Diffuse to Detect: A Generalizable Framework for Anomaly Detection with Diffusion Models – Applications to UAVs and Beyond

Mingze Gong*

The Hong Kong University of Science and Technology (Guangzhou), mgong081@connect.hkust-gz.edu.cn

Juan Du*

The Hong Kong University of Science and Technology (Guangzhou), The Hong Kong University of Science and Technology, juandu@ust.hk

Jianbang You

The Hong Kong University of Science and Technology (Guangzhou), The Hong Kong Polytechnic University, jianbangy@hkust-gz.edu.cn

Abstract. Anomaly detection in complex, high-dimensional data, such as UAV sensor readings, is essential for operational safety but challenging for existing methods due to their limited sensitivity, scalability, and inability to capture intricate dependencies. We propose the Diffuse to Detect (DTD) framework, a novel approach that innovatively adapts diffusion models for anomaly detection, diverging from their conventional use in generative tasks with high inference time. By comparison, DTD employs a single-step diffusion process to predict noise patterns, enabling rapid and precise identification of anomalies without reconstruction errors. This approach is grounded in robust theoretical foundations that link noise prediction to the data distribution's score function, ensuring reliable deviation detection. By integrating Graph Neural Networks to model sensor relationships as dynamic graphs, DTD effectively captures spatial (inter-sensor) and temporal anomalies. Its two-branch architecture, with parametric neural network-based energy scoring for scalability and nonparametric statistical methods for interpretability, provides flexible trade-offs between computational efficiency and transparency. Extensive evaluations on UAV sensor data, multivariate time series, and images demonstrate DTD's superior performance over existing methods, underscoring its generality across diverse data modalities. This versatility, combined with its adaptability, positions DTD as a transformative solution for safety-critical applications, including industrial monitoring and beyond.

Key words : Anomaly Detection, Diffusion Models, Unmanned Aerial Vehicles (UAVs), High-dimensional Sensor Data, Graph Neural Networks

1. Introduction

Unmanned Aerial Vehicles (UAVs) have become indispensable across diverse applications, including aerial surveillance, package delivery, disaster response, and environmental monitoring, due to their flexibility, cost-effectiveness, and ability to operate in challenging environments (Shakhatreh et al. 2019). As UAV deployment grows, ensuring operational reliability and safety becomes paramount, especially in high-stakes contexts like urban air mobility and infrastructure inspection. A key aspect of this is monitoring sensor data (encompassing metrics such as altitude, speed,

* These authors contributed equally to this work.

battery levels, and engine performance) to gain insights into a UAV's operational health. High-dimensional anomaly detection is essential for identifying critical issues, such as sudden engine failures, security breaches, or environmental disruptions, could compromise mission success or lead to accidents (Yang et al. 2024).

UAV sensor data, however, poses significant challenges for effective anomaly detection. These data are high-dimensional, collected by numerous sensors with high sampling rates, and exhibit complex multivariate dependencies that shift dynamically based on flight conditions (Huang et al. 2025). Anomalies, such as engine failures, often manifest as abrupt change in sensor readings (e.g., simultaneous drops in power and altitude). Furthermore, UAVs produce diverse data modalities, including structured time series, graphs of sensor interactions, and occasionally images, necessitating detection frameworks capable of handling heterogeneous inputs (Wang et al. 2019). Conventional statistical and deep learning methods struggle to address these complexities, under-scoring the need for tailored approaches to achieve rapid and reliable anomaly detection for UAV safety.

1.1. Limitations of Existing Methods

Current anomaly detection techniques, spanning statistical and deep learning paradigms, exhibit significant limitations when tasked with detecting anomalies in UAV sensor data. These limitations include:

- **Difficulty Detecting Subtle and Abrupt Anomalies in High-Dimensional Data**: Methods like autoencoders or statistical tests often fail to detect anomalies in high-dimensional UAV data, where noise and complex patterns can obscure critical deviations. For instance, abrupt changes following an engine failure may resemble normal pattern locally, leading to low reconstruction errors or insufficient sensitivity, resulting in detection failures (Huang et al. 2025).

- **Inadequate Modeling of Complex Sensor Dependencies**: UAVs produce multivariate time series with intricate inter-sensor relationships. Traditional methods often treat sensors independently or use simplistic correlation models, resulting in detection failures of anomalies arising from coordinated deviations, such as simultaneous drops in power and altitude post-engine failure, which are critical for accurate detection (Lin et al. 2010).

- **Limited Use and Adaptation of Generative Models for Structured Data**: Generative models such as GANs and VAEs are predominantly applied to image-based anomaly detection, with limited use for time series or other structed data, as their high computational cost hinders anomaly detection in UAV operational monitoring, where such data is commonly collected. Their reliance

on reconstruction errors often fails to capture complex anomalies, and advanced generative models are rarely adapted for real-world engineering challenges involving noisy UAV data (Liu et al. 2023).

- **Trade-offs Between Scalability and Interpretability**: Neural network-based approaches scale well for large datasets but lack transparency, obscuring why anomalies are flagged. Conversely, interpretable methods like Kernel Density Estimation (KDE) struggle with high-dimensional data, hindering real-time detection in UAV monitoring. This trade-off complicates diagnosis in safety-critical scenarios (Hu et al. 2020).

- **Limited Generalization Across Diverse Data Modalities**: Most anomaly detection frameworks are designed for specific data types, such as time series or images, and struggle to generalize to others, such as graphs, without significant adjustments. This specialization hinders their application in real-world UAV systems, where heterogeneous data (e.g., sensor readings, images, network graphs) are collected, requiring multiple models and complicating deployment for comprehensive monitoring (Wang et al. 2019).

These challenges are intensified by the curse of dimensionality, where high-dimensional data becomes sparse because the data points grow increasingly spread out and isolated across an exponentially expanding volume of space, undermining distance-based or density-based detection methods (Suboh et al. 2023). Moreover, the rarity and diversity of anomalies in UAV operations, coupled with the scarcity of labeled data, necessitate unsupervised anomaly detection approaches (Lin et al. 2024). However, developing robust unsupervised methods that can effectively handle the high-dimensional and noisy UAV sensor data remains a significant challenge. Together, these limitations highlight the inadequacy of existing approaches for rapidly and reliably detecting critical anomalies in UAV operations without labeled data.

1.2. Contributions of DTD

The identified limitations underscore the need for an anomaly detection framework that is sensitive, scalable, interpretable, and adaptable to the complex, multimodal data from UAVs. Diffusion models (DMs) predict noise patterns in a manner aligned with detecting anomalous deviations in UAV data, with a detailed theoretical derivation provided later. Inspired by recent advancements in generative modeling for anomaly detection (Livernoche et al. 2023, Wang et al. 2023), we propose the Diffuse to Detect (DTD) framework, a customized generative approach for rapid and effective anomaly detection. Unlike conventional methods relying on reconstruction errors to identify anomalies, DTD directly evaluates data deviations using the learned patterns from diffusion models, enhancing efficiency and supported by theoretical validation that validates its efficacy for

anomaly detection. It provides two diffusion-based pathways: DM-P, employing parametric scoring for efficient large-scale data handling of large-scale datasets, and DM-NP, utilizing nonparametric scoring for transparent and interpretable analysis. By reducing inference time compared to traditional anomaly detection generative models based on reconstruction, DTD addresses the real-time demands of UAV monitoring requirements, thereby improving safety and reliability through the following contributions:

- **Enhanced Sensitivity and Effective Modeling of Sensor Dependencies**: Our framework, leveraging the unique mechanism of diffusion models to predict noise patterns, effectively detects subtle irregularities in high-dimensional UAV sensor data, such as early mechanical faults. Additionally, the DTD framework transforms UAV time series into graphs modeling relationships such as among altitude, speed, engine power, and other parameters. This approach captures inter-sensor and temporal dependencies, enabling nuanced anomaly detection.

- **Advanced Application of Generative Models for Structured and Multimodal Data**: We pioneer the development of diffusion models for anomaly detection in structured UAV data, such as time series and graphs, overcoming the limitations of reconstruction-based methods like GANs and VAEs. This framework also generalizes to diverse data types, including images, ensuring robust monitoring for potential real-world UAV applications.

- **Scalable and Interpretable Anomaly Detection Across Diverse Modalities**: Our dual-branch framework leverages diffusion models, with branches selected based on user requirements to address distinct scenarios. The parametric branch, using energy-based scoring, ensures efficient processing of large, high-dimensional UAV data, while the nonparametric branch (KNN, KDE, iForest) provides transparent, diagnostic scoring for smaller-scale scenarios. By extracting robust features from complex data, such as sensor graphs, and enabling robust anomaly detection across diverse modalities such as time series, graphs, and images, it provides reliable, interpretable metrics tied to deviations. This unified approach eliminates the need for separate models, enhancing versatility, trust, and deployment efficiency for safety-critical UAV applications (Hu et al. 2020, Deng et al. 2024).

2. Related Works

Research on anomaly detection in high-dimensional sensor data is broad, spanning reconstruction models, probabilistic likelihood models, and graph-centric approaches. First, reconstruction objectives often miss subtle or abrupt anomalies that remain locally consistent with normal patterns.

Second, distance and subspace scores deteriorate as dimensionality increases or as correlations evolve over time. Third, static graph formulations capture fixed structure but struggle to track the multivariate dependencies that change during flight (see, e.g., Kim and Kim 2023, Ben-Gal et al. 2023, Tao and Du 2025, Xu et al. 2025). With this context, we now turn to UAV-specific methods.

2.1. Statistical Methods for UAV Anomaly Detection

Classical statistical methods, such as Gaussian Mixture Models (GMM), Support Vector Machines (SVM), and Principal Component Analysis (PCA), have been foundational for anomaly detection across domains (Chen et al. 2024). GMMs assume data follows a mixture of Gaussian distributions, but UAV sensor data often exhibits non-Gaussian characteristics, leading to suboptimal performance and sensitivity to outliers (Huang et al. 2025). The computational cost of GMMs also escalates in high-dimensional spaces due to increasing parameter estimation (Yu et al. 2020). SVMs, while effective for some datasets, are computationally intensive for large, high-dimensional UAV data and lack interpretability with non-linear kernels, complicating diagnosis in safety-critical applications. PCA reduces dimensionality but assumes linear relationships, potentially discarding critical information for detecting subtle anomalies in non-linear UAV data (Huang et al. 2025). These methods struggle with the curse of dimensionality, where sparse high-dimensional spaces render distance-based metrics less effective, reducing their ability to detect nuanced deviations (Suboh et al. 2023).

2.2. Deep Learning Methods for UAV Anomaly Detection

Deep learning approaches, particularly for time series data, include Autoencoders (AEs), Recurrent Neural Networks (RNNs), and their variants like LSTMs and GRUs. AEs detect anomalies via high reconstruction errors, but they may reconstruct subtle anomalies with low error, leading to missed detections (Dhakal et al. 2023). RNNs capture temporal dependencies but struggle with complex multivariate interactions in high-dimensional UAV data (Zhou et al. 2024). Generative models like GANs and VAEs, while effective for image anomaly detection, are underutilized for structured data like time series or graphs. When applied, these models rely on reconstruction errors, limiting sensitivity to anomalies that align locally with normal patterns (Ho et al. 2025, Liu et al. 2023). Advanced generative models, such as diffusion models, remain underexplored for UAV anomaly detection, representing an opportunity for capturing complex data distributions (Zhang 2024).

Overall, the literature reveals that prior work often misses non-linear inter-sensor relations, including altitude, groundspeed, and angular velocity, which leads to undetected interaction anomalies (Deng et al. 2024, Lin et al. 2010). High dimensionality degrades clustering and outlier methods

(Suboh et al. 2023), while real time use is limited by the scalability–interpretability trade-off (Hu et al. 2020) and by the need for separate models across modalities (Wang et al. 2019). These gaps call for a unified, sensitive, and interpretable approach. To address these gaps, our framework leverages diffusion models and Graph Neural Networks (GNNs) to capture non-linear dependencies, and by using single step noise prediction to retain sensitivity in high dimensions. Its two branch design balances scalable energy-based scoring (parametric) with interpretable diagnostics (nonparametric). The same framework applies to time series, graphs, and images, simplifying deployment while improving sensitivity, scalability, and interpretability for UAV anomaly detection tasks and broader applications.

3. Methodology
3.1. Preliminaries

Anomaly detection in UAVs identifies deviations in sensor streams that signal critical faults such as engine failures. UAVs produce high dimensional, multivariate time series from heterogeneous sensors such as altimeter, accelerometer and GPS. These sensors form an interdependent network, anomalies often appear as coordinated shifts across signals, necessitating models that capture both inter sensor and temporal dependencies (Deng et al. 2024, Huang et al. 2025).

Inspired by GNNs that can aggregate neighbor information to capture inter-sensor interactions (Ma et al. 2023), we model UAV sensor data relationships via a graph. The nodes, representing individual sensors or logical groups (flight dynamics: altitude, speed, throttle), provide spatial features, which enhance anomaly detection by overcoming independent sensor methods that miss correlated faults. The edges encode dependencies such as the relation between altitude and engine power.

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the sensor graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ represents N sensor nodes, each corresponding to a unique sensor, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes edges based on domain knowledge or correlation analysis of sensor interactions. Each node v_i, representing the i-th sensor, is associated with a feature vector space $x_i(t) \in \mathbb{R}^d$ at time t, capturing sensor measurements (e.g., climb rate, angular velocity). As illustrated in the upper panel of Figure 1, the multivariate time-series data from sensors is segmented using a sliding window approach, resulting in a dynamic graph sequence. The graph evolves over time, forming a sequence $\{\mathcal{G}_t\}$, where each graph $\mathcal{G}_t$ captures the sensor interactions and states at time t. Each sensor v_i has a anomaly-free distribution p_i, characterizing the normal behavior of its feature vector $x_i(t)$. The collection of distributions, denoted $p_{\text{data}} = \{p_i\}_{i=1}^{N}$, represents the expected behavior of the sensor network under normal
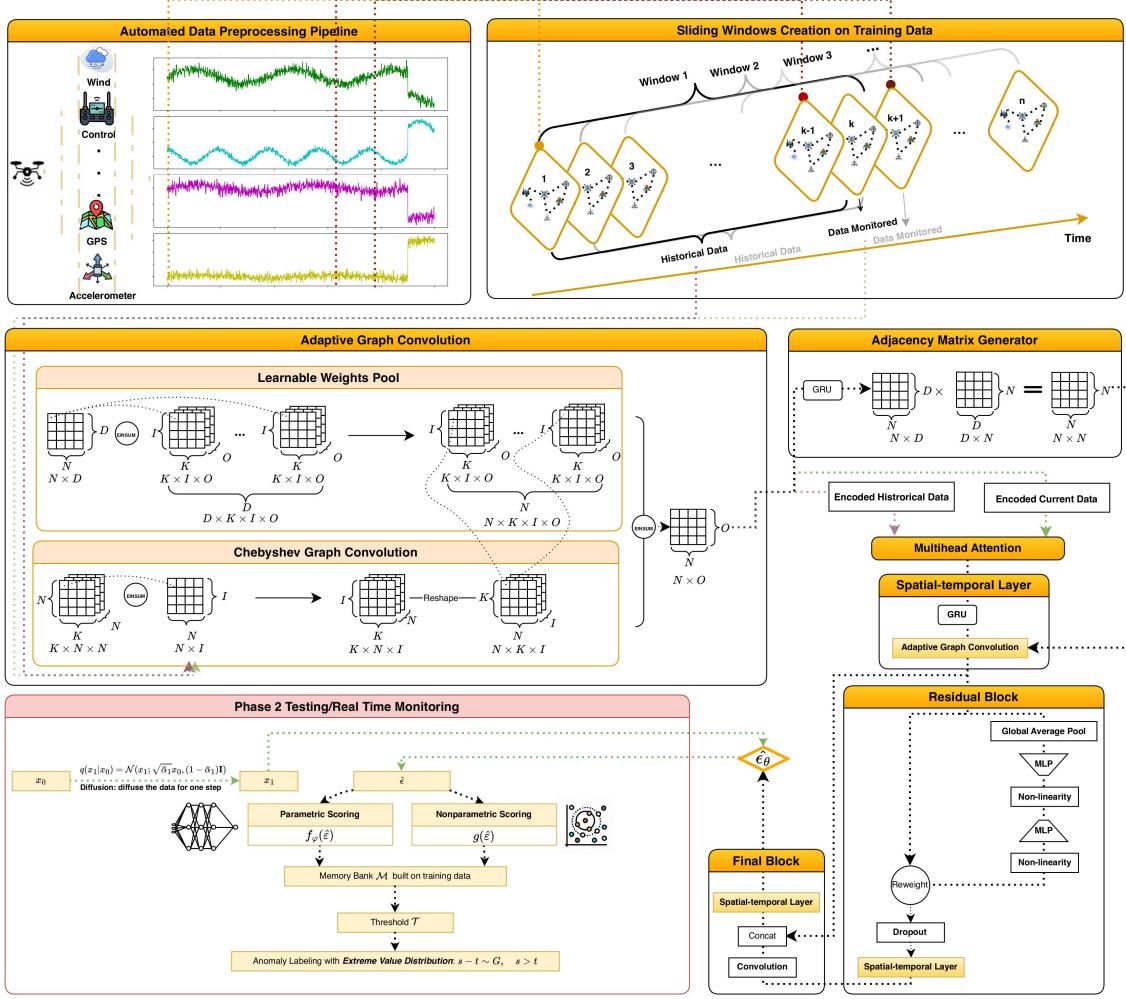
Figure 1 An Overview of the Framework. The Diffuse to Detect (DTD) framework first transforms UAV sensor data into a graph structure, where nodes represent sensors and edges initialized first and learnt through stochastic gradient descent. The framework then trains a diffusion model to learn the anomaly-free data distribution, enabling the prediction of noise patterns. For anomaly detection, the framework applies a single diffusion step to a test sample, generating a perturbed sample and predicting its noise. Two scoring branches (parametric and nonparametric) evaluate the predicted noise against the learned distribution, providing robust anomaly scores. Finally, the framework uses Extreme Value Theory (EVT) to compare scores against a threshold, enabling the detection of anomalies.

conditions. Anomalies are defined according to sensor states where the observed data distribution of the i-th sensor deviates significantly from its normal distribution p_i. The final goal of anomaly detection is to output a binary label $y^t \in \{0, 1\}$ indicating whether the UAV given at time t is anomalous (1) or normal (0), given all sensor measurements. To prevent confusion with the diffusion

process in later sections, the sensor-specific index i is omitted when referring to general features, using x to denote a sample's feature vector.

3.2. Overview of the DTD Framework

Diffusion models provide a principled framework for modeling the probabilistic structure of data distributions. These models learn an approximation p_θ to the normal data distribution p_{data}, where p_{data} represents the distribution of the original, noise-free (raw) data samples $x_0 \sim p_{\text{data}}$. We train a neural network that outputs $\epsilon_\theta(x_k, k)$, the predicted Gaussian perturbation ($\sim \mathcal{N}(0, I)$) applied to a noisy sample x_k at diffusion step $k \in \{0, \ldots, T-1\}$. Here, $k = 0$ corresponds to the original, noise-free data, while larger values of k represent increasingly noisy versions of the sample. Note that the diffusion time step k used in x_k refers to the level of noise applied during the diffusion process and should not be confused with the temporal index t in the graph sequence $\{\mathcal{G}_t\}$, which denotes actual time steps in the sensor data. The diffusion time step k governs the denoising schedule, while the chronological order of UAV observations corresponds to the temporal sequence of sensor data. Through training the neural network, the output ϵ_θ captures the noise patterns characteristic of normal data under varying perturbation levels, effectively encoding the probabilistic structure of p_{data} in its predictions.

For UAV anomaly detection (Fig. 1), we apply a single forward diffusion step ($k = 1$) to a test sample x to obtain x_1. The network predicts the perturbation $\hat{\epsilon} = \epsilon_\theta(x_1, 1)$. For normal data, $\hat{\epsilon}$ is approximately distributed as $\mathcal{N}(0, I)$, while anomalies will show systematic deviations. By limiting perturbation to a single step, our method preserves the intrinsic features of x, enhancing sensitivity to subtle anomalies, such as minor sensor discrepancies. To quantify such deviations, we propose two scoring mechanisms that leverage the learned representation reflected in the predicted noise $\hat{\epsilon}$, each addressing different practical considerations for anomaly detection. The **parametric** branch scales to high-dimensional UAV data using an energy-based model that scores divergence from $\mathcal{N}(0, I)$. The **nonparametric** branch favors interpretability, using kNN or KDE to compare $\hat{\epsilon}$ against the normal noise distribution. Users select a branch based on scalability versus transparency, leveraging the link between p_θ and the noise modeled by ϵ_θ to detect both abrupt faults and subtle irregularities. This design suits real-time UAV monitoring where both efficiency and precision are critical. The next sections provide the model formulation, theoretical justifications, and scoring details building on diffusion models (Ho et al. 2020).

3.3. Diffusion Model Formulation

Diffusion models approximate the distribution p_θ as p_{data} through a Markov chain that corrupts data with incremental noise and learns the reverse transitions for denoising-based reconstruction. The **forward process** operates as a Markov chain, applying a fixed noise schedule to data, while the **reverse process** samples cleaner data using a trained neural network. Below, we clearly explain the training of noise prediction model used to detect anomalies and the formulation in UAV applications.

The **forward process** perturbs a clean data sample $x_0 \in \mathbb{R}^d$, drawn from p_{data}, over T diffusion time steps $k = 0, 1, \ldots, T - 1$. Gaussian noise is added incrementally, producing noisy samples x_k. Mathematically, at diffusion time step k:

$$x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1 - \bar{\alpha}_k} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\bar{\alpha}_k = \prod_{s=0}^{k} \alpha_s$ with $\alpha_s = 1 - \beta_s$, $\beta_s \in (0, 1)$ a predefined noise increment, and ϵ is standard Gaussian noise. At $k = 0$, $\bar{\alpha}_0 = 1$, so x_0 is unchanged; at $k = T - 1$, $\bar{\alpha}_{T-1} \approx 0$, making x_{T-1} nearly pure noise. This forward process is characterized by a conditional Gaussian distribution:

$$q(x_k | x_0) = \mathcal{N}\left(x_k \middle| \sqrt{\bar{\alpha}_k} x_0, (1 - \bar{\alpha}_k) I\right), \tag{2}$$

indicating that x_k combines the original data x_0, scaled by $\sqrt{\bar{\alpha}_k}$, with noise of variance $1 - \bar{\alpha}_k$. In a UAV context, x_k represents sensor data with increasing perturbations, similar to noise encountered during flight.

In generative tasks, the **reverse process** of diffusion models reconstructs clean data by iteratively denoising x_{T-1}. However, its high computational cost, requiring thousands of steps, makes it unsuitable for real-time UAV anomaly detection. Our approach leverages only the **forward process**, directly analyzing the predicted noise for efficient anomaly detection, largely eliminating the need for iterative and time-consuming reconstruction. Further details on the **reverse process** and complete reconstruction of x_0 refer to Ho et al. (2020).

To support our anomaly detection framework, we utilize the noise prediction capability of diffusion models. A neural network, parameterized by θ, is trained to predict the noise ϵ_θ, approximating the noise ϵ introduced in Equation (1), using the noisy sample x_k, its historical window x_{hist}, and diffusion time step k. The noise ϵ is intrinsically linked to the data distribution, as x_k is generated from x_0 and ϵ according to Equation (1), enabling ϵ_θ to capture patterns of normal data. We leverage this predicted noise for anomaly detection, with theoretical justification provided in the next Subsection 3.4 and detailed derivations in Appendix A.

We completely redesigned the model to effectively process UAV data with inter-sensor and temporal dependencies, tailoring it for anomaly detection in UAV sensor systems. Predefining sensor relationships is impractical in this context due to the large number of sensors on a UAV and their complex interactions.

To address the challenge of modeling sensor relationships without relying on a predefined graph structure, we further developed the adaptive structure proposed by Bai et al. (2020) and implemented a purely data-driven approach to infer these relationships, as shown in Figure 1. Our adaptive graph convolution module employs a learnable weights pool and Chebyshev graph convolution (He et al. 2022) to capture spatial dependencies between sensors dynamically, eliminating the need for a static graph. We fuse encoded history with the current test sample using multihead attention (Vaswani et al. 2017), aligning long-range context with present observations to better expose subtle anomalies in multivariate UAV sensors. The fused features then pass to a spatiotemporal layer that combines a GRU for temporal dynamics with adaptive graph convolution for spatial dependencies, producing a comprehensive representation for detection. Finally, to produce noise predictions, we add a residual block followed by a projection block. The residual block stabilizes training and reduces overfitting, while the projection block concatenates features and applies a convolution to yield a refined representation for real-time parametric and nonparametric scoring.

Our framework includes two branches, each selected for specific scenarios: the parametric branch prioritizes computational efficiency, while the nonparametric branch emphasizes interpretability. We employ a cooperative training strategy to fully leverage the model's capacity, as detailed in Algorithm 1. Specifically, at each iteration, we draw x_0 from the dataset, pick a random diffusion step k, sample $\epsilon \sim \mathcal{N}(0, I)$, and form x_k via the forward process.

The diffusion loss, which forms the first part of the training objective, is calculated as:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0 \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0, I), k} \left[\| \epsilon - \epsilon_\theta(x_k, k, x_{hist}) \|_2^2 \right]. \tag{3}$$

In addition to the diffusion loss L_{DM}, each branch of the diffusion model incorporates a specific loss term to enhance anomaly detection capabilities.

For the nonparametric branch, an additional loss L_{NP} is computed using methods such as Kernel Density Estimation (KDE), k-Nearest Neighbors (kNN), or Isolation Forest (iForest). This loss leverages a memory bank $\mathcal{M}$ that stores predicted noise $\hat{\epsilon}^+$ from normal samples. The loss is calculated as:

$$L_{\text{NP}} = \mathcal{L}_{\text{nonparam}}(\hat{\epsilon}^+, \hat{\epsilon}^-, \mathcal{M}), \tag{4}$$

where:

- $\hat{\epsilon}^+ = \epsilon_\theta(x_0, 0, x_{\text{hist}})$ is the predicted noise for a normal sample.

- $\hat{\epsilon}^- = \epsilon_\theta(x_k, 0, x_{\text{hist}})$ is the predicted noise for an anomalous or noisy sample.

- $\mathcal{L}_{\text{nonparam}}$ distinguishes normal and anomalous noise patterns based on the chosen nonparametric method.

This approach allows the model to detect anomalies by comparing new noise predictions against a historical memory bank of normal patterns.

For the parametric branch, an Energy-Based Model (EBM) is employed, implemented as a neural network that processes predicted noise. The EBM uses positive samples $\hat{\epsilon}^+ = \epsilon_\theta(x_0, 0, x_{\text{hist}})$ from normal data and negative samples $\hat{\epsilon}^- = \epsilon_\theta(x_k, 1, x_{\text{hist}})$ from diffused data to distinguish between normal and anomalous patterns. The EBM loss is calculated as:

$$L_{\text{EBM}} = E_\phi(\hat{\epsilon}^+) - E_\phi(\hat{\epsilon}^-). \tag{5}$$

The total training loss combines the diffusion loss with the branch-specific loss, weighted by a hyperparameter λ:

- For the nonparametric branch:

$$L_{\text{total}} = L_{\text{DM}} + \lambda L_{\text{NP}}. \tag{6}$$

- For the parametric branch:

$$L_{\text{total}} = L_{\text{DM}} + \lambda L_{\text{EBM}}. \tag{7}$$

This cooperative scheme learns accurate noise prediction and strong normal–anomaly discrimination. Branch choice here depends on whether scalability and efficiency are prioritized with the parametric branch or interpretability and transparency with the nonparametric branch. The following subsection details the scoring mechanisms.

3.4. Noise-Based Anomaly Detection

To justify the efficacy of the predicted noise $\epsilon_\theta(x_k, k, x_{hist})$ for anomaly detection, we establish Proposition 1 linking the noise prediction to the score function of the data distribution, offering a solid foundation for detecting deviations in UAV sensor data.

PROPOSITION 1. *For a diffusion model with forward process* $x_k = \sqrt{\bar{\alpha}_k}\, x_0 + \sqrt{1 - \bar{\alpha}_k}\, \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, *the noise predictor satisfies*

$$\epsilon_\theta(x_k, k, x_{hist}) \approx -\sqrt{1 - \bar{\alpha}_k}\, \nabla_{x_k} \log p_k(x_k \mid x_{hist}),$$

Algorithm 1 Training Diffusion Model with Nonparametric or Parametric (EBM) Branch

---

Require: Dataset $D \sim p_{\text{data}}$ with sliding window pairs (x_0, x_{hist}), the pre-defined noise schedule $\{\beta_k\}_{k=0}^{T-1}$, a neural network $\epsilon_\theta(\cdot)$ parameterized by θ, branch type

1: Initialize model parameters of ϵ_θ
2: Initialize empty memory bank: $\mathcal{M} \leftarrow \emptyset$
3: **while** not converged **do**
4:     Sample $(x_0, x_{\text{hist}}) \sim D$
5:     Sample time step $k \sim \text{Uniform}\{0, 1, \ldots, T-1\}$
6:     Compute $\bar{\alpha}_k = \prod_{s=0}^{k}(1-\beta_s)$
7:     Sample noise $\epsilon \sim \mathcal{N}(0, I)$
8:     Compute noisy sample: $x_k = \sqrt{\bar{\alpha}_k}x_0 + \sqrt{1-\bar{\alpha}_k}\epsilon$
9:     Predict noise: $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$
10:     Compute diffusion loss: $L_{\text{DM}} = \|\epsilon - \hat{\epsilon}\|_2^2$
11:     **if** branch == "Nonparametric" **then**   ▷ e.g., KDE, kNN, iForest
12:         Predict positive sample: $\hat{\epsilon}^+ = \epsilon_\theta(x_0, 0, x_{\text{hist}})$
13:         Predict negative sample: $\hat{\epsilon}^- = \epsilon_\theta(x_k, 0, x_{\text{hist}})$
14:         Compute nonparametric loss: $L_{\text{NP}} = \mathcal{L}_{\text{nonparam}}(\hat{\epsilon}^+, \hat{\epsilon}^-, \mathcal{M})$
15:         **if** memory bank is full **then**
16:            Remove oldest entry from $\mathcal{M}$ ▷ FIFO update
17:         **end if**
18:         Update memory bank: $\mathcal{M} \leftarrow \mathcal{M} \cup \hat{\epsilon}^+$
19:         Compute total loss: $L_{\text{total}} = L_{\text{DM}} + \lambda L_{\text{NP}}$
20:     **else if** branch == "Parametric" **then**   ▷ EBM-based refinement
21:         Predict positive sample: $\hat{\epsilon}^+ = \epsilon_\theta(x_0, 0, x_{\text{hist}})$
22:         Predict initial negative sample: $\hat{\epsilon}^- = \epsilon_\theta(x_k, 0, x_{\text{hist}})$
23:         Refine negative sample via Langevin dynamics:
            $\hat{\epsilon}^- \leftarrow \text{Langevin}(\hat{\epsilon}^-, \nabla_{\hat{\epsilon}}E_\phi(\hat{\epsilon}))$
24:         Compute EBM loss: $L_{\text{EBM}} = E_\phi(\hat{\epsilon}^+) - E_\phi(\hat{\epsilon}^-)$
25:         Compute total loss: $L_{\text{total}} = L_{\text{DM}} + \lambda L_{\text{EBM}}$
26:     **end if**
27:     Update model parameters: $\theta \leftarrow \text{Adam}(\theta, \nabla_\theta L_{\text{total}})$
28: **end while**
29: **Output:** Trained model ϵ_θ

---

where $p_k(x_k)$ is the distribution of x_k. When k is small, such as $k = 1$, $\epsilon_\theta(x_0, 1, x_{hist}) \approx -\sqrt{1-\bar{\alpha}_0}\nabla_{x_0}\log p_{data}(x_0|x_{hist})$ encodes the local geometry of p_{data}.

The derivation in Appendix A uses the forward process (Eq. 1) and Tweedie's formula (Efron 2011) to show that the optimal predictor $\epsilon_\theta^*(x_k, k, x_{\text{hist}}) = \mathbb{E}[\epsilon \mid x_k, x_{\text{hist}}]$ is proportional to the negative score $\nabla_{x_k}\log p_k(x_k \mid x_{\text{hist}})$. The training objective drives $\epsilon_\theta(x_k, k, x_{\text{hist}}) \approx \epsilon_\theta^*$. Intuitively, for normal data, the prediction follows the noise pattern implied by p_{data} as the score points

Algorithm 2 Testing Algorithm in DTD Framework

Require: Test sample x, trained diffusion model ϵ_θ, energy-based model E_ϕ, memory bank $\mathcal{M} = \{\hat{\epsilon}_i^+\}_{i=1}^M$, time step $k = 1$

1: Sample noise $\epsilon \sim \mathcal{N}(0, I)$

2: Compute $\bar{\alpha}_k = \prod_{s=0}^{k}(1 - \beta_s)$ for $k = 1$

3: Compute noisy input: $x_k = \sqrt{\bar{\alpha}_k}x + \sqrt{1 - \bar{\alpha}_k}\epsilon$

4: Predict noise: $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist})$

5: **if** branch == "Nonparametric" **then**

6:    Compute anomaly score $s(x)$ using $\hat{\epsilon}$ and $\mathcal{M}$ ▷ Higher score means anomalous

7: **else if** branch == "Parametric" **then**

8:    Compute anomaly score: $s(x) = E_\phi(\hat{\epsilon})$ ▷ Higher energy indicates anomalous

9: **end if**

10: **Output:** Anomaly score $s(x)$

---

toward high-density regions, whereas faults induce deviations that ϵ_θ exposes as anomalies. Our approach's sensitivity and efficiency are supported by $\epsilon_\theta(x_k, k, x_{hist})$, which captures distributional discrepancies to enable rapid and robust detection of subtle faults in UAVs, enhancing safety and reliability in critical scenarios.

3.5. Scoring Methodology

The testing procedure for the DTD framework is outlined in Algorithm 2, which details the computation of anomaly scores for a test sample using the trained diffusion model and scoring branch. This algorithm leverages the predicted noise from a lightly perturbed input at diffusion time step $k = 1$ to assess deviations from normal data distributions, enabling efficient and robust anomaly detection in UAV sensor data and beyond. Instead of directly predicting the noise at $k = 0$, we apply a single diffusion step to the test sample. The justification for this approach is provided in Appendix C.

Our DTD framework employs two distinct scoring branches to quantify anomalies based on the predicted noise $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist})$, addressing different operational needs in UAV monitoring. The nonparametric scoring branch offers interpretable alternatives using statistical methods. The parametric scoring branch leverages an energy-based model for efficient and scalable anomaly detection, ideal for high-dimensional sensor data. Both approaches ensure robust detection by exploiting the diffusion model's learned representation of normal data distributions.

3.5.1. Nonparametric Scoring The nonparametric branch scores the predicted noise $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$, computed at a small step k (typically $k = 1$). For a test sample x we form $x_k = \sqrt{\bar{\alpha}_k}x + \sqrt{1 - \bar{\alpha}_k}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ and then compute $\hat{\epsilon}$. Deviations of $\hat{\epsilon}$ from the normal distribution of diffusion model yield high anomaly scores, while alignment with $\mathcal{N}(0, I)$ yields low scores. This design provides robust detection across images, time series, and sensor streams.

The framework maintains a memory bank $\{\hat{\epsilon}_i^+\}_{i=1}^M$ of predicted noises from normal training data, with $\hat{\epsilon}_i^+ = \epsilon_\theta(x_{k,i}, k)$. We set $k = 0$ to capture the empirical distribution of normal noise predictions. Nonparametric scorers compare a test $\hat{\epsilon}$ to this reference to quantify deviation, yielding low scores for normal data and high scores for anomalies across images and multivariate time series. Examples of applicable techniques include the following, tailored to experimental contexts:

- **Kernel Density Estimation (KDE).** Gaussian kernel with bandwidth $h = 1.06\,\sigma M^{-1/5}$:

$$s_{\text{KDE}}(\hat{\epsilon}) = -\log\left(\frac{1}{M}\sum_{i=1}^M K_h(\hat{\epsilon}, \hat{\epsilon}_i^+) + 10^{-8}\right).$$

Lower density implies a higher score.

- **k-Nearest Neighbors (kNN).** Mean distance to the k nearest neighbors:

$$s_{\text{kNN}}(\hat{\epsilon}) = \frac{1}{k}\sum_{j \in \mathcal{N}_k(\hat{\epsilon})} \|\hat{\epsilon} - \hat{\epsilon}_j^+\|_2.$$

Larger distances imply a higher score.

- **Isolation Forest (iForest).** Inverse average path length:

$$s_{\text{iForest}}(\hat{\epsilon}) = 2^{-\mathbb{E}[\text{path}(\hat{\epsilon})]/c(M)}, \quad c(M) \approx 2\ln(M-1) + 0.5772 - \frac{2(M-1)}{M}.$$

Shorter paths imply a higher score.

3.5.2. Parametric EBM Scoring The parametric branch uses an Energy-Based Model (EBM) with parameters ϕ to score the predicted noise $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$ from a small diffusion step (typically $k = 1$). For a test sample x with context x_{hist}, we form $x_k = \sqrt{\bar{\alpha}_k}\,x + \sqrt{1 - \bar{\alpha}_k}\,\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. We then compute $\hat{\epsilon}$ and evaluate the energy $E_\phi(\hat{\epsilon})$. Lower energy indicates conformity to normal data. Higher energy flags anomalies.

PROPOSITION 2. *Let the diffusion model be trained on $p_{data}(x \mid x_{hist})$ with noise predictor $\epsilon_\theta(x_k, k, x_{hist})$ and an EBM parameterized by ϕ. Then the energy $E_\phi(\epsilon_\theta(x_k, k, x_{hist}))$ serves as a monotone proxy for the negative log-likelihood $-\log p_{\theta,\phi}(x \mid x_{hist})$ under the induced diffusion-based likelihood.*

A derivation is provided in Appendix B. We construct $p_{\theta,\phi}(x \mid x_{\text{hist}})$ by marginalizing over the noise and relate the EBM energy to $-\log p_{\theta,\phi}$ up to an additive constant. Intuitively, for normal UAV sensor data conditioned on historical context x_{hist}, the predicted noise $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist})$ aligns with the training distribution, yielding low $E_\phi(\hat{\epsilon})$ due to the EBM optimization. Anomalous

data, such as erratic sensor readings deviating from expected patterns given x_{hist}, produce $\hat{\epsilon}$ with high energy, enabling effective anomaly detection.

The EBM defines an unnormalized probability distribution over the predicted noise:

$$p_\phi(\hat{\epsilon}) = \frac{1}{Z(\phi)} \exp\left[f_\phi(\hat{\epsilon}) \right],$$ (8)

where $f_\phi : \mathbb{R}^d \to \mathbb{R}$ is a negative energy function implemented as a multi-layer perceptron with parameters ϕ, and $Z(\phi) = \int \exp\left[f_\phi(\hat{\epsilon}) \right] d\hat{\epsilon}$ is the intractable normalizing constant. The anomaly score is defined as $E_\phi(\hat{\epsilon}) = -f_\phi(\hat{\epsilon})$, where high $E_\phi(\hat{\epsilon})$ (low $f_\phi(\hat{\epsilon})$) indicates anomalies, and low $E_\phi(\hat{\epsilon})$ corresponds to normal data.

The EBM is trained contrastively to distinguish normal from anomalous noise predictions. Positive samples $\hat{\epsilon}^+ = \epsilon_\theta(x_k, k)$ are generated from normal data $x \sim p_{\text{data}}$. Negative samples $\hat{\epsilon}^-$ are synthesized to be distinct from normal data using Markov chain Monte Carlo (MCMC) via Langevin dynamics, ensuring they lie further from the learned distribution. The dynamics iterate as:

$$\hat{\epsilon}_m^- = \hat{\epsilon}_{m-1}^- + \frac{\delta^2}{2}\nabla_{\hat{\epsilon}} f_\phi(\hat{\epsilon}_{m-1}^-) + \delta\eta_m, \quad \eta_m \sim \mathcal{N}(0, I),$$ (9)

where m indexes the MCMC step, δ is the step size, η_m is Brownian noise, and the chain is initialized from a uniform distribution or perturbed normal samples. This process drives $\hat{\epsilon}^-$ towards regions of lower probability under p_ϕ, enhancing their distinctiveness from $\hat{\epsilon}^+$.

The parametric approach is computationally efficient, requiring only a single forward pass through the diffusion model and EBM, making it scalable for high-dimensional UAV sensor data. It excels in real-time applications, detecting faults like sudden power drops with low latency, enhancing safety in critical scenarios.

3.6. Anomaly Labeling with EVT

We post-process DTD scores using Peaks Over Threshold (POT) from Extreme Value Theory. Scores s_t from the parametric and nonparametric branches are computed from the predicted noise $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$ at a small step k (typically $k = 1$) with $x_k = \sqrt{\bar{\alpha}_k}\,x + \sqrt{1 - \bar{\alpha}_k}\,\epsilon$. Using normal-training scores $\{s_i^+\}_{i=1}^M$, we choose a high threshold t, fit a Generalized Pareto Distribution to excesses $s > t$, and set the adaptive decision level

$$z_q = t + \frac{\hat{\sigma}}{\hat{\gamma}}\left(\left(\frac{qM}{N_t}\right)^{-\hat{\gamma}} - 1 \right),$$

where N_t is the number of excesses and q (10^{-3}) is a risk parameter. Test scores above z_q are labeled as anomalous. POT provides data-driven thresholds with small memory and low overhead, enabling reliable streaming and real-time UAV fault detection.

4. Experimental Setup
4.1. Dataset Description

To evaluate the DTD framework, we utilize four diverse datasets: the Air Lab Fault and Anomaly (ALFA) dataset, Biomisa Arducopter Sensory Critique (BASiC) dataset, the Server Machine Dataset (SMD), and the CIFAR-10 dataset.

The ALFA dataset, which is a real-world UAV corpus for fault and anomaly detection (Keipour et al. 2021) contains 47 autonomous flights with 66 minutes of normal data and 13 minutes of post-fault data. Scenarios cover 23 full engine failures and 24 control-surface faults (rudder, aileron, elevator). For ALFA, we built an automated preprocessing pipeline (to be released) that resamples to 50 Hz, interpolates for temporal consistency, and drops the first 20 s to remove early-flight instability such as takeoff and controller initialization, following Keipour et al. 2019. Guided by domain work (Chen et al. 2024, Jiang et al. 2024), we select informative signals and organize them into 19 graph nodes that act as logical sensors, with 14 features per node to support spatiotemporal modeling. The processed data are serialized for reproducibility and split 70%, 15%, 15% into train, validation, and test. Fault timestamps and types provide ground truth for both DTD branches. To broaden evaluation beyond ALFA, we also use BASiC (Ahmad et al. 2024), which offers 70 autonomous flights and more than seven hours of data.

SMD is a five-week collection of resource-utilization traces from 28 cluster machines. We evaluate two challenging sequences (Machine 1-1 and 2-1) highlighted by Tuli et al. (2022). The data are multivariate time series of system metrics such as CPU, memory, and disk I/O. Unlike ALFA's sudden faults, SMD exhibits intermittent anomalies that can be sporadic or persistent, stressing temporal robustness. We split the dataset into 70% training, 15% validation, and 15% test sets using the provided labels for evaluation.

CIFAR-10 contains 60,000 32×32 color images across 10 classes (Krizhevsky 2009). We use a one-vs-rest setup with one class as normal and the rest as anomalous. Images are embedded with ResNet-152 to obtain high-level features, then scored by DTD. The split is 70% train, 15% validation, and 15% test, with labels repurposed to define normal vs. anomalous instances. This setting evaluates DTD on image anomalies and complements the time-series datasets.

We selected these datasets for two reasons. First, their modality diversity, with time series (ALFA, BASiC, SMD) and images (CIFAR 10), shows DTD's generalizability to UAV relevant formats such as sensor streams and onboard image. Second, they present different anomaly profiles, where ALFA captures sudden faults, SMD contains intermittent anomalies during normal operation, and CIFAR 10 tests static image anomalies. This diversity shows robustness across heterogeneous data and real world scenarios. Detailed statistics are in Appendix E.

4.2. Evaluation Metrics

We evaluate the DTD framework using standard anomaly detection metrics: precision, recall, F1-score, and accuracy. Details for these metrics are provided in Appendix D. The experiments were conducted on an NVIDIA A6000 workstation equipped with 48 GB of GPU memory and 128 GB of RAM. To ensure robustness, all experiments were replicated five times with different random seeds, and the results were averaged. The code and datasets will be made publicly available upon publication.

5. UAV Case Study: Results and Discussion

We first evaluate DTD's parametric (DM-P) and nonparametric (DM-NP) branches on UAV sensor data, while later sections report results on other modalities.

5.1. Performance Metrics and Visualization of Anomaly Scores

The left column of Table 1 presents a comparative evaluation of the DTD framework's branches against established baselines, including the ALFA baseline (Keipour et al. 2021) and an advanced Transformer-based approach (Ahmad et al. 2024). The results underscore the exceptional performance of DTD's branches, with DM-NP demonstrating leading accuracy and robustness, closely followed by DM-P, both surpassing the baselines in effectively identifying anomalies while minimizing false positives and negatives. As shown in Table 1, we observe similar trends in the BASiC dataset, where DM-NP and DM-P outperform the Transformer baseline, particularly in precision and F1-score. Though the Transformer baseline exhibits superior recall on this dataset, this advantage comes at the expense of significantly lower precision and F1-score, whereas DM-NP and DM-P achieve a more balanced and reliable performance across all metrics, reinforcing their suitability for robust UAV anomaly detection across diverse datasets.

For the ALFA dataset (Flight 0), where an engine failure occurs around sample index 4000, Figure 2 (a) and Figure 2 (b) depict the DM-NP and DM-P anomaly scores, respectively, with score increases aligning well with the ground truth regions shaded in red. Figure 2 (c) and Figure 2

Table 1   **Performance on ALFA and BASIC datasets (UAV sensor data, graph-structured). Reported values are the mean ± standard deviation over 5 runs.**

Method	ALFA			BASIC		
	Prec.	Rec.	F1	Prec.	Rec.	F1
DM-NP	**0.9914**±0.006	**0.9890**±0.006	**0.9901**±0.005	**0.9897**±0.014	0.9415±0.018	**0.9648**±0.009
DM-P	**0.9943**±0.008	**0.9678**±0.006	**0.9807**±0.004	**0.9911**±0.015	0.9027±0.032	**0.9443**±0.018
ALFA Base. (Keipour et al. 2021)	0.7365	0.9115	0.8119	–	–	–
Transformer (Ahmad et al. 2024)	0.8720	0.9437	0.8928	0.5360	**1.0000**	0.6979

(a) DM-NP (KDE) Scores (ALFA)

(c) DM-NP (KDE) Predictions (ALFA)

(b) DM-P Scores (ALFA)

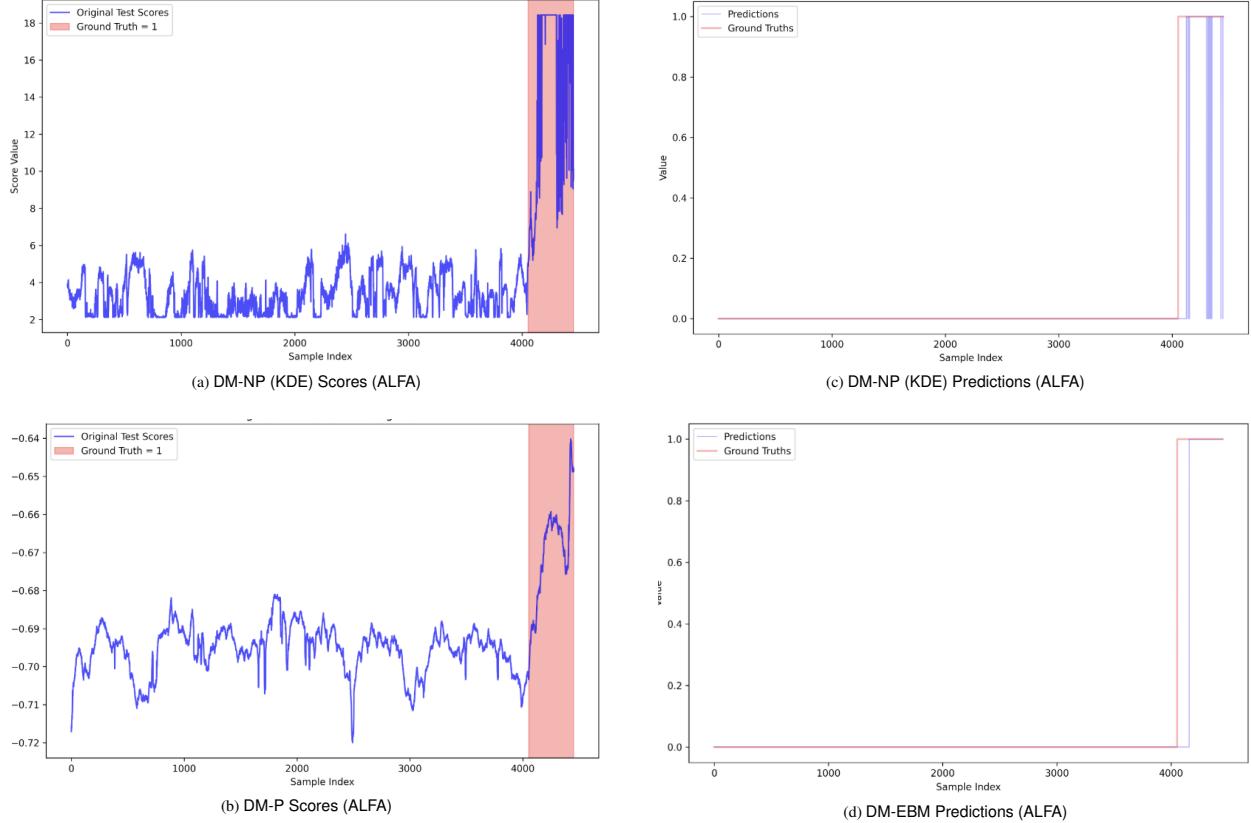(d) DM-EBM Predictions (ALFA)

Figure 2   Anomaly scores and predictions for ALFA dataset. Left column (a–b): scores; right column (c-d): predictions. Ground truth (fault) regions are shaded in red.

(d) show the corresponding predictions, where both branches accurately capture the fault onset. The DTD framework achieves zero false positives in the ALFA dataset, a critical advantage for safety-critical UAV applications. While some scores in fault regions may appear less pronounced, the framework prioritizes detection accuracy over score magnitude. As long as scores exceed the threshold set by Section 3.6, anomalies are reliably identified, meeting the task's requirements, though minor detection delays may occur in rare cases. Similar performance is observed in the BASiC dataset, visualized in Appendix G .
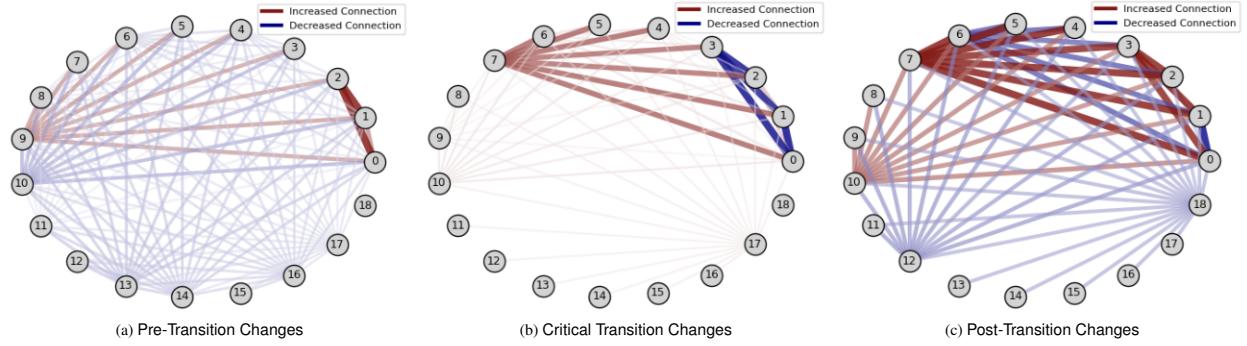
(a) Pre-Transition Changes   (b) Critical Transition Changes   (c) Post-Transition Changes

Figure 3   **Visualization of UAV nodes connection in the DM-P (KDE) for graph-structured ALFA sensor data.**

5.2. Analysis of Node Connection Changes in UAV Sensor Data

Using ALFA's graph-structured UAV data, we analyze how DTD tracks inter-sensor relationships across three phases (Pre-Transition, Critical-Transition, Post-Transition). As shown in Figure 3, nodes are sensors (Table 6, Appendix F); red/blue edges mark strengthened/weakened connections between successive phases. Changes are minimal pre-transition, localized at onset, and widespread post-transition, reflecting the anomaly's expanding impact on sensor dynamics.

5.2.1. Pre-Transition Changes During stable flight, connectivity shifts are small. Figure 3 (a) shows slight strengthening from node 2 (local position) to nodes 1 (flight dynamics) and 0 (navigation errors), consistent with routine stabilization against minor drifts. Modest gains around nodes 8 (global position) and 9 (magnetic field) likely reflect benign navigation adjustments.

5.2.2. Critical Transition Changes At anomaly onset (e.g., engine failure), Figure 3 (b) shows strong connection increases between node 7 (control outputs) and nodes 4–6 (wind, GPS velocity, measured velocity), indicating rapid fusion to counter thrust/stability loss. Meanwhile, node 3 (IMU) shows weakend connections toward nodes 0–2 (navigation errors, flight dynamics, local position), suggesting disrupted IMU reliability and a shift to other cues.

5.2.3. Post-Transition Changes As recovery begins, connectivity reaches peak. Figure 3 (c) shows node 7 (control outputs) strengthening broadly to nodes 0–6 (navigation errors, flight dynamics, local position, IMU, wind, GPS velocity, measured velocity), indicating system-wide fusion to regain stability. By contrast, node 12 (yaw) weakens toward nodes 0–2, consistent with prioritizing velocity/position over heading.

5.2.4. Summary of Sensor Relationship Dynamics Sensor relationships shift with operating phase. Changes are minimal at the begining, accurately localized when anomaly occurs, and broadest after anomaly happens, highlighting DTD's tracking capability of anomaly-driven sensor dynamics.

Such non-static dependencies motivate our adaptive, data-driven graph learning (Section 3). Rather than fixing the graph, DTD learns sensor relationships via GNNs and captures both spatial (inter-sensor) and temporal dependencies, thereby improving sensitivity to subtle faults and providing a comprehensive view of the UAV's state.

5.3. Comparative Energy Assignment Analysis of DM-NP and DM-P Branches

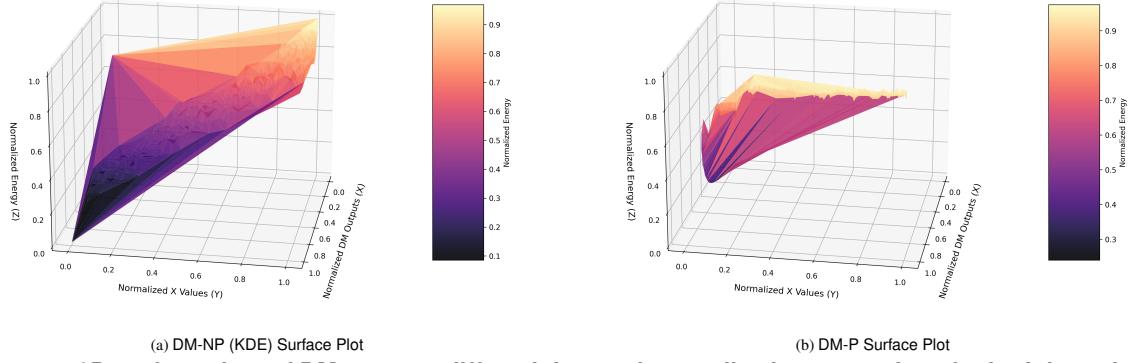(a) DM-NP (KDE) Surface Plot

(b) DM-P Surface Plot

Figure 4   **3D surface plots of DM outputs, diffused data and normalized energy values for both branches on ALFA dataset: (a) DM-NP (KDE) and (b) DM-P. The *x*-axis represents noise prediction outputs, the *y*-axis represents diffused values (0 to 1), and the *z*-axis represents energy outputs.**

Figure 4 compares DM-NP (KDE) and DM-P via 3D surfaces with diffusion level on the *x*-axis, noise-prediction output on the *y*-axis, and energy on the *z*-axis (higher implies more likely anomalous). DM-NP shows a sharp high-energy peak at large diffusion with elevated noise outputs, while DM-P rises more gradually and spreads energy across a wider range. Notably, for DM-NP, when no noise is added ($y = 0$), some diffusion outputs are non-zero. By contrast, DM-P produces near zero diffusion outputs in this case, but DM-NP's assigned energy remains near zero, correctly identifying normal data. This highlights that DM outputs alone are insufficient for anomaly detection, emphasizing the necessity of the scoring mechanism. Both branches effectively assign high energy to diffused anomalous data, with DM-NP's concentrated peak and DM-P's wider distribution showcasing their complementary strengths.

6.   Evaluation on Additional Modalities and More Visualizations

To assess the versatility of the DTD framework beyond UAV sensor data, we evaluate its performance on two distinct modalities: multivariate time series data from SMD and image data from the CIFAR-10 dataset. We also provide additional plots in Appendix H that visualizes DM-P (EBM) on SMD,

Table 2   Performance on SMD dataset (multivariate time series). Reported values are the mean ± standard deviation over 5 independent runs.

Method	Machine 1-1			Machine 2-1		
	Prec.	Rec.	F1	Prec.	Rec.	F1
DM-NP (KDE)	**0.9947**±0.0034	**0.9993**±0.0011	**0.9970**±0.0016	0.9821±0.0056	**0.9945**±0.0031	**0.9883**±0.0025
DM-NP (KNN)	**0.9975**±0.0025	**0.9979**±0.0012	**0.9977**±0.0008	0.9702±0.0123	**0.9932**±0.0000	**0.9815**±0.0063
DM-P	**0.9965**±0.0018	**0.9996**±0.0007	**0.9980**±0.0008	0.9815±0.0064	**0.9925**±0.0045	**0.9869**±0.0027
TranAD (Tuli et al. 2022)	0.9026	0.9974	0.9476	**1.0000**	0.3744	0.5448
D3R (Wang et al. 2023)	0.8589	0.9935	0.9213	0.7482	0.9790	0.8482

Table 3   Performance on CIFAR-10 dataset (images). Reported values are the mean ± standard deviation over 5 independent runs.

Method	Prec.	Acc.	F1
DM-NP (iForest)	**0.9948**±0.0004	0.9593±0.0026	**0.9693**±0.0017
DM-P	**0.9942**±0.0002	0.9547±0.0013	**0.9665**±0.0012
NeuTraL-F (Qiu et al. 2025)	–	0.9530±0.0600	–
PANDA (Reiss et al. 2021)	–	**0.9620**	–

covering the PCA of the raw data, one-step diffusion, the energy gradient at $k = 1$, and example trajectories of normal and anomalous points. These plots collectively demonstrate our framework's ability to distinguish normal from anomalous data.

6.1.   Evaluation on Multivariate Time Series Data

Unlike ALFA/BASiC, which we model as graphs, SMD is evaluated as raw multivariate time series, demonstrating DTD's adaptability without graph construction. Against recent time-series baselines TranAD (Tuli et al. 2022) and D3R (Wang et al. 2023), DTD achieves near-perfect metrics and consistently outperforms them (as shown in Table 2). TranAD sometimes excels in precision but at the cost of recall and F1-Score. Further visualizations of anomaly scores and predictions are provided in Appendix G.

6.2.   Evaluation on Image Data

The CIFAR-10 dataset (Krizhevsky 2009) evaluates DTD for image anomaly detection in a one-vs-rest setup. Table 3 compares our parametric (DM-P) and nonparametric (DM-NP) branches with NeuTraL-F (Qiu et al. 2025) and PANDA (Reiss et al. 2021). DTD attains the best overall performance, with DM-NP yielding the top F1-Score and precision, and DM-P closely matching PANDA while surpassing NeuTraL-F. Unlike PANDA, which requires computationally intensive kNN-based detection at test time and storage of the full training dataset, DTD leverages efficient single-pass diffusion model evaluations, enabling rapid inference suitable for real-time applications,

as described in Section 3 (Reiss et al. 2021). These results complement our structured-data findings and underscore DTD's cross-modal effectiveness.

7. Conclusion

Anomaly detection is critical for safety and reliability in complex systems such as UAVs and industrial monitoring. We introduced DTD, a diffusion based framework that detects anomalies across UAV sensor data, multivariate time series, and images. The framework offers two scoring branches: parametric (DM-P) for efficiency and nonparametric (DM-NP) for interpretability, and our theory justifies using predicted noise as a principled anomaly signal. Across ALFA, BASiC, SMD, and CIFAR 10, DTD outperforms strong baselines and remains robust to high dimensional data and complex dependencies. DTD is practical for real systems, and future work will extend it to richer data structures and broader safety critical deployments.

References

Ahmad MW, Akram MU, Mohsan MM, Saghar K, Ahmad R, Butt WH (2024) Transformer-based sensor failure prediction and classification framework for UAVs. *Expert Systems with Applications* 248:123415, ISSN 09574174, URL http://dx.doi.org/10.1016/j.eswa.2024.123415.

Bai L, Yao L, Li C, Wang X, Wang C (2020) Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. URL http://arxiv.org/abs/2007.02842.

Ben-Gal I, Bacher M, Amara M, Shmueli E (2023) A Nonparametric Subspace Analysis Approach with Application to Anomaly Detection Ensembles. *INFORMS Journal on Data Science* 2(2):99–115, ISSN 2694-4022, 2694-4030, URL http://dx.doi.org/10.1287/ijds.2023.0027.

Chen H, Lyu Y, Shi J, Zhang W (2024) UAV Anomaly Detection Method Based on Convolutional Autoencoder and Support Vector Data Description with 0/1 Soft-Margin Loss. *Drones* 8(10):534, ISSN 2504-446X, URL http://dx.doi.org/10.3390/drones8100534.

Deng H, Lu Y, Tao Y, Liu Z, Chen J (2024) Unmanned Aerial Vehicles anomaly detection model based on sensor information fusion and hybrid multimodal neural network. *Engineering Applications of Artificial Intelligence* 132:107961, URL http://dx.doi.org/10.1016/j.engappai.2024.107961.

Dhakal R, Bosma C, Chaudhary P, Kandel LN (2023) UAV Fault and Anomaly Detection Using Autoencoders. *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, 1–8, ISSN 2155-7209, URL http://dx.doi.org/10.1109/DASC58513.2023.10311126.

Efron B (2011) Tweedie's Formula and Selection Bias. *Journal of the American Statistical Association* 106(496):1602–1614, ISSN 0162-1459, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3325056/.

He M, Wei Z, Wen JR (2022) Convolutional Neural Networks on Graphs with Chebyshev Approximation, Revisited. *The Thirty-Sixth Annual Conference on Neural Information Processing Systems* .

Ho J, Jain A, Abbeel P (2020) Denoising Diffusion Probabilistic Models. *NeurIPS* (arXiv), URL `http://arxiv.org/abs/2006.11239`.

Ho TKK, Karami A, Armanfard N (2025) Graph Anomaly Detection in Time Series: A Survey. URL `http://dx.doi.org/10.48550/arXiv.2302.00058`.

Hu W, Gao J, Li B, Wu O, Du J, Maybank S (2020) Anomaly Detection Using Local Kernel Density Estimation and Context-Based Regression. *IEEE Transactions on Knowledge and Data Engineering* 32(2):218–233, ISSN 1041-4347, 1558-2191, 2326-3865, URL `http://dx.doi.org/10.1109/TKDE.2018.2882404`.

Huang H, Wang P, Pei J, Wang J, Alexanian S, Niyato D (2025) Deep Learning Advancements in Anomaly Detection: A Comprehensive Survey. URL `http://dx.doi.org/10.48550/arXiv.2503.13195`.

Huet A, Navarro JM, Rossi D (2022) Local Evaluation of Time Series Anomaly Detection Algorithms. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 635–645, URL `http://dx.doi.org/10.1145/3534678.3539339`.

Jiang G, Nan P, Zhang J, Li Y, Li X (2024) Robust Spatial-Temporal Autoencoder for Unsupervised Anomaly Detection of Unmanned Aerial Vehicle With Flight Data. *IEEE Transactions on Instrumentation and Measurement* 73:1–14, ISSN 0018-9456, 1557-9662, URL `http://dx.doi.org/10.1109/TIM.2024.3428649`.

Keipour A, Mousaei M, Scherer S (2019) Automatic Real-time Anomaly Detection for Autonomous Aerial Vehicles. *2019 International Conference on Robotics and Automation (ICRA)*, 5679–5685, URL `http://dx.doi.org/10.1109/ICRA.2019.8794286`.

Keipour A, Mousaei M, Scherer S (2021) ALFA: A dataset for UAV fault and anomaly detection. *The International Journal of Robotics Research* 40(2-3):515–520, ISSN 0278-3649, 1741-3176, URL `http://dx.doi.org/10.1177/0278364920966642`.

Kim H, Kim H (2023) Contextual anomaly detection for high-dimensional data using Dirichlet process variational autoencoder. *IISE Transactions* 55(5):433–444, ISSN 2472-5854, 2472-5862, URL `http://dx.doi.org/10.1080/24725854.2021.2024925`.

Krizhevsky A (2009) Learning Multiple Layers of Features from Tiny Images .

Lin H, Liu G, Wu J, Zhao JL (2024) Deterring the Gray Market: Product Diversion Detection via Learning Disentangled Representations of Multivariate Time Series. *INFORMS Journal on Computing* 36(2):571–586, ISSN 1091-9856, URL `http://dx.doi.org/10.1287/ijoc.2022.0155`.

Lin R, Khalastchi E, Kaminka GA (2010) Detecting anomalies in unmanned vehicles using the Mahalanobis distance. *2010 IEEE International Conference on Robotics and Automation*, 3038–3044 (Anchorage, AK: IEEE), ISBN 978-1-4244-5038-1, URL `http://dx.doi.org/10.1109/ROBOT.2010.5509781`.

Liu F, Zhu X, Feng P, Zeng L (2023) Anomaly Detection via Progressive Reconstruction and Hierarchical Feature Fusion. *Sensors* 23(21):8750, ISSN 1424-8220, URL `http://dx.doi.org/10.3390/s23218750`.

Livernoche V, Jain V, Hezaveh Y, Ravanbakhsh S (2023) On Diffusion Modeling for Anomaly Detection. *ICLR 2024*, 31, URL `https://openreview.net/forum?id=lR3rk7ysXz`.

Ma X, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Xiong H, Akoglu L (2023) A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *IEEE Transactions on Knowledge and Data Engineering* 35(12):12012–12038, ISSN 1041-4347, 1558-2191, 2326-3865, URL http://dx.doi.org/10.1109/TKDE.2021.3118815.

Qiu C, Kloft M, Mandt S, Rudolph M (2025) Self-Supervised Anomaly Detection With Neural Transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47(3):2170–2185, ISSN 1939-3539, URL http://dx.doi.org/10.1109/TPAMI.2024.3519543.

Reiss T, Cohen N, Bergman L, Hoshen Y (2021) PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2805–2813 (Nashville, TN, USA: IEEE), ISBN 978-1-6654-4509-2, URL http://dx.doi.org/10.1109/CVPR46437.2021.00283.

Shakhatreh H, Sawalmeh AH, Al-Fuqaha A, Dou Z, Almaita E, Khalil I, Othman NS, Khreishah A, Guizani M (2019) Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* 7:48572–48634, ISSN 2169-3536, URL http://dx.doi.org/10.1109/ACCESS.2019.2909530.

Suboh S, Aziz IA, Shaharudin SM, Ismail SA, Mahdin H (2023) A Systematic Review of Anomaly Detection within High Dimensional and Multivariate Data. *JOIV : International Journal on Informatics Visualization* 7(1):122, ISSN 2549-9904, 2549-9610, URL http://dx.doi.org/10.30630/joiv.7.1.1297.

Tao C, Du J (2025) PointSGRADE: Sparse learning with graph representation for anomaly detection by using unstructured 3D point cloud data. *IISE Transactions* 57(2):131–144, ISSN 2472-5854, 2472-5862, URL http://dx.doi.org/10.1080/24725854.2023.2285840.

Tuli S, Casale G, Jennings NR (2022) TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *VLDB 2022* (arXiv), URL http://dx.doi.org/10.48550/arXiv.2201.07284.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, eds., *Advances in Neural Information Processing Systems*, volume 30 (Curran Associates, Inc.), URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang B, Liu D, Peng X, Wang Z (2019) Data-Driven Anomaly Detection of UAV based on Multimodal Regression Model. *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 1–6, ISSN 2642-2077, URL http://dx.doi.org/10.1109/I2MTC.2019.8827154.

Wang C, Zhuang Z, Qi Q, Wang J, Wang X, Sun H, Liao J (2023) Drift doesn't Matter: Dynamic Decomposition with Diffusion Reconstruction for Unstable Multivariate Time Series Anomaly Detection. *NeurIPS 2023*, 17.

Xu J, Wang J, Zhou Z, Lu T (2025) Toward Graph Data Collaboration in a Data-Sharing-Free Manner: A Novel Privacy-Preserving Graph Pretraining Model. *INFORMS Journal on Computing* ijoc.2023.0115, ISSN 1091-9856, 1526-5528, URL http://dx.doi.org/10.1287/ijoc.2023.0115.

Yang L, Li S, Zhang Y, Zhu C, Liao Z (2024) Deep Learning-Assisted Unmanned Aerial Vehicle Flight Data Anomaly Detection: A Review. *IEEE Sensors Journal* 24(20):31681–31695, ISSN 1558-1748, URL `http://dx.doi.org/10.1109/JSEN.2024.3451648`.

Yu Y, Lv P, Tong X, Dong J (2020) Anomaly Detection in High-Dimensional Data Based on Autoregressive Flow. Nah Y, Cui B, Lee SW, Yu JX, Moon YS, Whang SE, eds., *Database Systems for Advanced Applications*, volume 12113, 125–140 (Cham: Springer International Publishing), ISBN 978-3-030-59415-2 978-3-030-59416-9, URL `http://dx.doi.org/10.1007/978-3-030-59416-9_8`.

Zhang H (2024) An attempt to generate new bridge types from latent space of generative flow. URL `http://dx.doi.org/10.48550/arXiv.2401.10299`, arXiv preprint.

Zhou S, He Z, Chen X, Chang W (2024) An Anomaly Detection Method for UAV Based on Wavelet Decomposition and Stacked Denoising Autoencoder. *Aerospace* 11(5):393, ISSN 2226-4310, URL `http://dx.doi.org/10.3390/aerospace11050393`.

Appendix A: Derivation of Noise Prediction as Scaled Score Function

This appendix provides the detailed proof of Proposition 1, which states that the predicted noise $\epsilon_\theta(x_k, k, x_{hist})$ in a diffusion model, conditioned on historical data x_{hist}, approximates a scaled score function, $\epsilon_\theta(x_k, k, x_{hist}) \approx -\sqrt{1 - \bar{\alpha}_k} \nabla_{x_k} \log p_k(x_k | x_{hist})$, and at small k, $\epsilon_\theta(x_0, 0, x_{hist}) \approx -\sqrt{1 - \bar{\alpha}_0} \nabla_{x_0} \log p_{\text{data}}(x_0 | x_{hist})$. The proof leverages the forward process of the diffusion model, Tweedie's formula, and the training objective to establish this relationship, justifying the use of predicted noise for anomaly detection in contexts where historical data informs the model.

We begin with the forward process defined in Equation (1) as follows:

$$x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1 - \bar{\alpha}_k} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

where $x_0 \sim p_{\text{data}}(\cdot | x_{hist})$, $\bar{\alpha}_k = \prod_{s=0}^{k}(1 - \beta_s)$, and $\beta_s \in (0, 1)$ is the noise schedule. The data distribution $p_{\text{data}}(x_0 | x_{hist})$ is conditioned on historical data x_{hist}, reflecting the temporal context of the system, such as past UAV sensor readings. The neural network $\epsilon_\theta(x_k, k, x_{hist})$ is trained to predict ϵ, minimizing the objective in Equation (3) as follows:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0 \sim p_{\text{data}}(\cdot | x_{hist}), \epsilon \sim \mathcal{N}(0, I), k} \left[\| \epsilon - \epsilon_\theta(x_k, k, x_{hist}) \|_2^2 \right].$$

To show that $\epsilon_\theta(x_k, k, x_{hist})$ approximates a scaled score, we first derive the conditional expectation of the noise given x_k and x_{hist}. Rearranging Equation (1) for x_0:

$$x_0 = \frac{x_k - \sqrt{1 - \bar{\alpha}_k} \epsilon}{\sqrt{\bar{\alpha}_k}}. \tag{10}$$

Taking the conditional expectation given the fixed x_k and x_{hist}:

$$
\begin{aligned}
\mathbb{E}[x_0 | x_k, x_{hist}] &= \mathbb{E}\left[\left. \frac{x_k - \sqrt{1 - \bar{\alpha}_k} \epsilon}{\sqrt{\bar{\alpha}_k}} \right| x_k, x_{hist} \right] \\
&= \frac{x_k}{\sqrt{\bar{\alpha}_k}} - \frac{\sqrt{1 - \bar{\alpha}_k}}{\sqrt{\bar{\alpha}_k}} \mathbb{E}[\epsilon | x_k, x_{hist}],
\end{aligned}
\tag{11}
$$

Solving for $\mathbb{E}[\epsilon | x_k, x_{hist}]$:

$$\mathbb{E}[\epsilon | x_k, x_{hist}] = \frac{x_k - \sqrt{\bar{\alpha}_k} \mathbb{E}[x_0 | x_k, x_{hist}]}{\sqrt{1 - \bar{\alpha}_k}}. \tag{12}$$

The optimal noise predictor, minimizing the expected squared error, is $\epsilon_\theta^*(x_k, k, x_{hist}) = \mathbb{E}[\epsilon | x_k, x_{hist}]$. Thus, the training objective ensures $\epsilon_\theta(x_k, k, x_{hist}) \approx \epsilon_\theta^*(x_k, k, x_{hist})$ for normal data conditioned on x_{hist}.

Next, we connect $\mathbb{E}[\epsilon | x_k, x_{hist}]$ to the score function $\nabla_{x_k} \log p_k(x_k | x_{hist})$ using Tweedie's formula (Ho et al. 2020). For a Gaussian perturbation $x = z + \sigma \epsilon$, Tweedie's formula states:

$$\nabla_x \log p(x) = \frac{1}{\sigma^2} \left(\mathbb{E}[z | x] - x \right). \tag{13}$$

In our model, $x_k = \sqrt{\bar{\alpha}_k} x_0 + \sqrt{1 - \bar{\alpha}_k} \epsilon$, where $z = \sqrt{\bar{\alpha}_k} x_0$ and $\sigma = \sqrt{1 - \bar{\alpha}_k}$. Conditioning on x_{hist}, we apply Tweedie's formula to the conditional distribution $p_k(x_k | x_{hist})$:

$$\nabla_{x_k} \log p_k(x_k|x_{hist}) = \frac{1}{1-\bar{\alpha}_k} \left(\mathbb{E}[\sqrt{\bar{\alpha}_k}x_0|x_k, x_{hist}] - x_k \right). \tag{14}$$

As $\mathbb{E}[\sqrt{\bar{\alpha}_k}x_0|x_k, x_{hist}] = \sqrt{\bar{\alpha}_k}\mathbb{E}[x_0|x_k, x_{hist}]$, we have:

$$\nabla_{x_k} \log p_k(x_k|x_{hist}) = \frac{1}{1-\bar{\alpha}_k} \left(\sqrt{\bar{\alpha}_k}\mathbb{E}[x_0|x_k, x_{hist}] - x_k \right). \tag{15}$$

From Equation (12), compute:

$$x_k - \sqrt{\bar{\alpha}_k}\mathbb{E}[x_0|x_k, x_{hist}] = \sqrt{1-\bar{\alpha}_k}\mathbb{E}[\epsilon|x_k, x_{hist}]$$
$$= \sqrt{1-\bar{\alpha}_k}\epsilon_\theta^*(x_k, k, x_{hist}). \tag{16}$$

Substitute into the score function:

$$\nabla_{x_k} \log p_k(x_k|x_{hist}) = \frac{1}{1-\bar{\alpha}_k} \left(-\sqrt{1-\bar{\alpha}_k}\epsilon_\theta^*(x_k, k, x_{hist}) \right)$$
$$= -\frac{\epsilon_\theta^*(x_k, k, x_{hist})}{\sqrt{1-\bar{\alpha}_k}}. \tag{17}$$

Thus:

$$\epsilon_\theta^*(x_k, k, x_{hist}) = -\sqrt{1-\bar{\alpha}_k}\nabla_{x_k} \log p_k(x_k|x_{hist}). \tag{18}$$

Since $\epsilon_\theta(x_k, k, x_{hist}) \approx \epsilon_\theta^*(x_k, k, x_{hist})$ due to training, we obtain:

$$\epsilon_\theta(x_k, k, x_{hist}) \approx -\sqrt{1-\bar{\alpha}_k}\nabla_{x_k} \log p_k(x_k|x_{hist}). \tag{19}$$

For small k, such as $k = 1$, $\bar{\alpha}_0 \approx 1$, so $x_0 \approx x_k$, and $p_k(x_k|x_{hist}) \approx p_{\text{data}}(x_0|x_{hist})$. Hence:

$$\epsilon_\theta(x_0, 0, x_{hist}) \approx -\sqrt{1-\bar{\alpha}_0}\nabla_{x_0} \log p_{\text{data}}(x_0|x_{hist}). \tag{20}$$

This completes the first part of Proposition 1, showing that $\epsilon_\theta(x_k, k, x_{hist})$ encodes the geometry of the conditional data distribution $p_k(x_k|x_{hist})$ via the score function.

To validate consistency, we examine the training objective in Equation (3). Substituting $\epsilon = \frac{x_k - \sqrt{\bar{\alpha}_k}x_0}{\sqrt{1-\bar{\alpha}_k}}$ from Equation (1) into the loss:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0 \sim p_{\text{data}}(\cdot|x_{hist}), \epsilon \sim \mathcal{N}(0, I), k} \left[\left\| \frac{x_k - \sqrt{\bar{\alpha}_k}x_0}{\sqrt{1-\bar{\alpha}_k}} - \epsilon_\theta(x_k, k, x_{hist}) \right\|_2^2 \right]. \tag{21}$$

Taking expectations over ϵ, the optimal $\epsilon_\theta(x_k, k, x_{hist})$ matches $\mathbb{E}[\epsilon|x_k, x_{hist}]$. From Equations (12) and (19), the expected noise $\mathbb{E}[\epsilon|x_k, x_{hist}] = -\sqrt{1-\bar{\alpha}_k}\nabla_{x_k} \log p_k(x_k|x_{hist})$ confirms that the training aligns $\epsilon_\theta(x_k, k, x_{hist})$ with the scaled negative score conditioned on x_{hist}. This resembles denoising score matching, where the conditional score $\nabla_{x_k} \log p_k(x_k|x_{hist})$ is approximated, reinforcing the theoretical foundation.

Thus, Proposition 1 holds, as $\epsilon_\theta(x_k, k, x_{hist})$ reliably approximates the scaled score of the conditional distribution, enabling anomaly detection by identifying deviations from the learned data distribution given historical context.

Appendix B: Derivation of EBM Scoring Likelihood

This appendix derives the diffusion-based likelihood used in Proposition 2 and shows that the EBM energy $E_\phi(\hat{\epsilon})$ approximates the negative log-likelihood $-\log p_{\theta,\phi}(x \mid x_{\text{hist}})$ for anomaly detection. Here $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$. The derivation integrates the diffusion model's noise prediction, conditioned on historical data x_{hist}, with the EBM's energy function, providing a probabilistic justification for the parametric scoring mechanism.

We define the diffusion-based likelihood as a marginal over the noise ϵ:

$$p_{\theta,\phi}(x|x_{hist}) = \int p(\epsilon) p_{\theta,\phi}(x|\epsilon, x_{hist}) \, d\epsilon, \tag{22}$$

where $p(\epsilon) = \mathcal{N}(\epsilon|0, I)$ is the noise prior, and $p_{\theta,\phi}(x|\epsilon, x_{hist})$ is the conditional likelihood. For a test sample x, the noisy sample at time step k is based on the forward diffusion process in Equation (1) as follows:

$$x_k = \sqrt{\bar{\alpha}_k} x + \sqrt{1 - \bar{\alpha}_k} \epsilon,$$

and the diffusion model predicts $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist})$, conditioned on x_{hist}. We model the conditional likelihood using the EBM in Equation (8) which is equivalent to:

$$p_{\theta,\phi}(x|\epsilon, x_{hist}) = \frac{1}{Z_{\theta,\phi}(\epsilon, x_{hist})} \exp\left(-\tilde{E}_{\theta,\phi}(x, \epsilon, x_{hist})\right),$$

with the energy function:

$$\tilde{E}_{\theta,\phi}(x, \epsilon, x_{hist}) = E_\phi(\hat{\epsilon}) + \frac{1}{2\sigma^2} \|\hat{\epsilon} - \epsilon\|_2^2, \tag{23}$$

where $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist})$, σ^2 is a variance hyperparameter, and $Z_{\theta,\phi}(\epsilon, x_{hist}) = \int \exp\left(-\tilde{E}_{\theta,\phi}(x', \epsilon, x_{hist})\right) dx'$ is the normalization constant. The term $\frac{1}{2\sigma^2} \|\hat{\epsilon} - \epsilon\|_2^2$ ensures $\hat{\epsilon} \approx \epsilon$ for normal data, while $E_\phi(\hat{\epsilon})$ penalizes deviations from normality given x_{hist}.

Substituting into the marginal likelihood:

$$p_{\theta,\phi}(x|x_{hist}) = \int \mathcal{N}(\epsilon|0, I) \cdot \frac{1}{Z_{\theta,\phi}(\epsilon, x_{hist})}$$
$$\cdot \exp\left(-E_\phi(\hat{\epsilon}) - \frac{1}{2\sigma^2} \|\hat{\epsilon} - \epsilon\|_2^2\right) d\epsilon. \tag{24}$$

The integral is intractable because the partition function $Z_{\theta,\phi}(\epsilon, x_{\text{hist}})$ depends on ϵ and x_{hist}, and this dependence interacts with the nonlinear mappings $\hat{\epsilon} = \epsilon_\theta(x_k, k, x_{\text{hist}})$ and $E_\phi(\hat{\epsilon})$. We approximate it using a single Monte Carlo sample $\epsilon \sim \mathcal{N}(0, I)$:

$$\log p_{\theta,\phi}(x|x_{hist}) \approx \log \mathcal{N}(\epsilon|0, I) - E_\phi(\hat{\epsilon})$$
$$- \frac{1}{2\sigma^2} \|\hat{\epsilon} - \epsilon\|_2^2 - \log Z_{\theta,\phi}(\epsilon, x_{hist}). \tag{25}$$

Since $\log \mathcal{N}(\epsilon|0, I) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \|\epsilon\|_2^2$, we have:

$$\log p_{\theta,\phi}(x|x_{hist}) \approx -\frac{d}{2} \log(2\pi) - \frac{1}{2} \|\epsilon\|_2^2 - E_\phi(\hat{\epsilon})$$
$$- \frac{1}{2\sigma^2} \|\hat{\epsilon} - \epsilon\|_2^2 - \log Z_{\theta,\phi}(\epsilon, x_{hist}). \tag{26}$$

For normal data conditioned on x_{hist}, $\hat{\epsilon} \approx \epsilon$ (from diffusion model training), so $\|\hat{\epsilon} - \epsilon\|_2^2 \approx 0$, and $E_\phi(\hat{\epsilon})$ is small due to EBM training. For anomalies, $\hat{\epsilon}$ deviates, increasing both the squared L2 norm $\|\hat{\epsilon} - \epsilon\|_2^2$ and the energy score $E_\phi(\hat{\epsilon})$. Since $Z_{\theta,\phi}(\epsilon, x_{hist})$ is difficult to compute and relatively constant across samples, we focus on the dominant term for anomaly scoring:

$$\log p_{\theta,\phi}(x|x_{hist}) \approx -E_\phi(\hat{\epsilon}) + \text{const}, \tag{27}$$

yielding the anomaly score:

$$s(x) = E_\phi(\hat{\epsilon}), \quad \hat{\epsilon} = \epsilon_\theta(x_k, k, x_{hist}). \tag{28}$$

To ensure $E_\phi(\hat{\epsilon})$ distinguishes normal from anomalous data, the EBM is trained with the contrastive loss:

$$\mathcal{L}_{\text{EBM}} = \mathbb{E}_{x \sim p_{\text{data}}(\cdot|x_{hist})} \left[E_\phi(\hat{\epsilon}^+) \right] - \mathbb{E}_{\hat{\epsilon}^- \sim p_{\text{neg}}} \left[E_\phi(\hat{\epsilon}^-) \right], \tag{29}$$

where $\hat{\epsilon}^+$ comes from normal data conditioned on x_{hist}, and $\hat{\epsilon}^-$ from anomalous synthetic samples via the forward diffusion. This validates Proposition 2 by showing that $E_\phi(\hat{\epsilon})$ is a proxy for $-\log p_{\theta,\phi}(x \mid x_{\text{hist}})$ and the high scores indicate anomalies.

Appendix C: Justification for One-Step Diffusion and Noise Prediction in Anomaly Detection

This appendix justifies the use of diffusing the original data for one time step and predicting the noise in our diffusion-based anomaly detection framework, conditioned on historical data x_{hist}. We combine empirical evidence from prior work on diffusion models with a mathematical derivation to demonstrate its advantages over direct noise prediction from the original data, enhancing our justifications.

The previous studies on diffusion models (Ho et al. 2020) suggest that predicting the noise added during the diffusion process outperforms other parameterizations, such as predicting the mean of the reverse process or the original data. This approach aligns with generative modeling principles and improves the quality of generated samples. For anomaly detection, accurately modeling the normal conditional $p(x \mid x_{hist})$ is also key to detecting deviations. The empirical advantage therefore supports predicting noise rather than alternative targets.

Considering a test sample x, we diffuse x for one time step ($k = 1$) using the forward process in Equation (1) as follows:

$$x_1 = \sqrt{\bar{\alpha}_1} x + \sqrt{1 - \bar{\alpha}_1} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

where $\bar{\alpha}_1 = 1 - \beta_1$, and β_1 is a small noise variance. The diffusion model, parameterized by ϵ_θ, predicts the noise $\hat{\epsilon} = \epsilon_\theta(x_1, 1, x_{hist})$, conditioned on x_{hist}, and is trained on normal data $x_0 \sim p_{\text{data}}(\cdot|x_{hist})$ to minimize Equation (3) as follows:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x_0 \sim p_{\text{data}}(\cdot|x_{hist}), \epsilon \sim \mathcal{N}(0, I), k} \left[\|\epsilon - \epsilon_\theta(x_k, k, x_{hist})\|_2^2 \right].$$

Our goal is to detect anomalies by analyzing $\hat{\epsilon}$, leveraging its connection to the conditional data distribution.

First, consider predicting noise directly from the original data without diffusion ($k = 0$). Here, $x_0 = x$, and $\bar{\alpha}_0 = 1$, so no noise is added. The predicted noise becomes:

$$\epsilon_\theta(x_0, 0, x_{hist}) \approx -\sqrt{1 - \bar{\alpha}_0} \nabla_{x_0} \log p_0(x_0|x_{hist})$$
$$= 0 \cdot \nabla_{x_0} \log p_{\text{data}}(x|x_{hist}) = 0, \tag{30}$$

as derived in Proposition 1. This result is trivial and independent of x, offering no discriminative power for anomaly detection. Thus, direct prediction fails to provide meaningful information about the data's conformity to $p_{\text{data}}(x|x_{hist})$.

Now, diffusing for one step ($k = 1$) introduces a small perturbation. For normal data $x \sim p_{\text{data}}(\cdot|x_{hist})$, the model is trained such that $\epsilon_\theta(x_1, 1, x_{hist}) \approx \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Define the predicted noise distribution for normal data:

$$p_{\hat{\epsilon}}^{\text{normal}}(\hat{\epsilon}) = \mathbb{E}_{x \sim p_{\text{data}}(\cdot|x_{hist}), \epsilon \sim \mathcal{N}(0,I)} \left[\delta(\hat{\epsilon} - \epsilon_\theta(x_1, 1, x_{hist})) \right]. \tag{31}$$

Since ϵ_θ minimizes the mean squared error to ϵ, $p_{\hat{\epsilon}}^{\text{normal}}(\hat{\epsilon}) \approx \mathcal{N}(0, I)$. For anomalous data $x \nsim p_{\text{data}}(\cdot|x_{hist})$, x_1 lies outside the manifold learned during training, and $\epsilon_\theta(x_1, 1, x_{hist})$ deviates from $\mathcal{N}(0, I)$, reflecting a mismatch with the expected noise pattern given x_{hist}.

To formalize this, consider the expected norm of the predicted noise. For normal data:

$$\mathbb{E}_{x \sim p_{\text{data}}(\cdot|x_{hist})} \left[\|\epsilon_\theta(x_1, 1, x_{hist})\|_2^2 \right] \approx \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[\|\epsilon\|_2^2 \right] = d, \tag{32}$$

where d is the data dimension. For anomalous data, let $x \sim p_{\text{anom}}$. Since ϵ_θ is not trained on p_{anom}, $\hat{\epsilon}$ may exhibit larger variance or bias. Suppose $\epsilon_\theta(x_1, 1, x_{hist}) \sim \mathcal{N}(\mu_{\text{anom}}, \Sigma_{\text{anom}})$, where $\mu_{\text{anom}} \neq 0$ or $\Sigma_{\text{anom}} \neq I$. Then:

$$\mathbb{E}_{x \sim p_{\text{anom}}} \left[\|\epsilon_\theta(x_1, 1, x_{hist})\|_2^2 \right] = \text{tr}(\Sigma_{\text{anom}}) + \|\mu_{\text{anom}}\|_2^2, \tag{33}$$

which typically exceeds d due to distributional mismatch. This deviation enables anomaly detection by thresholding $\|\hat{\epsilon}\|_2^2$.

From Proposition 1, $\epsilon_\theta(x_1, 1, x_{hist}) \approx -\sqrt{1 - \bar{\alpha}_1} \nabla_{x_1} \log p_1(x_1|x_{hist})$. For small k, $p_1(x_1|x_{hist}) \approx p_{\text{data}}(x|x_{hist})$, so:

$$\hat{\epsilon} \approx -\sqrt{1 - \bar{\alpha}_1} \nabla_x \log p_{\text{data}}(x|x_{hist}). \tag{34}$$

The score function's magnitude is larger in low-density regions, which are typical for anomalies. Thus, $\|\hat{\epsilon}\|_2$ is larger for anomalous x, providing a theoretical basis for detection. In practice, we use the full distribution of $\hat{\epsilon}$ for robustness. Diffusing for one step and predicting noise conditioned on x_{hist} offers:

1. Non-trivial signal: Unlike $k = 0$, where $\hat{\epsilon} = 0$, $k = 1$ yields a meaningful $\hat{\epsilon}$ reflecting the conditional data distribution.

2. Distributional sensitivity: $\hat{\epsilon}$ approximates $\mathcal{N}(0, I)$ for normal data but deviates for anomalies, enabling detection given x_{hist}.

3. Score function insight: $\hat{\epsilon}$ encodes the conditional score, linking to density-based anomaly principles.

4. Efficiency: A single diffusion step and prediction suffice, ideal for real-time applications.

Table 4   Fault types in the processed ALFA dataset, detailing test cases and flight time.

Fault Category	Test Flights	Before Fault (s)	With Fault (s)
Engine full power loss	23	2282	362
Rudder stuck to left	1	60	9
Rudder stuck to right	2	107	32
Elevator stuck at zero	2	181	23
Left aileron stuck at zero	3	228	183
Right aileron stuck at zero	4	442	231
Both ailerons stuck at zero	1	66	36
Rudder and aileron at zero & Aileron Zero	1	116	27
Total	47	3935	777

Appendix D: Evaluation Metrics Details

- Precision: Proportion of true positives (TP) among all positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

- Recall: Proportion of TP among all actual positives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

- F1-score: Harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

- Accuracy: Proportion of correctly classified samples:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

To account for the unique nature of UAV faults, such as engine failures that occur once and persist until the flight ends, we adjust these metrics using the affiliation approach (Huet et al. 2022). Classical metrics punish predictions slightly off the fault's start (e.g., a second late) and overvalue long faults, skewing scores. Affiliation metrics measure the time gap between predictions and the fault's start, comparing to a random guess to compute precision and recall. It ensures: (1) predictions close to the fault's onset (e.g., detecting engine failure within seconds) are rewarded; (2) the fault is scored as one event, avoiding bias from its length; and (3) only predictions near the actual fault improve the score, preventing unrelated guesses (e.g., random alerts far from the fault) from falsely boosting results. For image data (CIFAR-10 dataset), to align with baselines like NeuTraL-F and PANDA (Table V), we use accuracy, ensuring fair comparison with established image anomaly detection methods.

Appendix E: Datasets Statistics

Table 4 provides details on the fault types in the processed ALFA dataset, including the number of test cases and flight times before and after faults. Table 5 summarizes the SMD and CIFAR-10 datasets used in our experiments, including the number of training and test samples, dimensions, and anomaly percentages.

Table 5   SMD and CIFAR-10 Datasets.

Dataset	Train	Test	Dimensions	Anomalies (%)
SMD	708,405	708,420	38 (4 traces)	4.16
CIFAR-10	50,000	10,000	3,072 (32×32×3)	90.00 (test)

Table 6   Sensor Descriptions for ALFA Dataset.

Index	Sensor Name	Description
0	mavros_nav_info_errors	Tracking, airspeed, and altitude errors
1	mavros_vfr_hud	Data for HUD (climb, altitude, groundspeed, heading, throttle)
2	mavros_local_position_odom	Local position and odometry (angular/linear velocities, orientation)
3	mavros_imu_data	IMU state (angular velocity, linear acceleration, orientation)
4	mavros_wind_estimation	Wind estimation by FCU (wind speed components)
5	mavros_global_position_raw_gps_vel	GPS-based velocity (linear velocities)
6	mavros_nav_info_velocity	Commanded and measured velocity
7	mavros_rc_out	Remote control outputs (throttle, aileron channels)
8	mavros_global_position_global	Global position info (altitude, longitude, latitude)
9	mavros_imu_mag	Magnetic field components
10	mavros_setpoint_raw_target_global	Setpoint messages (acceleration or force setpoints)
11	mavctrl_path_dev	Path deviation
12	mavros_nav_info_yaw	Commanded and measured yaw
13	mavros_nav_info_pitch	Commanded and measured pitch
14	mavros_global_position_rel_alt	Relative altitude
15	mavctrl_rpy	Measured roll, pitch, and yaw
16	mavros_global_position_compass_hdg	Compass heading
17	mavros_nav_info_roll	Commanded and measured roll
18	mavros_imu_atm_pressure	Atmospheric pressure

Appendix F: Sensor Descriptions fo ALFA Dataset

Table 6 lists the sensors used in the ALFA dataset after we processed the raw data and converted it into a graph strucutre for our experiments. Each sensor is indexed and described, providing insights into the UAV's operational parameters and environmental conditions during flights.

Appendix G: Additional Visualizations of Experiments

Figures 5 and 6 provide additional visualizations of anomaly scores and predictions on the BASIC and SMD datasets from both branches (DM-NP and DM-P). The left column shows scores and the right column shows the corresponding predictions. On BASIC (graph-structured UAV data), we observe zero false positives. On SMD datasets (pure time series), we almost miss no true positives. These results exceed recent baselines reported in the main paper.

Appendix H: Gradient Dynamics and Trajectory Analysis of DM-P

To further evaluate the effectiveness of the DM-P framework, parameterized by an Energy-Based Model (EBM), we present a visualization analysis using the SMD dataset in Figure 7. Figure 7 (a) depicts the original data distribution after PCA, with normal data (blue dots) clustering densely around 0.0 on the *x*-axis and anomalous data (red 'x' markers)
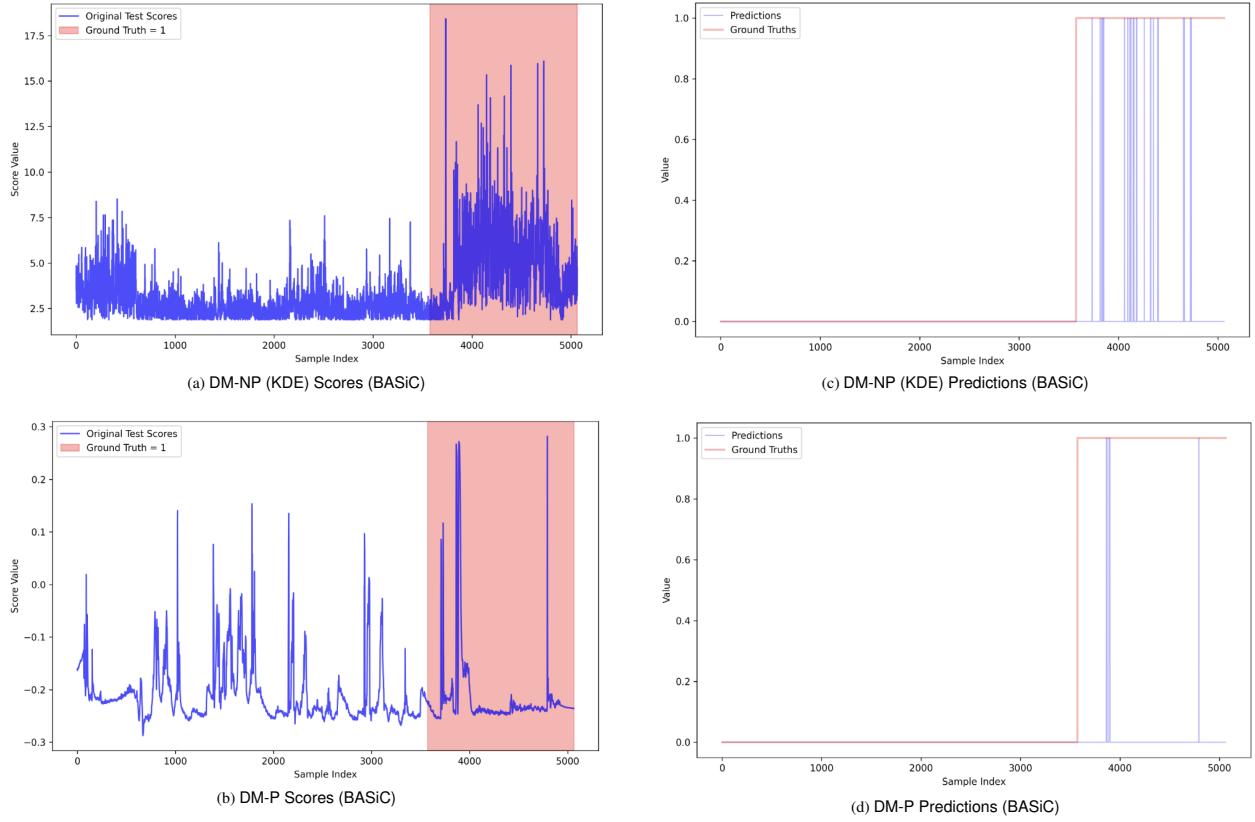
Figure 5 **BASiC: anomaly scores (left) and predictions (right). Ground-truth fault regions are shaded in red.**

scattered along a diagonal from 0.0 to 3.0, with notable overlap in the lower left quadrant that complicates the anomaly detection task. Figure 7 (b) shows the data after one diffusion time step ($k = 1$), where a density plot (purple to yellow gradient) highlights the densest region around 0.0, preserving normal data structure while spreading anomalous data. Figure 7 (c) visualizes the DM-P energy gradient field at $k = 1$, with arrows guiding normal data toward high-density regions and anomalous data along the diagonal in varied directions, maintaining separation. Figure 7 (d) illustrates data trajectories: a blue trajectory from a normal region converges to the high-density area, an orange trajectory from an anomalous region remains stationary, and a green trajectory from another anomalous region near normal data shifts toward the anomalous zone.

(a) DM-NP (KNN) Scores (SMD 2-1)

(e) DM-NP (KNN) Predictions (SMD 2-1)

(b) DM-P Scores (SMD 2-1)

(f) DM-P Predictions (SMD 2-1)

(c) DM-NP (KDE) Scores (SMD 1-1)

(g) DM-NP (KDE) Predictions (SMD 1-1)

(d) DM-P Scores (SMD 1-1)
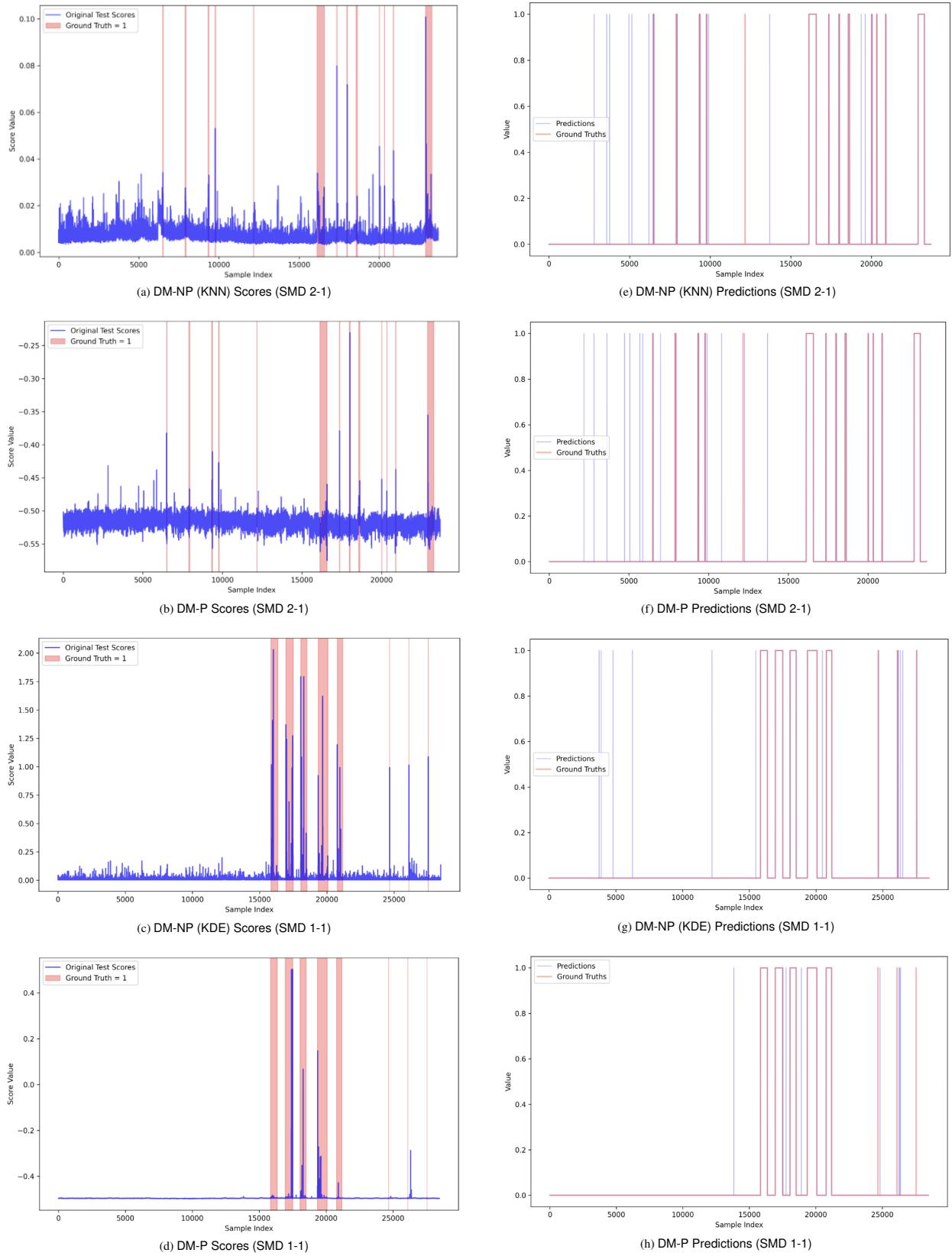
(h) DM-P Predictions (SMD 1-1)

Figure 6   SMD: anomaly scores (left) and predictions (right). Top row: Machine 2-1. Bottom row: Machine 1-1. Ground-truth fault regions are shaded in red.
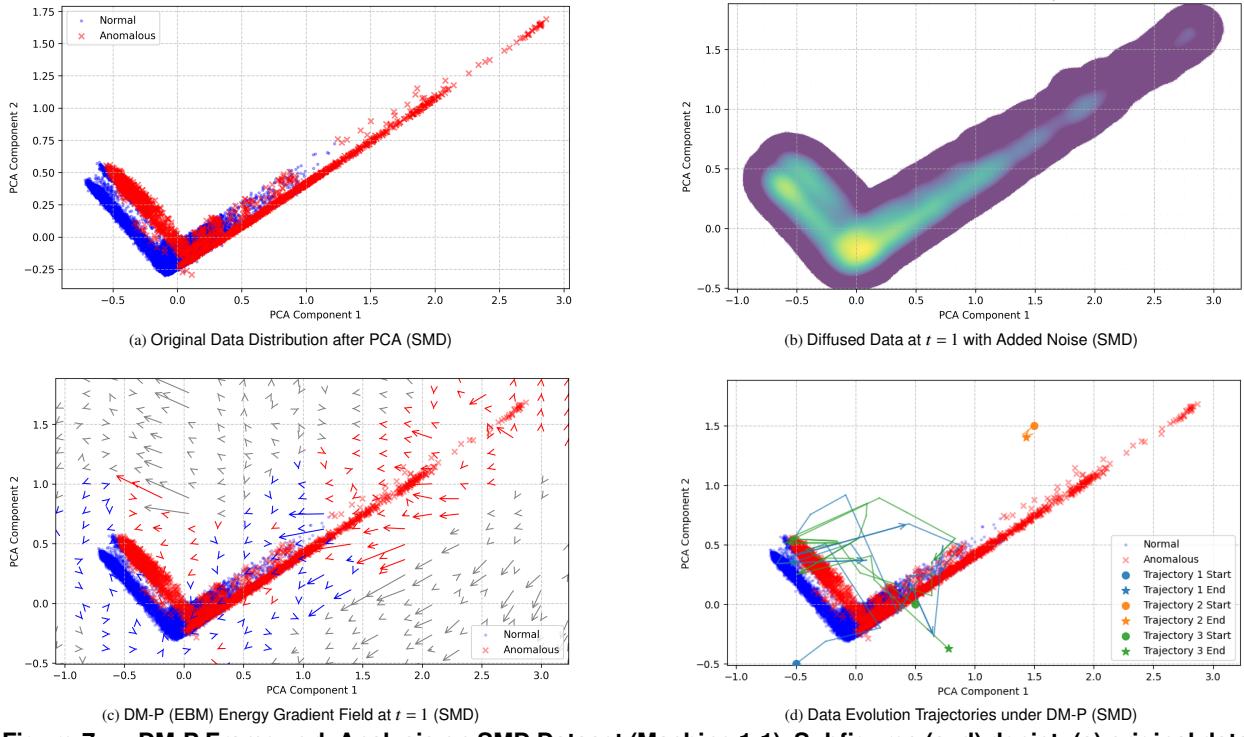
(a) Original Data Distribution after PCA (SMD)

(b) Diffused Data at $t = 1$ with Added Noise (SMD)

(c) DM-P (EBM) Energy Gradient Field at $t = 1$ (SMD)

(d) Data Evolution Trajectories under DM-P (SMD)

Figure 7 **DM-P Framework Analysis on SMD Dataset (Machine 1-1). Subfigures (a–d) depict: (a) original data distribution after PCA, (b) diffused data at time step $k = 1$, (c) energy gradient field at $t = 1$, and (d) data evolution trajectories.**