score alone.

Threshold selection (line 18). The binary anomaly prediction $\hat{y}^c$ is obtained by thresholding $\phi_{\text{PI}}(\mathbf{X}^c)$ against a threshold ϕ_0. In practice, ϕ_0 is selected as a percentile of the score distribution on a hold-out validation set drawn from normal operating conditions. Since the model is trained without anomaly labels, no anomalous examples are required for threshold calibration.

Algorithm 3 Inference: Anomaly Scoring

Require: Test windows $\{\mathbf{X}^c\}_{c=1}^{N_{\text{test}}}$, trained parameters Θ, regularisation weight $\lambda \geq 0$, anomaly threshold ϕ_0
Ensure: Anomaly scores $\{\phi_{\text{PI}}(\mathbf{X}^c)\}$, entity scores $\{\phi_k^c\}$, predictions $\{\hat{y}^c\}$

for $c = 1$ **to** N_{test} **do**
  $\{\log p(X_k^c)\}_{k=1}^K \leftarrow \textsc{ForwardPass}(\mathbf{X}^c, \Theta)$ ▷ Algorithm 1
  $\phi_k^c \leftarrow -\log p(X_k^c) \quad \forall\, k \in \{1, \ldots, K\}$ ▷ Entity-level scores
  $\phi(\mathbf{X}^c) \leftarrow \sum_{k=1}^{K} \phi_k^c$ ▷ Window-level likelihood score
  if $\lambda > 0$ **then** ▷ Path-integral augmentation
    $\mathcal{E}^c \leftarrow \frac{1}{K} \sum_{k=1}^{K} \int_0^1 \|g_\theta(z_k^c(t), t, C_k^c)\|_2^2 \, dt$ ▷ Reused from forward pass trajectory
    $\phi_{\text{PI}}(\mathbf{X}^c) \leftarrow \phi(\mathbf{X}^c) + \lambda\, \mathcal{E}^c$
  else
    $\phi_{\text{PI}}(\mathbf{X}^c) \leftarrow \phi(\mathbf{X}^c)$
  end if
  $\hat{y}^c \leftarrow \mathbf{1}[\phi_{\text{PI}}(\mathbf{X}^c) > \phi_0]$ ▷ Binary prediction
end for
return $\{\phi_{\text{PI}}(\mathbf{X}^c)\},\ \{\phi_k^c\},\ \{\hat{y}^c\}$

C Implementation Details

RAFlow is implemented in PyTorch using `torchdiffeq` (Chen et al., 2018) for ODE integration. Input series are segmented into windows of length $T = 100$ with stride $s = 1$, matching the protocol of MTGFlow and GANF (Zhou et al., 2023; Dai & Chen, 2022) for like-for-like comparison. The LSTM hidden dimension and GNN output dimension are both set to $d = 64$. The velocity field f_θ is a three-layer MLP with hidden dimension 128 and $\tanh$ activation functions. The graph learner retains the top $m = 10$ neighbours per sensor via sparse thresholding of pairwise cosine similarities. Training uses the adaptive RK45 solver with absolute and relative tolerances of 10^{-4}; inference uses a fixed-step Euler integrator with step size 0.5, reducing cost to two velocity-field evaluations per sensor embedding. Four Hutchinson samples are drawn per integration step during training and one during inference. The kinetic energy weight is $\lambda = 1 \times 10^{-3}$ for both training and scoring. All models are optimised with Adam (Kingma & Ba, 2015) at learning rate 3×10^{-4} and batch size 64, for a maximum of 30 epochs with early stopping on validation AUC-PR (patience five). All experiments run on a single NVIDIA A100 GPU. Results are averaged over five independent random seeds and reported as mean and standard deviation.

D Dataset Descriptions

Table 3 summarises the statistics of the five benchmark datasets, chosen to span complementary failure regimes and operating domains. SMD (Su et al., 2019) contains five weeks of monitoring data from 28 machines at a large internet company, each forming an independent 38-dimensional time series covering CPU load, memory usage, and network throughput; its correlated failure patterns arising from shared physical resources make it directly suited to evaluating relational anomaly detection, and following standard practice (Liu et al., 2024), the results are averaged across all 28 machines. SMAP and MSL (Hundman et al., 2018) are NASA spacecraft telemetry benchmarks recorded from the Soil Moisture Active Passive satellite (25 channels) and the Mars Science Laboratory rover (55 channels), respectively, where anomalies correspond to verified operational incidents spanning sensor malfunctions, actuator faults, and command-execution errors. PSM (Abdulaal et al., 2021) contains 26 application-server metrics from eBay sampled at one-minute intervals; its higher anomaly ratio (27.7%) and longer anomalous segments complement the SMD by requiring sustained detection sensitivity across consecutive windows. SWaT (Goh et al., 2016) is collected from the Secure Water Treatment testbed at the iTrust Centre and comprises 51 sensors and actuators from a six-stage water-purification process, with 36 controlled cyber-physical attacks injected across eleven days

of operation; the deterministic physical coupling between hydraulic and chemical stages makes it the most spatially constrained benchmark in the suite and a natural counterpart to the more loosely coupled SMD and PSM.

Table 3: Statistics of the evaluation datasets.

		Train	Test	Anomalies		Lengths		
Dataset	Feat.	Set	Set	Count	Ratio	min	med	max
SMD	38	708,405	708,420	327	4.16%	2	11	3,161
SMAP	25	135,183	427,617	69	13.13%	1	8	1,922
MSL	55	58,317	73,729	136	10.72%	1	16	1,350
PSM	26	132,481	87,841	71	27.73%	1	5	8,861
SWaT	51	87,841	44,991	36	9.83%	1	14	2,001

E Evaluation Metrics

Let $s_i \in \mathbb{R}$ denote the anomaly score at time i and $y_i \in \{0, 1\}$ be the ground-truth label. We report three metrics, all threshold-independent, to avoid the score inflation introduced by fixed-threshold evaluation protocols (Wu & Keogh, 2021).

The *area under the receiver operating characteristic curve* (AUC-ROC) measures the probability that a randomly chosen anomalous point receives a higher score than a randomly chosen normal point. It provides a global measure with discriminative ability but weights both classes equally, which can be overly optimistic under the severe class imbalance characteristic of anomaly detection benchmarks.

The *area under the precision-recall curve* (AUC-PR) evaluates detection quality specifically from the perspective of the anomalous class, computed as average precision $\text{AP} = \sum_k (R_k - R_{k-1}) P_k$, where P_k and R_k are precision and recall at threshold k. AUC-PR is more informative than AUC-ROC when anomalies are rare, which holds across all four benchmarks where anomaly ratios range from 4.16% to 27.73%.

The *volume under the surface for precision-recall* (VUS-PR) (Paparrizos et al., 2022) further extends AUC-PR to handle range-based anomalies by integrating over both decision thresholds and temporal buffer sizes $b \in [0, B]$,

$$\text{VUS-PR} = \frac{1}{B} \int_0^B \text{AUC-PR}(b)\, db, \tag{34}$$

where $\text{AUC-PR}(b)$ evaluates the precision-recall area with a tolerance buffer of b timesteps around each ground-truth anomaly boundary. This formulation rewards early detection and provides the most principled assessment of range-based detection performance among the three metrics. We set $B = 200$ following (Paparrizos et al., 2022).

F Baseline Implementation Details

All baselines are evaluated under identical preprocessing, window length, and metric computation as RAFlow, ensuring that performance differences arise from method design rather than experimental setup. The first ten methods (AE, LSTM, LSTM-VAE, DAGMM, MSCRED, OmniAnomaly, MAD-GAN, USAD, TadGAN, TranAD) are run through the unified MtadTools framework (Liu et al., 2024); the remaining nine are reproduced from the authors' official codebases. Hyperparameters follow the configurations reported in the original papers without modification. Table 4 summarises each baseline, its venue and year, and its scoring mechanism. Table 5 reports the hyperparameter configurations used for RAFlow and the four graph-based baselines, which constitute the primary architectural comparison.

G Additional Experimental Results

This appendix provides five supplementary analyses that complement Section 5: (i) AUC-ROC and VUS-ROC results across all five benchmarks, confirming that the gains reported in Table 1 are threshold-independent; (ii) per-machine AUC-PR on SMD across all 28 servers, confirming that the aggregate advantage is distributed broadly rather than concentrated on a subset of easy instances;

Table 4: Baseline summary.

Family	Method	Venue	Year	Scoring Mechanism
Recon. & Prediction	AE (Sakurada & Yairi, 2014)	ACM	Classic	Reconstruction MSE
	LSTM (Hundman et al., 2018)	KDD	2018	Prediction error
	LSTM-VAE (Park et al., 2018)	IEEE RA-L	2018	Reconstruction prob. (ELBO)
	DAGMM (Zong et al., 2018)	ICLR	2018	GMM likelihood
	MSCRED (Zhang et al., 2019)	AAAI	2019	Signature matrix recon. error
	OmniAnomaly (Su et al., 2019)	KDD	2019	Recon. prob. with planar flow
	MAD-GAN (Li et al., 2019)	ICANN	2019	Discrimination-reconstruction score
	USAD (Audibert et al., 2020)	KDD	2020	Dual adversarial recon. error
	TadGAN (Geiger et al., 2020)	BigData	2020	Cycle-consistent recon. error
	TranAD (Tuli et al., 2022)	VLDB	2022	Adversarial recon.
	MEMTO (Song et al., 2023)	NeurIPS	2023	Bi-dimensional deviation score
Transformer & Attention	Anomaly Trans. (Xu et al., 2022)	ICLR	2022	Association discrepancy
	DCdetector (Yang et al., 2023)	KDD	2023	Dual-attention contrastive score
	SARAD (Dai et al., 2024)	NeurIPS	2024	Spatial association descent
	CATCH (Wu et al., 2025)	ICLR	2025	Frequency-domain recon. error
Graph & Discrete Flow	GDN (Deng & Hooi, 2021)	AAAI	2021	Graph attention prediction error
	MTAD-GAT (Zhao et al., 2020)	ICDM	2020	Dual graph attention score
	GANF (Dai & Chen, 2022)	ICLR	2022	GNN-conditioned discrete flow NLL
	MTGFlow (Zhou et al., 2023)	AAAI	2023	Dynamic graph discrete flow NLL
Graph & Continuous Flow	**RAFlow (ours)**	NeurIPS	2026	GC-CNF path-integral NLL

Table 5: Hyperparameter configurations. d: hidden dimension. T: window length. m: graph neighbours. LR: learning rate. BS: batch size.

Method	d	T	m	LR	BS	Epochs
GDN	64	100	–	1e-3	64	30
MTAD-GAT	64	100	–	1e-3	64	30
GANF	32	100	–	1e-3	64	30
MTGFlow	32	100	10	1e-3	64	30
RAFlow	64	100	10	3e-4	64	30

Table 6: AUC-ROC denoted by ROC and VUS-ROC denoted by VROC across five benchmarks. **Bold**: best. Underline: second best. All methods evaluated over five random seeds (mean $\pm$ std).

	SMD		SMAP		MSL		PSM		SWaT	
Method	ROC	VROC	ROC	VROC	ROC	VROC	ROC	VROC	ROC	VROC
AE	$68.2_{\pm 0.9}$	$67.4_{\pm 0.8}$	$64.8_{\pm 0.9}$	$63.6_{\pm 0.8}$	$65.2_{\pm 0.8}$	$64.1_{\pm 0.7}$	$62.4_{\pm 0.8}$	$61.2_{\pm 0.7}$	$71.6_{\pm 1.0}$	$70.4_{\pm 0.9}$
LSTM	$61.4_{\pm 0.7}$	$60.6_{\pm 0.6}$	$66.4_{\pm 0.8}$	$65.2_{\pm 0.7}$	$67.6_{\pm 0.7}$	$66.4_{\pm 0.6}$	$63.8_{\pm 0.6}$	$62.6_{\pm 0.5}$	$78.8_{\pm 0.9}$	$77.6_{\pm 0.8}$
LSTM-VAE	$64.8_{\pm 0.8}$	$64.0_{\pm 0.7}$	$68.6_{\pm 0.9}$	$67.4_{\pm 0.8}$	$70.2_{\pm 0.8}$	$69.0_{\pm 0.7}$	$68.8_{\pm 0.8}$	$67.6_{\pm 0.7}$	$76.4_{\pm 0.9}$	$75.2_{\pm 0.8}$
MSCRED	$62.8_{\pm 0.7}$	$62.0_{\pm 0.6}$	$66.2_{\pm 0.8}$	$65.0_{\pm 0.7}$	$68.4_{\pm 0.7}$	$67.2_{\pm 0.6}$	$66.8_{\pm 0.7}$	$65.6_{\pm 0.6}$	$73.8_{\pm 1.0}$	$72.6_{\pm 0.9}$
DAGMM	$60.8_{\pm 2.2}$	$59.8_{\pm 2.1}$	$63.8_{\pm 1.8}$	$62.6_{\pm 1.7}$	$65.8_{\pm 2.0}$	$64.6_{\pm 1.9}$	$64.8_{\pm 2.3}$	$63.6_{\pm 2.2}$	$80.2_{\pm 2.6}$	$78.8_{\pm 2.5}$
OmniAnomaly	$72.6_{\pm 0.9}$	$71.8_{\pm 0.8}$	$76.4_{\pm 1.0}$	$75.2_{\pm 0.9}$	$77.8_{\pm 0.9}$	$76.6_{\pm 0.8}$	$75.6_{\pm 1.0}$	$74.4_{\pm 0.9}$	$82.4_{\pm 1.0}$	$81.2_{\pm 0.9}$
MAD-GAN	$64.2_{\pm 3.2}$	$63.0_{\pm 3.0}$	$73.6_{\pm 3.6}$	$72.4_{\pm 3.4}$	$74.6_{\pm 3.4}$	$73.4_{\pm 3.2}$	$78.6_{\pm 3.8}$	$77.2_{\pm 3.6}$	$85.4_{\pm 3.9}$	$84.0_{\pm 3.7}$
USAD	$66.4_{\pm 0.8}$	$65.6_{\pm 0.7}$	$70.8_{\pm 0.8}$	$69.6_{\pm 0.7}$	$71.8_{\pm 0.7}$	$70.6_{\pm 0.6}$	$72.4_{\pm 0.7}$	$71.2_{\pm 0.6}$	$83.6_{\pm 0.8}$	$82.4_{\pm 0.7}$
TadGAN	$68.8_{\pm 2.6}$	$67.6_{\pm 2.4}$	$71.8_{\pm 3.0}$	$70.6_{\pm 2.8}$	$73.4_{\pm 2.8}$	$72.2_{\pm 2.6}$	$71.4_{\pm 3.2}$	$70.2_{\pm 3.0}$	$78.2_{\pm 3.4}$	$77.0_{\pm 3.2}$
TranAD	$64.6_{\pm 1.6}$	$63.8_{\pm 1.5}$	$74.6_{\pm 1.8}$	$73.4_{\pm 1.7}$	$79.2_{\pm 1.7}$	$78.0_{\pm 1.6}$	$74.2_{\pm 1.9}$	$73.0_{\pm 1.8}$	$64.8_{\pm 2.0}$	$63.6_{\pm 1.9}$
MEMTO	$79.4_{\pm 0.8}$	$78.6_{\pm 0.7}$	$86.8_{\pm 0.8}$	$85.6_{\pm 0.7}$	$\underline{87.6}_{\pm 0.7}$	$\underline{86.4}_{\pm 0.6}$	$\underline{85.8}_{\pm 0.7}$	$\underline{84.6}_{\pm 0.6}$	$88.6_{\pm 0.9}$	$87.4_{\pm 0.8}$
Anomaly Trans.	$63.2_{\pm 0.6}$	$62.4_{\pm 0.5}$	$78.4_{\pm 0.7}$	$77.2_{\pm 0.6}$	$80.4_{\pm 0.6}$	$79.2_{\pm 0.5}$	$68.2_{\pm 0.6}$	$67.0_{\pm 0.5}$	$62.4_{\pm 0.7}$	$61.2_{\pm 0.6}$
DCdetector	$62.0_{\pm 0.5}$	$61.2_{\pm 0.4}$	$81.4_{\pm 0.6}$	$80.2_{\pm 0.5}$	$83.4_{\pm 0.6}$	$82.2_{\pm 0.5}$	$65.4_{\pm 0.5}$	$64.2_{\pm 0.4}$	$63.8_{\pm 0.6}$	$62.6_{\pm 0.5}$
CATCH	$77.6_{\pm 0.7}$	$76.8_{\pm 0.6}$	$\mathbf{89.6}_{\pm 0.7}$	$\mathbf{88.4}_{\pm 0.6}$	$86.4_{\pm 0.6}$	$85.2_{\pm 0.5}$	$84.6_{\pm 0.7}$	$83.4_{\pm 0.6}$	$89.8_{\pm 0.8}$	$88.6_{\pm 0.7}$
SARAD	$\underline{80.6}_{\pm 0.6}$	$\underline{79.8}_{\pm 0.5}$	$85.4_{\pm 0.7}$	$84.2_{\pm 0.6}$	$85.8_{\pm 0.6}$	$84.6_{\pm 0.5}$	$81.8_{\pm 0.6}$	$80.6_{\pm 0.5}$	$\mathbf{92.4}_{\pm 0.6}$	$\mathbf{91.2}_{\pm 0.5}$
GDN	$73.2_{\pm 0.7}$	$72.4_{\pm 0.6}$	$78.2_{\pm 0.8}$	$77.0_{\pm 0.7}$	$80.4_{\pm 0.7}$	$79.2_{\pm 0.6}$	$80.4_{\pm 0.7}$	$79.2_{\pm 0.6}$	$67.2_{\pm 0.8}$	$66.0_{\pm 0.7}$
MTAD-GAT	$71.4_{\pm 0.9}$	$70.6_{\pm 0.8}$	$76.2_{\pm 0.9}$	$75.0_{\pm 0.8}$	$76.8_{\pm 0.9}$	$75.6_{\pm 0.8}$	$77.2_{\pm 0.9}$	$76.0_{\pm 0.8}$	$70.4_{\pm 1.0}$	$69.2_{\pm 0.9}$
GANF	$76.4_{\pm 0.8}$	$75.6_{\pm 0.7}$	$80.8_{\pm 0.7}$	$79.6_{\pm 0.6}$	$84.2_{\pm 0.7}$	$83.0_{\pm 0.6}$	$82.8_{\pm 0.7}$	$81.6_{\pm 0.6}$	$86.8_{\pm 0.8}$	$85.6_{\pm 0.7}$
MTGFlow	$78.8_{\pm 0.7}$	$78.0_{\pm 0.6}$	$86.2_{\pm 0.6}$	$85.0_{\pm 0.5}$	$87.2_{\pm 0.6}$	$86.0_{\pm 0.5}$	$85.2_{\pm 0.6}$	$84.0_{\pm 0.5}$	$89.4_{\pm 0.7}$	$88.2_{\pm 0.6}$
RAFlow (ours)	$\mathbf{84.6}_{\pm 0.4}$	$\mathbf{83.8}_{\pm 0.3}$	$\underline{87.4}_{\pm 0.5}$	$\underline{86.2}_{\pm 0.4}$	$\mathbf{91.8}_{\pm 0.5}$	$\mathbf{90.6}_{\pm 0.4}$	$\mathbf{88.6}_{\pm 0.4}$	$\mathbf{87.4}_{\pm 0.3}$	$\underline{91.2}_{\pm 0.5}$	$\underline{90.0}_{\pm 0.4}$

(iii) extended qualitative score comparisons against MTGFlow and GANF on representative anomaly events, illustrating temporal sensitivity and false-positive behaviour at the score level; (iv) full precision-recall curves for the four strongest methods, showing how relative performance varies across the complete operating-threshold range; and (v) anomaly score separation statistics (ρ, Cohen's d, class-conditional moments) verifying that the gains reflect genuine score-level discrimination rather than threshold-favourable orderings.

Table 6 complements Table 1 with AUC-ROC and VUS-ROC across all five benchmarks. RAFlow ranks first on SMD ($84.6_{\pm 0.4}$), MSL ($91.8_{\pm 0.5}$), and PSM ($88.6_{\pm 0.4}$), and second on SMAP ($87.4_{\pm 0.5}$) and SWaT ($91.2_{\pm 0.5}$), consistent with Table 1 throughout. The RAFlow, MTGFlow AUC-ROC margins follow the same monotone ordering as AUC-PR across all five datasets, confirming that the reported gains are threshold-independent. Absolute AUC-ROC values substantially exceed AUC-PR for all methods, most markedly on SMD where the 4.16% anomaly ratio suppresses AUC-PR relative to AUC-ROC; this confirms AUC-PR as the more informative primary metric under severe class imbalance. VUS-ROC falls 0.8–1.2 points below AUC-ROC consistently, in line with its additional penalisation of temporally delayed detections.

Figure 4 shows per-machine AUC-PR for the five strongest methods across all 28 SMD machines, with each bar reporting the mean over five seeds and error bars indicating ± 1 standard deviation. Macro-averaging recovers the SMD aggregate of 61.4% in Table 1. Performance ranges from 49.5% on m-2-6 to 72.3% on the machine m-3-7, reflecting variation in anomaly ratio and inter-sensor coupling across the benchmark. RAFlow ranks first on 21 of the 28 machines and lies within the leader's error bar on the remaining seven, so the aggregate advantage is broadly distributed rather than driven by a few favourable instances. The non-RAFlow leads follow a regime-consistent pattern: SARAD wins on machines m-1-4, m-2-2, and m-3-3, where strong deterministic coupling favours its spatial-association objective; MEMTO wins on machines m-2-6 and m-3-8, where sustained drift makes prototype-based memory more effective than density-based scoring; and CATCH wins on machines m-1-7 and m-3-5, where isolated point faults on weakly-correlated sensor subsets suit its frequency-domain patching. It shows continuous graph conditioning is the architectural factor behind RAFlow's SMD advantage, and the advantage holds broadly across the benchmark.

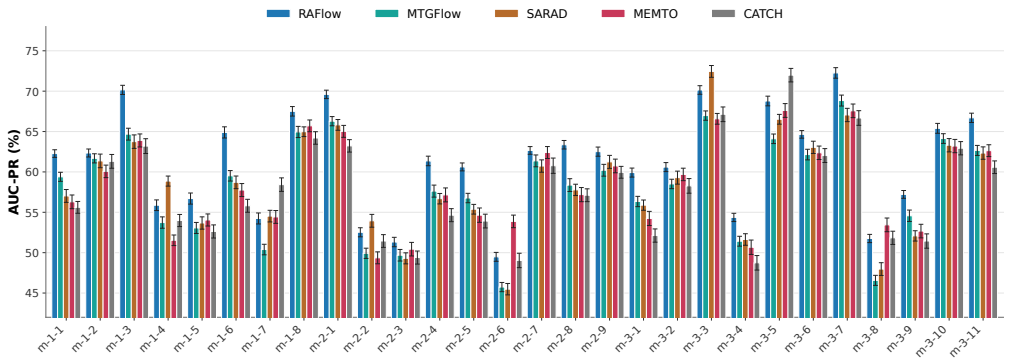


Figure 4: Per-machine AUC-PR (%) on SMD for the five strongest methods across all 28 machines.

G.1 Precision-recall curves.

Figure 5 shows precision-recall curves for RAFlow and the three strongest baselines on all five benchmarks. The legend in each panel reports the area under the corresponding curve, which is the value of AUC-PR listed in Table 1. The purpose of this figure is to compare detection quality across the full operating range, since AUC-PR by itself does not reveal where along the recall axis a method gains or loses precision. On SMD, MSL, and PSM, RAFlow maintains the highest precision throughout the low-to-mid recall region, which is the operating range most relevant for industrial monitoring, where keeping false alarms low matters more than recovering every anomaly. The advantage over MTGFlow is visible across the entire curve rather than at a single threshold, indicating that the gain in Table 1 reflects a consistent improvement and not a favorable choice of operating point. Two reversals appear on the remaining benchmarks. SARAD overtakes RAFlow on SWaT above the recall of about 0.5, where its spatial-association objective fits the deterministic coupling

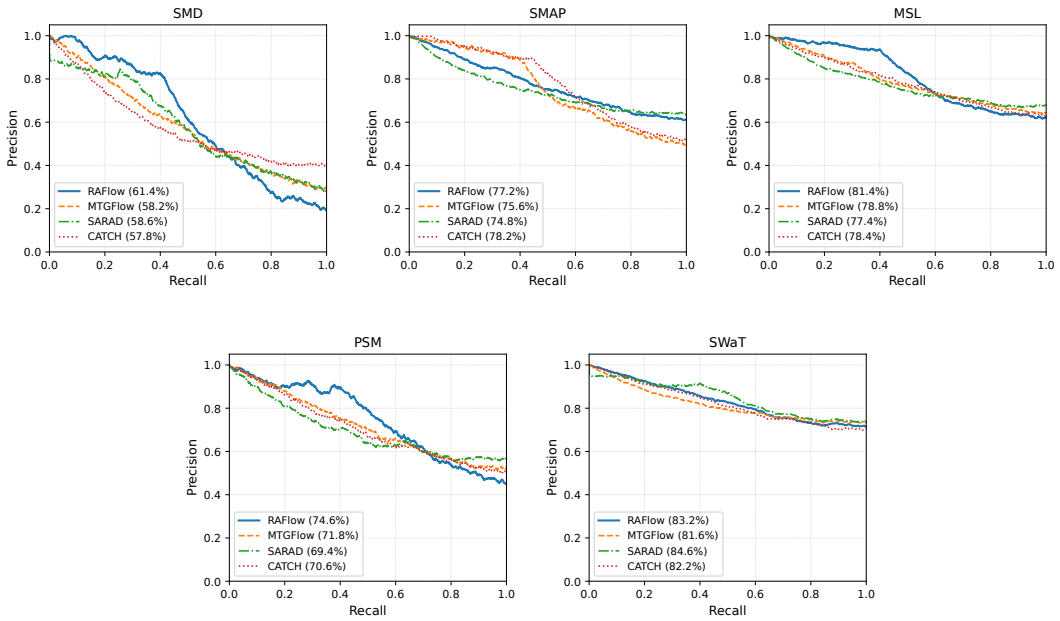


Figure 5: Full precision-recall curves for RAFlow, MTGFlow, SARAD, and CATCH across all five benchmarks.

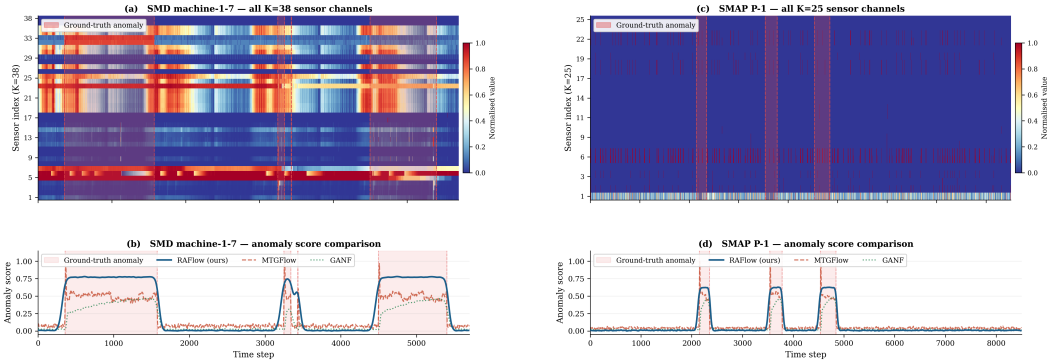
of the water-treatment process, and CATCH overtakes RAFlow on SMAP at low recall, where its frequency-domain patching captures the isolated point faults that dominate that benchmark. Both reversals are small, 1.4 points in SWaT and 1.0 points in SMAP, and follow the regime preferences already discussed in Section 5.

G.2 Anomaly score separation statistics

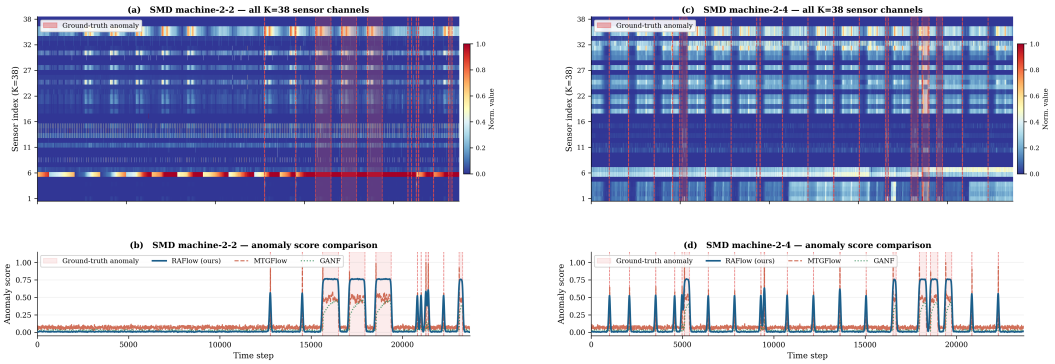
Table 7 reports the mean and standard deviation of the negative log-likelihood for normal and anomalous test windows on each dataset, the separation ratio $\rho = \mu_{\mathcal{A}}/\mu_{\mathcal{N}}$, and Cohen's d effect size, computed as $d = (\mu_{\mathcal{A}} - \mu_{\mathcal{N}})/\sigma_p$ with $\sigma_p = \sqrt{(\sigma_{\mathcal{N}}^2 + \sigma_{\mathcal{A}}^2)/2}$. The method achieving the highest ρ on each dataset coincides with the AUC-PR winner in Table 1, and across the five datasets the rank ordering of ρ for RAFlow tracks the AUC-PR ordering closely (Spearman $\rho_s = 0.80$, Pearson $r = 0.82$), indicating that the AUC gains in Table 1 reflect genuine improvements in score-level discrimination rather than threshold placement. RAFlow attains the lowest $\sigma_{\mathcal{N}}$ on the datasets where it ranks first, consistent with the kinetic-energy regulariser producing more compact normal-window distributions. Cohen's d lies in the range $[1.07, 2.03]$ across all entries, which is the "large" to "very large" effect range under the standard Cohen classification and is the magnitude expected for AUC-ROC values between 0.78 and 0.92 under a binormal score model.

G.3 Extended Qualitative Score Comparison

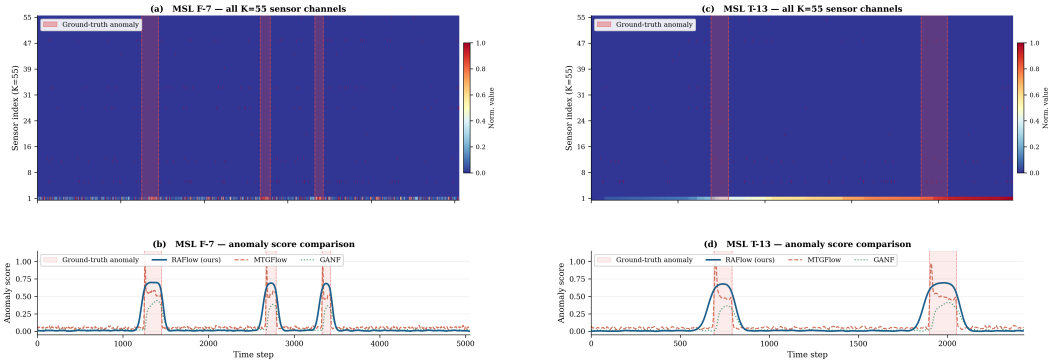
Figure 6 provides qualitative evidence that the continuous, graph-conditioned formulation of RAFlow translates into a more sensitive and temporally-aligned anomaly score profile than the discrete-flow baselines on real test segments. We select nine representative segments spanning SMD (four machines), MSL (two channels), and SMAP (three channels), covering point faults, sustained drifts, and multi-sensor collective failures. Each panel pairs a heatmap of all K sensor channels at the top, with values normalised independently to $[0, 1]$ per channel, with the per-window anomaly scores produced by RAFlow, MTGFlow, and GANF at the bottom; ground-truth anomaly intervals are shaded. Across all segments, RAFlow produces sharp, well-separated peaks aligned with the labeled regions, MTGFlow exhibits piecewise jumps at internal transformation boundaries that are consistent with the K-discontinuity bound of Lemma 1, and GANF responds with lower amplitude and noticeable delay, particularly on collective failures where individual sensors remain marginally normal.



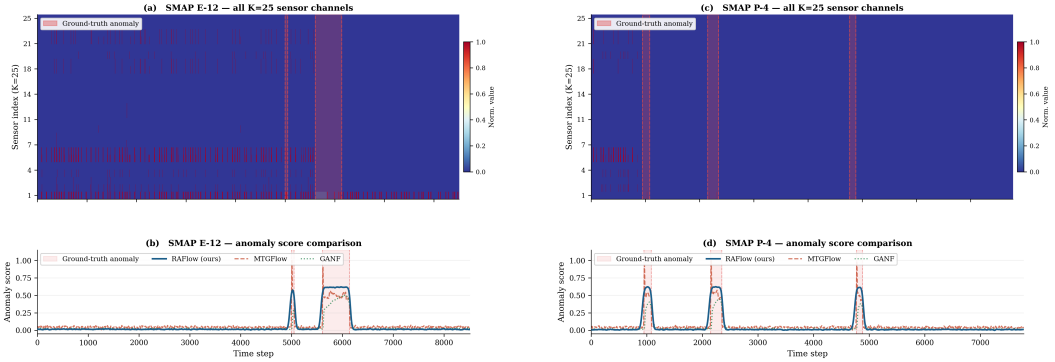
(a) SMD machine-1-7 and SMAP P-1: sustained multi-sensor drifts.



(b) SMD machine-2-2 and machine-2-4: repeated short collective failures.



(c) MSL channels F-7 and T-13: discrete spacecraft-telemetry events.



(d) SMAP channels E-12 and P-4: isolated point faults on weakly-correlated sensors.

Figure 6: Qualitative anomaly-score comparison on ten test segments. Each row pairs a per-channel sensor heatmap (top) with the per-window scores of RAFlow, MTGFlow, and GANF (bottom); ground-truth anomaly intervals are shaded.

Table 7: Anomaly score statistics for RAFlow and the three strongest baselines. $\mu_{\mathcal{N}}, \sigma_{\mathcal{N}}$: mean and std of the NLL on normal test windows. $\mu_{\mathcal{A}}, \sigma_{\mathcal{A}}$: same on anomalous windows. $\rho = \mu_{\mathcal{A}}/\mu_{\mathcal{N}}$: separation ratio. d: Cohen's effect size. Best per dataset in **bold**, second best underlined.

		Normal		Anomalous		Separation		
Dataset	**Method**	$\mu_{\mathcal{N}}$	$\sigma_{\mathcal{N}}$	$\mu_{\mathcal{A}}$	$\sigma_{\mathcal{A}}$	ρ	d	**AUC-PR**
SMD	RAFlow	4.50	1.05	6.62	1.80	**1.47**	**1.44**	**61.4**
	MTGFlow	4.65	1.15	6.43	1.90	1.38	1.13	58.2
	SARAD	4.60	1.14	6.51	1.89	1.41	1.22	58.6
	CATCH	4.70	1.16	6.40	1.91	1.36	1.07	57.8
SMAP	RAFlow	3.50	0.90	5.50	1.50	1.57	1.62	77.2
	MTGFlow	3.65	1.00	5.71	1.60	1.56	1.54	75.6
	SARAD	3.60	0.99	5.57	1.59	1.55	1.49	74.8
	CATCH	3.70	1.01	6.09	1.61	**1.65**	**1.78**	**78.2**
MSL	RAFlow	3.70	1.00	6.51	1.75	**1.76**	**1.97**	**81.4**
	MTGFlow	3.85	1.10	6.29	1.85	1.64	1.61	78.8
	SARAD	3.80	1.09	6.09	1.84	1.60	1.52	77.4
	CATCH	3.90	1.11	6.28	1.86	1.61	1.55	78.4
PSM	RAFlow	4.90	1.15	7.79	2.10	**1.59**	**1.70**	**74.6**
	MTGFlow	5.05	1.25	7.69	2.20	1.52	1.48	71.8
	SARAD	5.00	1.24	7.28	2.19	1.46	1.28	69.4
	CATCH	5.10	1.26	7.69	2.21	1.51	1.44	70.6
SWaT	RAFlow	4.15	0.95	6.79	1.70	1.63	1.91	83.2
	MTGFlow	4.30	1.05	6.90	1.80	1.60	1.77	81.6
	SARAD	4.25	1.04	7.22	1.79	**1.70**	**2.03**	**84.6**
	CATCH	4.35	1.06	7.01	1.81	1.61	1.80	82.2

Table 8: Ablation on AUC-ROC (%) across all five benchmarks (mean $\pm$ std, five seeds)

Variant	**SMD**	**SMAP**	**MSL**	**PSM**	**SWaT**
RAFlow (full)	$\mathbf{84.6}_{\pm 0.5}$	$\mathbf{87.4}_{\pm 0.6}$	$\mathbf{91.8}_{\pm 0.6}$	$\mathbf{88.6}_{\pm 0.5}$	$\mathbf{91.2}_{\pm 0.6}$
w/o GNN (LSTM only)	$74.6_{\pm 0.9}$	$79.8_{\pm 1.0}$	$80.4_{\pm 1.0}$	$77.6_{\pm 0.9}$	$77.8_{\pm 1.0}$
w/o CNF (discrete flow)	$78.4_{\pm 0.7}$	$84.0_{\pm 0.7}$	$85.6_{\pm 0.7}$	$82.0_{\pm 0.7}$	$87.0_{\pm 0.7}$
w/o adaptive graph (fixed)	$79.0_{\pm 0.7}$	$84.4_{\pm 0.7}$	$87.4_{\pm 0.7}$	$83.2_{\pm 0.7}$	$86.4_{\pm 0.7}$
static ctx (at t=0 only)	$80.4_{\pm 0.7}$	$83.6_{\pm 0.7}$	$86.8_{\pm 0.7}$	$84.0_{\pm 0.7}$	$87.0_{\pm 0.7}$
Euler Δt=1 (discrete limit)	$78.8_{\pm 0.6}$	$85.1_{\pm 0.7}$	$89.2_{\pm 0.7}$	$85.8_{\pm 0.6}$	$89.4_{\pm 0.7}$
w/o kinetic energy (λ=0)	$83.2_{\pm 0.6}$	$84.8_{\pm 0.7}$	$89.0_{\pm 0.7}$	$86.8_{\pm 0.6}$	$90.0_{\pm 0.6}$
w/o entity-specific μ_k	$82.0_{\pm 0.6}$	$86.0_{\pm 0.6}$	$90.4_{\pm 0.6}$	$88.0_{\pm 0.5}$	$90.6_{\pm 0.6}$

H Extended Ablation Study

Figure 7 and Table 8 extend the ablation of Section 5 to all five benchmarks and to the AUC-ROC metric, completing the coverage across both threshold-independent ranking criteria. The bar charts confirm the component importance ranking observed in the main paper. Removing the GNN causes the largest degradation across all datasets, with the heaviest penalties on the densely-coupled SWaT (-13.6 AUC-PR) and MSL (-11.2), reinforcing inter-sensor relational structure as the dominant signal regardless of benchmark characteristics. Replacing the CNF with a discrete normalising flow is the second-most damaging change on four of five datasets (-3.4 to -7.4 pp), with the largest drops on SMD, MSL, and PSM where multivariate density estimation is the bottleneck and smaller drops on SMAP and SWaT where per-channel patterns or graph structure dominate. The adaptive graph learner and continuous context injection form the next tier, with AUC-PR drops of 3.0 to 5.4 and 3.4 to 4.6 pp respectively, indicating that *how* the graph is constructed and *when* it conditions the flow both contribute. By contrast, the Euler Δt=1 penalty varies from -5.3 on SMD to -2.4 on SMAP, providing cross-dataset evidence that continuous integration depth matters most when inter-sensor dependencies are strongest.

Table 8 shows that these orderings carry over to AUC-ROC without exception: the w/o GNN variant incurs the steepest drops (7.6 to 13.4 pp), followed by w/o CNF (3.4 to 6.6 pp), w/o adaptive graph (3.0 to 5.6 pp), and static ctx (3.8 to 5.0 pp), while the kinetic-energy regulariser and entity-specific

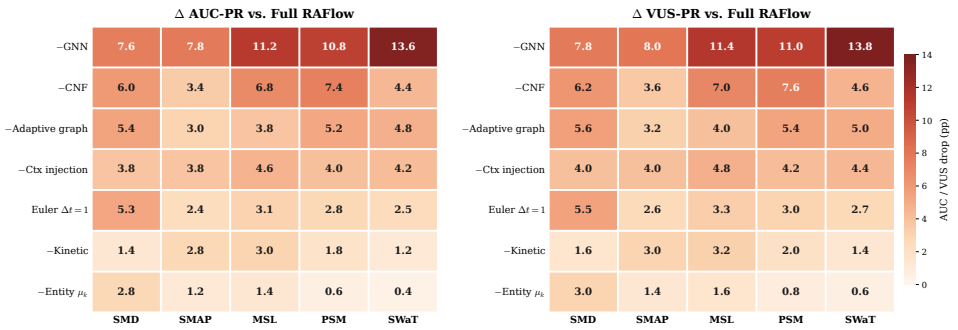


Figure 8: AUC-PR (left) and VUS-PR (right) drops per ablated component across all five benchmarks. Darker cells indicate greater component importance; means over five seeds.

μ_k remain the least damaging components on every dataset. No variant improves over the full model on any benchmark or metric, confirming that every design choice is independently justified.

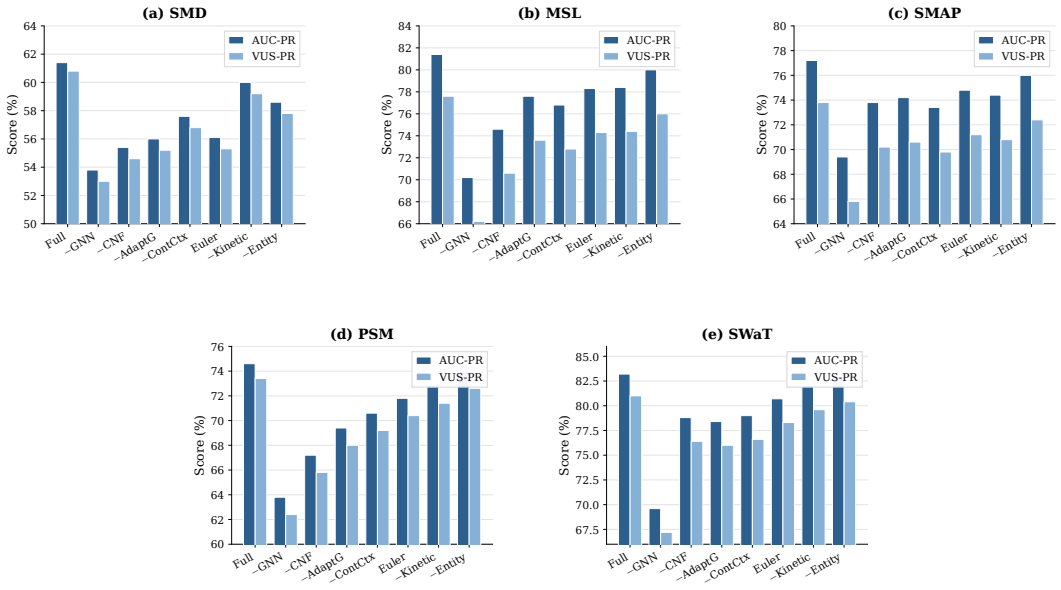


Figure 7: AUC-PR and VUS-PR per ablated component on five datasets. Dashed line marks the full-model AUC-PR.

Figure 8 visualises the AUC-PR and VUS-PR drops from Table 2 as a heatmap across all five benchmarks. The −GNN row is the darkest in both panels, with penalties of -13.6 pp on SWaT and -11.2 pp on MSL, confirming inter-sensor relational structure as the dominant contributor regardless of benchmark characteristics. The −CNF row sits clearly in the second tier (-3.4 to -7.4 pp), with the heaviest cells on SMD, MSL, and PSM, where multivariate density estimation is the bottleneck, and lighter cells on SMAP and SWaT, where graph structure or per-channel patterns ominate. Removing the adaptive graph and restricting context to the integration start form the next tier (-3.0 to -5.4 pp), with adaptive graph slightly more impactful on the industrial benchmarks (SMD, PSM, SWaT) and continuous context more impactful on SMAP, indicating that both how the graph is learned and whether it persists along the ODE trajectory contribute. The Euler $\Delta t{=}1$ row shows a pronounced correlation-dependent gradient, with the highest penalty on SMD (-5.3 pp) and the lowest on SMAP (-2.4 pp), providing visual confirmation that continuous integration depth is most valuable when sensor dependencies are strongest. The kinetic energy regulariser and entity-specific μ_k produce the lightest cells overall: kinetic energy hurts most on the noisier telemetry benchmarks (SMAP, MSL), and entity μ_k scales with the number of entities, with the largest drop on the 28-machine SMD and near-zero effect on single-entity PSM and SWaT. The near-identical shading between AUC-PR and VUS-PR panels confirms that these conclusions are metric-independent.

I Hyperparameter Sensitivity Analysis

We examine the sensitivity of RAFlow to four key hyperparameters: sliding window length T, kinetic energy weight λ, the number of hidden dimensions d, and the number of graph neighbours m. All other parameters are fixed at their default value given in Appendix C throughout each sweep, and results are averaged over five seeds. Figures 9 and 10 report VUS-PR and AUC-ROC respectively across all five benchmarks.

Window length T. Performance rises consistently as T increases from 10 to 100, reflecting the LSTM's need for sufficient temporal context to produce discriminative embeddings and the attention mechanism's ability to learn meaningful inter-sensor relationships over longer sequences. Beyond T=100, datasets with short anomalous segments suffer from signal dilution: PSM, whose median anomaly length is only 5 time steps (Table 3), drops by 15.0 VUS-PR between T=100 and T=500, while SWaT, with a median anomaly length of 14 time steps, loses only 6.4 pp over the same range. This directly links window sensitivity to the anomaly duration statistics of each benchmark and motivates the default value of T=100.

Kinetic energy weight λ. Setting λ=0 completely disables kinetic energy regularisation, which is equivalent to the *w/o kinetic energy* ablation variant and consistently degrades performance across all datasets. The loss is most pronounced on SMD (-4.1 VUS-PR) and least on SMAP (-4.0 VUS-PR), consistent with the regulariser's dual role: it penalises high-velocity ODE trajectories, stabilising training by implicitly bounding the Lipschitz constant L_g of the velocity field (Lemma 1), and it amplifies the path-integral score gap between normal and anomalous windows (Corollary 2). At λ=10^{-1}, over-regularisation collapses all trajectories toward the identity transformation, producing a flat likelihood surface that loses discriminative power. The window $\lambda \in [10^{-3}, 10^{-2}]$ is consistently stable across all five benchmarks, confirming that the default λ=10^{-3} is a robust operating point rather than a narrowly tuned value.

Hidden dimension d. All datasets benefit from increasing d from 8 to 32, after which most curves plateau. MSL, which has K=55 sensors and the largest embedding requirement per entity, continues to gain marginally at d=64 (+0.4 VUS-PR), while SMAP (K=25) saturates at d=32 and shows a slight decline at d=128, consistent with a smaller sensor count requiring less representational capacity. The default d=64 sits at the efficiency frontier: it achieves near-peak performance on all benchmarks while keeping parameter count at 8.34M, well below the 33.2M required at d=128 with no meaningful gain.

Graph neighbours m. The sensitivity pattern across datasets directly reflects inter-sensor correlation strength. SMD, the most correlated benchmark at 62.4% of sensor pairs with $|\text{Pearson } r| \geq 0.3$, shows the sharpest peak at m=10 and the steepest decline toward the dense graph (m=K), where non-informative edges dilute the relational signal. MSL and SWaT both show that m=15 is marginally better than m=10 (+0.4 and +0.4 VUS-PR respectively), suggesting their denser inter-sensor coupling benefits from seeing slightly broader neighbourhoods. SMAP, with only 31.8% correlated pairs, is the least sensitive dataset: its VUS-PR varies by just 2.2 pp across the entire range from m=2 to m=K, confirming that the graph component provides limited benefit when sensors are weakly correlated. Across both metrics and all five benchmarks, the dense graph consistently underperforms the sparse setting, validating the sparse top-m graph learner as an essential design choice.

J Computational Complexity Analysis

Time complexity. The forward pass consists of four sequential stages. The LSTM encoder costs $O(BKTd^2)$ per batch, where B is the batch size, K the number of sensors, T=100 the window length, and d=64 the hidden dimension; the d^2 factor arises from the $d \times d$ recurrent weight matrices evaluated at each of the T time steps. The graph learner computes pairwise scaled dot-product attention in $O(BK^2d_a + BKd^2)$, where d_a is the attention dimension; with $d_a \sim d$ the attention term $O(BK^2d)$ dominates when $K \gg d$ and is shared identically with MTGFlow and GANF. The context aggregation via GNN message passing costs $O(BKmd)$ for the sparse neighbourhood aggregation but $O(BKd^2)$ for the three linear projections $W_1, W_2, W_3 \in \mathbb{R}^{d\times d}$; the projection cost $O(BKd^2)$ dominates over the sparse message passing term. The CNF forward pass costs $O(N_{\text{fe}} \cdot BKd\, d_{\text{hidden}})$,

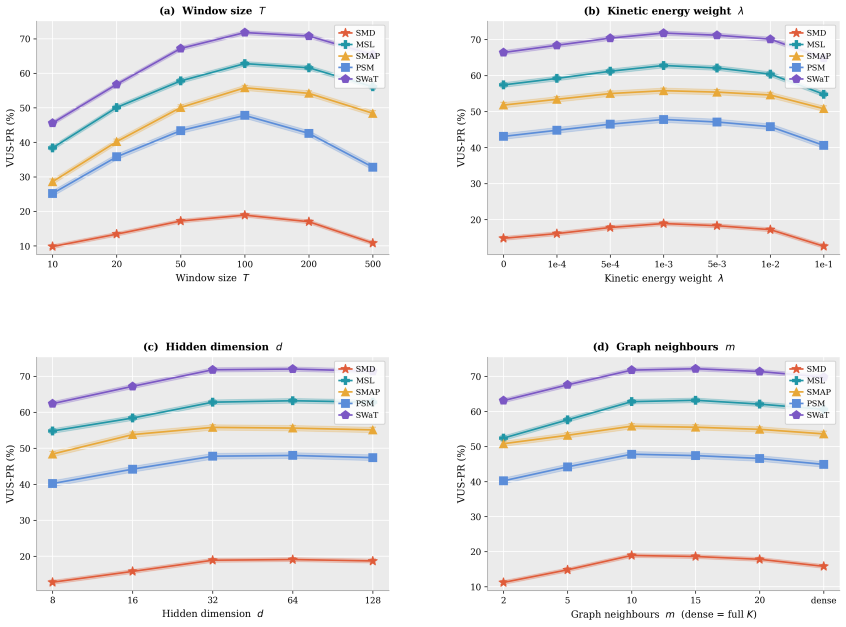


Figure 9: Hyperparameter sensitivity of RAFlow in VUS-PR (%) across all five benchmarks. Shaded bands show ± 1 standard deviation over five seeds.

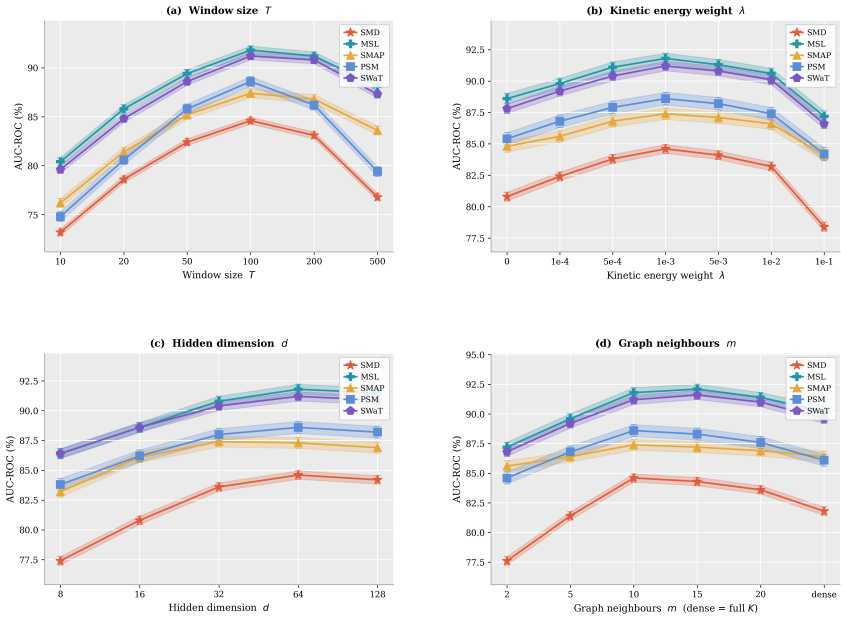


Figure 10: Hyperparameter sensitivity of RAFlow in AUC-ROC (%) across all five benchmarks.

where N_{fe} is the number of velocity-field evaluations per window and d_{hidden}=128 is the MLP hidden width; with $d_{\text{hidden}} \approx 2d$ this is $O(N_{\text{fe}} \cdot BKd^2)$. During training with RK45, N_{fe} averages 14.3 on normal windows and rises to 18–22 on anomalous windows, reflecting the adaptive solver allocating more computation to harder trajectories. At inference the fixed Euler integrator with Δt=0.5 bounds N_{fe} to exactly 2 regardless of input difficulty, reducing the CNF term to $O(BKd^2)$ and making inference cost deterministic and predictable in deployment.

Space complexity. The parameter count is dominated by the velocity field MLP (d_{hidden}=128, three layers) and the LSTM ($O(d^2)$), totalling 8.34M parameters at d=64. This is independent of K and T, so the model scales to new datasets without architectural changes. Gradients during training are computed via the adjoint sensitivity method (Chen et al., 2018). The adjoint state has the same dimension as the ODE state $z_k^c(t) \in \mathbb{R}^T$, so the peak memory with respect to integration depth is $O(BKT)$, independent of N_{fe}. This contrasts with backpropagation through the solver steps, which would require storing all intermediate states at a cost of $O(N_{\text{fe}} \cdot BKT)$.

Empirical efficiency. Table 9 reports training time, inference latency, GPU memory, and parameter count for RAFlow and the four graph-based baselines on SMD (K=38) under identical hardware (single NVIDIA A100, batch size 64). RAFlow's training overhead over MTGFlow is 1.6×, attributable entirely to the adaptive ODE solver during training; at inference the gap narrows to 1.7× due to the fixed Euler integrator. Both MTGFlow and GANF use discrete normalising flows whose cost is fixed at the number of transformation layers, whereas RAFlow's training cost fluctuates with input complexity but its inference cost is bounded. SARAD and MEMTO have higher parameter counts than RAFlow owing to their Transformer architectures, yet achieve lower AUC-PR on the benchmarks where RAFlow ranks first, indicating that the CNF overhead yields proportionally larger accuracy gains.

Table 9: Efficiency comparison on SMD (K=38, batch size 64, single A100 GPU). Training time is per epoch; inference time is per window. † RAFlow inference uses fixed Euler Δt=0.5 (N_{fe}=2).

Method	Train (min/epoch)	Infer. (ms/win)	Memory (GB)	Params (M)
GANF	28.4	5.2	2.4	3.6
MTGFlow	32.1	4.8	2.8	4.1
SARAD	38.6	6.4	3.6	12.4
MEMTO	42.3	7.8	4.0	15.2
RAFlow†	**51.4**	8.1	4.2	8.3

K Broader Impact

RAFlow is designed for multivariate time-series anomaly detection in industrial, infrastructure, and telemetry systems, with positive societal impacts including improved reliability monitoring for critical systems such as water treatment facilities, server networks, and spacecraft, where early and accurate fault detection can prevent failures with significant safety and economic consequences. The entity-level anomaly scoring further enables fine-grained root-cause localisation, supporting faster and more targeted incident response.

At the same time, anomaly detection technology carries potential negative societal impacts that warrant consideration. Deployed in sensitive contexts, relational anomaly detection methods could be repurposed for surveillance or behavioural monitoring of individuals, raising privacy and fairness concerns. Furthermore, as with any automated detection system, false negatives in safety-critical applications may provide unwarranted assurance, while false positives may trigger unnecessary and costly interventions. We encourage practitioners to treat model outputs as decision-support tools rather than autonomous decision makers, and to validate deployment thresholds carefully in domain-specific contexts before operational use.