

EVOLVE: Uncertainty-Aware Multivariate Time Series Anomaly Prediction with Adaptive Prototype Memories

Jianxin Jin¹, Kai Zhao², Shiyao Hu¹, Yang Shu¹, Peng Chen³, Meng Wang³,

Zhongwen Rao⁴, Yijie Li⁵, Jing Wang⁵, Bin Yang¹, Chenjuan Guo¹

¹East China Normal University, ²Aalborg University, ³2012 APPLab, Huawei,

⁴Huawei Technologies Co., Ltd, ⁵Huawei Cloud, Huawei,

{jxjin,syhu}@stu.ecnu.edu.cn, {kaiz}@cs.aau.dk, {yshu,byang,cjguo}@dase.ecnu.edu.cn,

{chenpeng192,wangmeng71,raozhongwen,liyijie5,wangjing630}@huawei.com

Abstract—Unsupervised multivariate time series anomaly prediction (AP) aims to estimate future anomaly risks from current observations, providing early-warning signals for proactive intervention and maintenance in complex systems. Existing unsupervised AP methods follow a perturbation-based paradigm, where predefined or random perturbations are injected into historical windows to simulate abnormal future evolution. However, these methods share perturbations across different variables and contexts, and learn deterministic history-to-future evolution. In reality, anomaly evolution is variable- and context-dependent: the same perturbation may be amplified into abnormal future dynamics in one variable or context, while being absorbed as normal fluctuation in another. In addition, future evolution is not uniquely determined by a historical window, since similar histories may lead to multiple plausible trajectories, such as gradual degradation or abrupt changes. To address these problems, we propose EVOLVE, a unified framework that models unsupervised AP via variable- and context-conditioned perturbations and uncertainty-aware future reasoning. EVOLVE constructs context-specific anomaly perturbations through an adaptive anomaly prototype memory, which can dynamically update and retrieve relevant perturbations at the variable level. EVOLVE models uncertainty by sampling multiple latent historical states and decoding through complementary smooth and deviation-amplifying evolution hypotheses, and the disagreement among decoded future trajectories is used to estimate future anomaly risk. Extensive experiments on six real-world benchmark datasets demonstrate that EVOLVE consistently outperforms state-of-the-art anomaly prediction baselines.

Index Terms—Time Series, Anomaly Prediction, Adaptive Memory, Uncertainty

I. INTRODUCTION

Traditional unsupervised time series anomaly detection (AD) aims to identify abnormal patterns after they have already occurred [1], [2]. It has been widely studied due to its critical role in maintaining the safety and stability of complex systems, such as industrial control systems, cloud computing platforms, AIOps, and transportation systems [3], [4]. However, in many real-world applications, detecting anomalies after their occurrence is often insufficient, since system failures or abnormal events may have already caused service degradation, financial loss, or safety risks. Time series

anomaly prediction (AP) addresses this limitation by estimating the risk of future anomalies from current observations before abnormal events fully manifest [5], [6]. By providing early-warning signals, unsupervised AP enables proactive intervention, predictive maintenance, and risk-aware decision making in complex systems. Despite its practical importance, unsupervised AP is challenging because future anomaly labels are typically unavailable during training, making it difficult to directly learn the transition from normal histories to abnormal futures.

To address the absence of future anomaly labels, existing unsupervised AP methods commonly construct surrogate supervision through perturbation-based training. Given historical observations that are assumed to be normal or mostly normal, these methods inject synthetic anomaly perturbations into historical windows and train the model to predict or distinguish the corresponding abnormal future evolution [7]–[10]. The perturbations are usually sampled from predefined anomaly libraries or generated by random noise, serving as proxy abnormal patterns when ground-truth anomaly evolution is unavailable [8], [11], [12]. This perturbation injection and evolution prediction paradigm provides a practical way to learn the transition from normal histories to potential abnormal futures, and has become a common framework for unsupervised anomaly prediction. Nevertheless, reliable AP in heterogeneous multivariate time series remains intrinsically challenging, because anomaly evolution is variable- and context-dependent, and future evolution is inherently uncertain.

Challenge 1: Perturbation responses are highly heterogeneous across different variables and contexts [13]. As illustrated in Figure 1, the same type of perturbation may be amplified into abnormal future evolution in one variable or context state, while being absorbed as a normal fluctuation in another. This phenomenon indicates that anomaly prediction cannot rely on globally shared or random perturbations; instead, the injected anomaly patterns should be conditioned on the variable-level characteristics and historical context of each time series. However, existing unsupervised anomaly

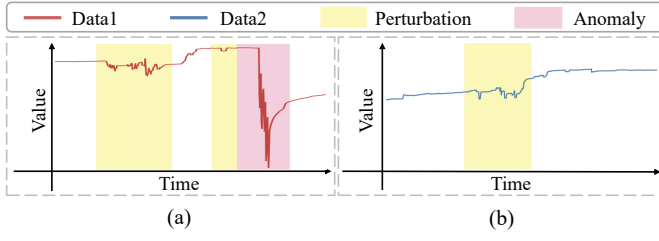


Fig. 1. Different impacts of similar historical perturbations on future evolution under time series heterogeneity. (a) Perturbations are amplified with abnormal evolution. (b) Perturbations are absorbed with normal evolution.

prediction methods typically simulate abnormal futures using predefined anomaly libraries or random perturbations shared across different time series [7]–[9]. Such static or context-agnostic perturbation strategies make it difficult to capture heterogeneous perturbation responses, especially when different variables exhibit distinct sensitivities to similar disturbances.

Challenge 2: Future anomaly evolutions are highly uncertain. Anomaly prediction is inherently uncertain because the same historical window may correspond to multiple plausible future evolutions under different system conditions or failure modes. On the one hand, noisy or incomplete historical observations may lead to ambiguous latent semantics, making it difficult to determine whether the current state is stable or approaching abnormal evolution [14]. On the other hand, even similar historical patterns may evolve into different future trajectories, such as normal absorption, gradual degradation, or abrupt abnormal changes [15], [16]. Nevertheless, most existing anomaly prediction methods adopt deterministic mappings from history to future, producing a single predicted outcome or anomaly decision [7], [10], [17]. This is insufficient to characterize the diversity of possible future evolutions, which leads to unreliable anomaly prediction.

To address the above challenges, we propose **EVOLVE**, a unified framework for unsupervised multivariate time series anomaly prediction. Instead of treating perturbation injection as data augmentation, EVOLVE models anomaly prediction with *variable-conditioned perturbation* and *uncertainty-aware multi-hypothesis future reasoning*. For the first challenge, EVOLVE introduces an adaptive anomaly prototype memory to construct context-specific anomaly perturbations. The memory combines prior anomaly patterns with dynamically updated prototypes accumulated during training. Given a historical representation, EVOLVE retrieves relevant anomaly prototypes at the variable level and injects them into the latent representation, producing anomaly-based historical states tailored to the current time series. Moreover, an adversarial perturbation learning mechanism encourages the injection module to generate hard and informative perturbations, enabling the model to better distinguish heterogeneous anomaly responses across different variables and temporal contexts.

For the second challenge, EVOLVE develops an uncertainty-aware module together with a dual-structured decoder. The latent module parameterizes the distribution

for historical representations and samples multiple possible states, capturing the uncertainty of historical semantics. Each sampled latent state is then decoded through two complementary evolution hypotheses: a smooth evolution hypothesis, which models normal absorption of perturbations, and a deviation-amplifying evolution hypothesis, which models the amplification of local deviations into potential abnormal dynamics. By generating multiple plausible future paths and measuring their disagreement, EVOLVE explicitly characterizes predictive uncertainty and estimates future anomaly risk without relying on labeled anomalies.

The main contributions of this work are summarized as follows:

- We formulate unsupervised time series anomaly prediction as variable-conditioned multi-hypothesis future risk estimation, where anomaly risk is inferred from the uncertainty among plausible future evolutions.
- We propose an adaptive anomaly prototype memory with variable-level perturbation retrieval and dynamic updating, enabling context-specific anomaly injection beyond static or globally shared perturbation libraries.
- We design an uncertainty-aware latent modeling and dual-structured decoding mechanism to generate multiple future evolution paths under smooth and deviation-amplifying hypotheses.
- Extensive experiments on six real-world benchmark datasets demonstrate the effectiveness and efficiency of EVOLVE.

II. RELATED WORK

A. Unsupervised Time Series Anomaly Prediction

Time series anomaly prediction aims to anticipate future abnormal events before they fully occur, providing more proactive warning signals than conventional anomaly detection. Compared with post-hoc anomaly detection, unsupervised anomaly prediction is more challenging because future anomaly labels are unavailable during training, making it difficult to directly learn the transition from normal histories to abnormal futures [7], [8], [12], [18], [19].

One line of studies formulates anomaly prediction as precursor identification. PAD introduces Precursor-of-Anomaly (PoA) detection to identify early signals that precede future anomalies [20]. IGCL develops a generative contrastive learning framework that constructs diverse anomaly precursor samples without requiring anomaly labels [8]. TAP combines reconstructive learning with contrastive objectives to model anomaly-related patterns with different reaction times [7]. These methods provide useful formulations for unsupervised AP, but their precursor or perturbation construction strategies are usually shared across different time series, making it difficult to adapt to heterogeneous variable-level responses.

Recent methods further model abnormal future evolution through forecasting-based objectives. A2P leverages anomaly-aware forecasting and synthetic anomaly prompting to inject anomaly-related representations during training [9]. RED-F proposes a reconstruction-elimination based dual-stream

contrastive forecasting framework, where a purified normal-pattern baseline and the original sequence are forecasted in parallel to amplify weak anomaly precursors through relative trajectory comparison [10]. FCM introduces future context modeling for abnormality forecasting by exploiting future-oriented contextual information and normality correlations between observations and future contexts [21]. These forecasting-based AP methods improve the modeling of future abnormal patterns and serve as strong baselines in our experiments. However, existing AP methods usually construct precursor or perturbation patterns in a globally shared manner, or model future evolution through deterministic forecasting objectives. Such context-agnostic perturbation construction and single-path future modeling are insufficient for heterogeneous multivariate time series, where the same disturbance may be amplified in one variable or temporal state but absorbed in another. In contrast, EVOLVE constructs anomaly perturbations conditioned on variable-level historical representations through an adaptive anomaly prototype memory, and further reasons over multiple plausible future evolutions for anomaly risk estimation.

B. Point-level Anomaly Prediction and Future Uncertainty

Existing anomaly prediction studies differ in the granularity of their prediction targets. Some methods adopt an interval-level formulation, where the objective is to determine whether an anomaly will occur within a future window based on historical observations [7], [8], [20]. For example, PAD outputs an overall risk score for a future interval, while IGCL and TAP infer future anomaly likelihood by modeling precursor patterns or reaction periods [7], [8], [20]. Although interval-level prediction is useful for early warning, it does not localize fine-grained anomalous timestamps within the future horizon, which limits its applicability in scenarios requiring precise intervention.

To provide more precise warning signals, recent studies have considered point-level anomaly prediction. You et al. [19] formulate anomaly prediction as estimating future anomaly indicators at each timestamp. A2P, RED-F, and FCM further perform step-wise or future-oriented anomaly prediction for future sequences, enabling more fine-grained temporal localization [9], [10], [21]. Our work also follows the point-level AP setting, where the model predicts anomaly risks for future timestamps rather than only producing an interval-level warning. However, point-level AP is inherently more difficult because future evolution before anomalies occur is often uncertain: similar historical windows may lead to normal absorption, gradual degradation, or abrupt abnormal changes.

Uncertainty-aware time series forecasting has been studied in probabilistic forecasting and VAE-style generative models [16], [22], which usually aim to model the distribution of future observations or improve forecasting likelihood. However, these methods are not specifically designed for anomaly prediction, where the key objective is to estimate whether future trajectories will diverge toward abnormal states. Most existing AP methods still adopt deterministic history-to-future

mappings and generate a single future outcome, limiting their ability to characterize multiple plausible future evolutions. Different from generic stochastic forecasting, EVOLVE performs task-specific uncertainty modeling for AP: it samples multiple latent historical hypotheses and decodes them through smooth and deviation-amplifying evolution hypotheses. The predictive disagreement among decoded future trajectories is then used to estimate future anomaly risk.

III. METHODOLOGY

A. Problem Setup

Let $X_{1:T} = (x_1, x_2, \dots, x_T) \in \mathbb{R}^{T \times C}$ denote a historical multivariate time series window of length T with C variables, where $x_t \in \mathbb{R}^C$ is the observation at time step t . Given $X_{1:T}$, unsupervised time series anomaly prediction (AP) aims to estimate anomaly risks for the future horizon $X_{T+1:T+F}$ and produce future anomaly labels $\hat{Y}_{T+1:T+F} = (\hat{y}_{T+1}, \dots, \hat{y}_{T+F})$, where $\hat{y}_t \in \{0, 1\}$ indicates whether time step t is predicted as anomalous [7], [8]. During training, future observations can be used as self-supervised signals to learn future evolution, while future anomaly labels are unavailable. During inference, the true future observations are unavailable, and the model must estimate future anomaly risks solely from the historical window.

In EVOLVE, anomaly risk is estimated from the predictive disagreement among multiple decoded future trajectories. Given a historical window, the model generates K plausible future trajectories:

$$\hat{X}_{T+1:T+F}^{(k)} = (\hat{x}_{T+1}^{(k)}, \hat{x}_{T+2}^{(k)}, \dots, \hat{x}_{T+F}^{(k)}), \quad k = 1, \dots, K, \quad (1)$$

where $\hat{x}_t^{(k)} \in \mathbb{R}^C$ denotes the k -th predicted future observation at time step t . We emphasize that the anomaly score is not computed from the variance of latent variables, but from the predictive variance among multiple decoded future trajectories. Intuitively, if the historical window admits a stable future evolution, different plausible trajectories should remain consistent. In contrast, when the system is approaching abnormal evolution, plausible future trajectories tend to diverge, resulting in higher predictive disagreement.

Specifically, let $\hat{x}_{t,c}^{(k)}$ denote the prediction of the c -th variable at future time step t from the k -th trajectory. The anomaly score at time step t is computed by aggregating variable-wise predictive variance:

$$s_t = \frac{1}{C} \sum_{c=1}^C \text{Var}(\hat{x}_{t,c}^{(1)}, \hat{x}_{t,c}^{(2)}, \dots, \hat{x}_{t,c}^{(K)}), \quad t = T+1, \dots, T+F. \quad (2)$$

A larger s_t indicates stronger disagreement among plausible future trajectories and thus higher future anomaly risk. Binary future anomaly labels are then obtained by thresholding the anomaly score:

$$\hat{y}_t = \begin{cases} 1, & s_t \geq \tau, \\ 0, & s_t < \tau, \end{cases} \quad t = T+1, \dots, T+F, \quad (3)$$

where τ is the decision threshold.

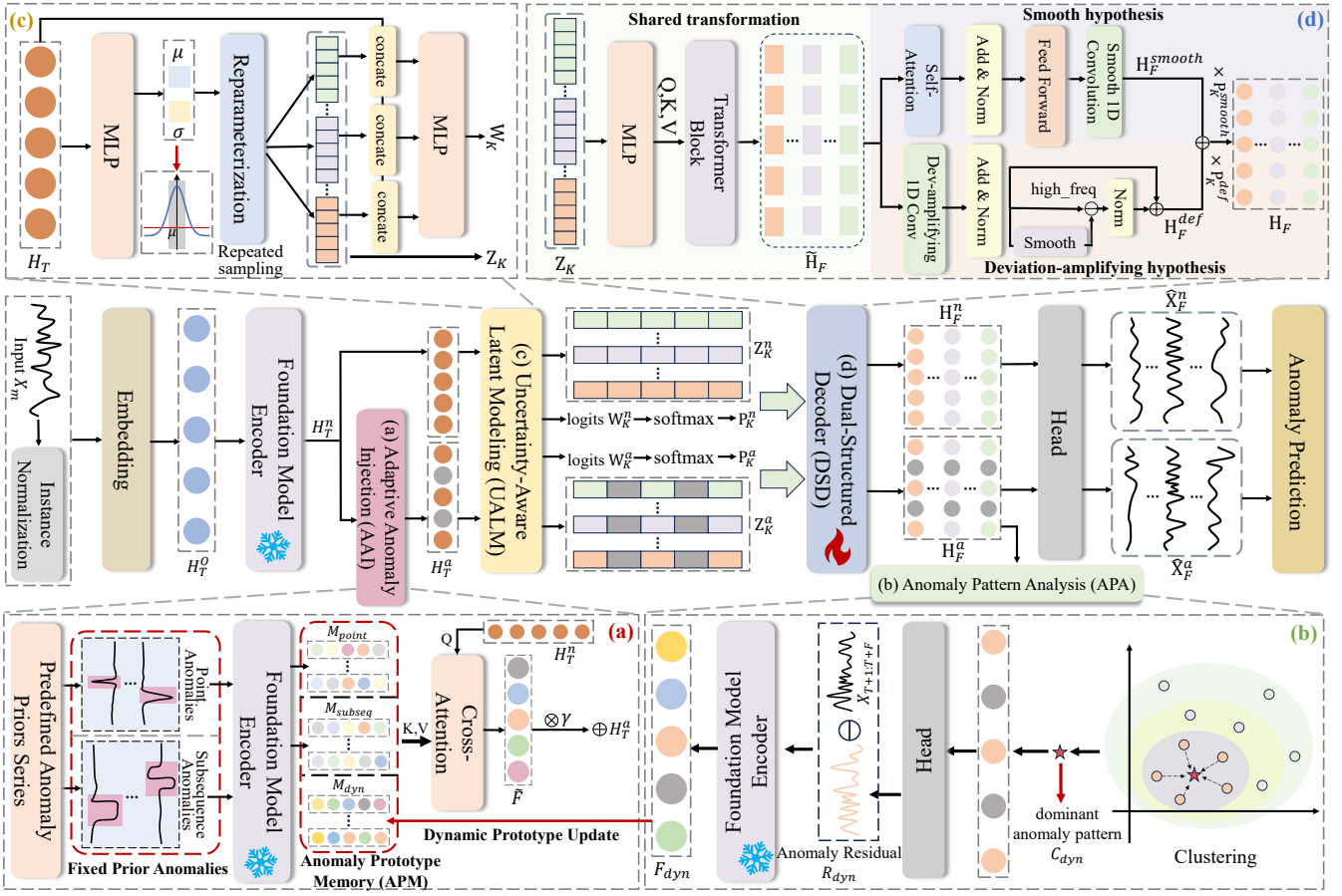


Fig. 2. Overall architecture of EVOLVE. Given a historical window, EVOLVE first encodes the input into a normal historical representation H_T^n . During training, Adaptive Anomaly Injection (AAI) retrieves variable-conditioned perturbations from the Anomaly Prototype Memory (APM) and constructs an anomaly-aware historical representation H_T^a , while Anomaly Pattern Analysis (APA) progressively updates the dynamic prototype bank using residual-based anomaly-aware future representations. Both the normal and anomaly-aware historical views are then processed by Uncertainty-Aware Latent Modeling (UALM), which samples multiple latent semantic hypotheses and estimates their evolution-mode probabilities. The Dual-Structured Decoder (DSD) decodes these hypotheses through smooth and deviation-amplifying evolution hypotheses to generate multiple plausible future trajectories. During inference, anomaly-aware injection, APA, and dynamic prototype updating are disabled, and future anomaly scores are computed from the predictive disagreement among decoded trajectories.

B. Design Principles

EVOLVE is built upon two design principles for unsupervised anomaly prediction in heterogeneous multivariate time series.

Principle 1: Variable-conditioned perturbation modeling. Anomaly responses are variable- and context-dependent. The same perturbation may be amplified into abnormal future dynamics for one variable or context state, while being absorbed as normal fluctuation for another. Therefore, perturbations should not be treated as globally shared data augmentation patterns. Instead, they should be conditioned on variable-level historical representations and temporal contexts.

Principle 2: Multi-hypothesis future reasoning. Future evolution in anomaly prediction is not uniquely determined by the historical window. Similar histories may lead to multiple plausible futures, including normal absorption, gradual degradation, and abrupt abnormal changes. Therefore, anomaly prediction should reason over multiple possible future trajectories rather than relying on a single deterministic prediction.

Following these principles, EVOLVE constructs variable-conditioned anomaly prototype modeling, and estimates future anomaly risk from the disagreement among multiple decoded future trajectories.

C. Overall Framework

Figure 2 presents the overall framework of EVOLVE. The framework consists of three major components: (1) adaptive anomaly prototype modeling, which constructs an anomaly-aware historical view from the normal historical representation; (2) uncertainty-aware latent modeling, which samples multiple latent semantic hypotheses from historical representations; and (3) dual-structured decoding, which decodes each latent hypothesis into future trajectories under complementary evolution hypotheses.

Given an input historical window $X_{1:T}$, EVOLVE first applies instance normalization and an embedding layer to obtain an initial representation H_T^O . A time series encoder is then used to obtain the normal historical representation

H_T^n . This representation serves as the *normal historical view*, which describes the observed historical state without injected perturbations.

To model heterogeneous anomaly responses, EVOLVE further constructs an *anomaly-aware historical view*. Specifically, the Adaptive Anomaly Injection (AAI) module retrieves relevant anomaly prototypes from the Anomaly Prototype Memory (APM) according to the current variable-level historical representation, and injects the retrieved perturbation into H_T^n to obtain H_T^a . The APM contains both fixed prior anomaly patterns and dynamically updated anomaly prototypes. The dynamic prototype bank is initialized as an empty queue and is progressively updated during training by Anomaly Pattern Analysis (APA), rather than being randomly initialized. In this way, the anomaly-aware view provides context-specific perturbations for learning heterogeneous abnormal evolution.

Both the normal historical view H_T^n and the anomaly-aware historical view H_T^a are passed into the Uncertainty-Aware Latent Modeling (UALM) module. For each view, UALM parameterizes a latent distribution over historical semantics and samples K latent hypotheses. Each latent hypothesis is also associated with a two-dimensional evolution-mode logit vector, corresponding to a smooth evolution hypothesis and a deviation-amplifying evolution hypothesis. Softmax is applied along this two-hypothesis dimension independently for each latent sample.

The sampled latent hypotheses are then decoded by the Dual-Structured Decoder (DSD). DSD contains a shared transformation module followed by two branches corresponding to complementary evolution hypotheses. The smooth evolution hypothesis models stable future evolution where perturbations are absorbed by the system, while the deviation-amplifying evolution hypothesis models the amplification of local deviations into potential abnormal dynamics. The two decoded representations are fused according to the evolution-mode probabilities, producing K plausible future trajectories.

During training, both H_T^n and H_T^a are used. The normal historical view is encouraged to produce consistent future trajectories, while the anomaly-aware historical view is encouraged to produce diverse and deviation-aware trajectories. An adversarial perturbation learning strategy further encourages AAI to generate hard but plausible perturbations. During inference, anomaly-aware injection and dynamic prototype updating are disabled. EVOLVE only uses the normal historical view to generate K future trajectories, and computes anomaly scores from their predictive variance as defined in Eq. (2).

D. Adaptive Anomaly Prototype Modeling

In heterogeneous multivariate time series, anomaly responses are highly variable- and context-dependent. A similar disturbance may be amplified into abnormal future dynamics for one variable or temporal state, while being absorbed as normal fluctuation for another. This motivates us to construct anomaly perturbations conditioned on the current historical representation, rather than using static or globally shared perturbations. To this end, EVOLVE introduces an adaptive

anomaly prototype modeling mechanism, which constructs an anomaly-aware historical view H_T^a from the normal historical view H_T^n .

This mechanism consists of three components. First, the Anomaly Prototype Memory (APM) stores both general anomaly priors and dynamically updated anomaly prototypes. Second, Adaptive Anomaly Injection (AAI) retrieves variable-relevant anomaly prototypes from APM and injects them into H_T^n to obtain H_T^a . Third, Anomaly Pattern Analysis (APA) extracts residual-based dynamic prototypes from anomaly-aware future representations during training and updates the dynamic memory bank. The dynamic update is performed only during training, while inference uses only the normal historical view for future anomaly scoring.

1) *Anomaly Prototype Memory*: To support variable-conditioned perturbation modeling, we introduce the *Anomaly Prototype Memory (APM)* as shown in Figure 2(a). APM maintains three types of anomaly prototype banks: a point-wise prior bank, a subsequence-wise prior bank, and a dynamic prototype bank:

$$\mathcal{M}_b = \{F_i^b\}_{i=1}^{n_b}, \quad b \in \{\text{point}, \text{subseq}, \text{dyn}\}, \quad (4)$$

where $F_i^b \in \mathbb{R}^D$ denotes an anomaly prototype in the same representation space as the historical representation, n_b is the current number of stored prototypes, and N_b is the capacity of the corresponding bank with $n_b \leq N_b$.

The point-wise and subsequence-wise banks provide general anomaly priors. They are initialized from predefined point-wise and subsequence-wise anomaly patterns [23], [24], which are encoded by the shared time series encoder to form fixed prior prototypes. These fixed banks offer stable references for common abnormal patterns, especially at the early stage of training.

In contrast, the dynamic prototype bank is designed to capture context-dependent anomaly patterns that cannot be fully covered by predefined priors. Importantly, the dynamic bank is initialized as an empty queue rather than random vectors, i.e., $n_{\text{dyn}} = 0$ at the beginning of training. At the beginning of training, prototype retrieval relies only on the fixed point-wise and subsequence-wise banks. As training proceeds, APA progressively writes residual-based dynamic prototypes into the dynamic bank, and a first-in-first-out (FIFO) strategy is adopted when the bank reaches its capacity. This design avoids introducing noisy random prototypes and enables the memory to gradually accumulate context-dependent anomaly patterns during training.

2) *Adaptive Anomaly Injection*: Given the normal historical view H_T^n , Adaptive Anomaly Injection (AAI) retrieves context-relevant anomaly prototypes from APM and constructs the anomaly-aware historical view H_T^a . Let

$$\mathcal{M}_{\text{all}} = \text{Concat}(\mathcal{M}_{\text{point}}, \mathcal{M}_{\text{subseq}}, \mathcal{M}_{\text{dyn}}) \quad (5)$$

denote the stacked memory of all available anomaly prototypes. AAI uses the normal historical representation as the query and the prototype memory as keys and values:

$$\tilde{F} = \text{CrossAttn}(Q = H_T^n, K = \mathcal{M}_{\text{all}}, V = \mathcal{M}_{\text{all}}), \quad (6)$$

where $\tilde{F}$ has the same representation shape as H_T^n and denotes the retrieved anomaly perturbation conditioned on the current historical context.

The anomaly-aware historical view is then constructed as

$$H_T^a = H_T^n + \gamma \cdot \tilde{F}, \quad (7)$$

where γ is a learnable injection strength that controls the contribution of the retrieved perturbation. We use γ to denote the injection strength and reserve the abnormal margin in the training objective as a separate symbol, avoiding notation ambiguity. Through cross-attention retrieval, AAI adaptively selects perturbations according to the current variable-level historical representation, enabling variable-conditioned anomaly injection rather than globally shared perturbation augmentation.

3) *Dynamic Prototype Updating*: The dynamic prototype bank is updated by *Anomaly Pattern Analysis (APA)* during training, as shown in Figure 2(b). The goal of APA is to refine the memory with residual patterns exposed by anomaly-aware decoding, so that APM can gradually adapt to heterogeneous and context-dependent anomaly evolution.

After the anomaly-aware historical view H_T^a is constructed, the subsequent UALM and DSD modules decode it into K anomaly-aware future representations. Let $\mathbf{H}_F^a = \{(H_F^a)^{(k)}\}_{k=1}^K$ denote these future representations. APA clusters these representations in the latent future space and selects the cluster with the largest number of members as the dominant anomaly-aware evolution pattern. The mean representation of this dominant cluster is denoted as

$$C_{\text{dyn}}. \quad (8)$$

Since C_{dyn} may still contain substantial normal future evolution context, directly writing it into the memory may blur anomaly-specific information. Therefore, APA first maps C_{dyn} to the time series space through a lightweight prediction head and computes its residual with respect to the observed future window:

$$R_{\text{dyn}} = \text{Head}(C_{\text{dyn}}) - X_{T+1:T+F}. \quad (9)$$

The residual highlights the deviation between anomaly-aware future evolution and the actual future observations, thereby suppressing normal background information and emphasizing anomaly-specific patterns.

The residual R_{dyn} is then encoded by the shared time series encoder to obtain a dynamic prototype:

$$F_{\text{dyn}} = \text{Encoder}(R_{\text{dyn}}). \quad (10)$$

Finally, F_{dyn} is written into the dynamic prototype bank. When the bank reaches its capacity, the oldest prototype is removed following FIFO replacement.

It is worth noting that APA is used only during training. Future observations are used as self-supervised signals to construct residual-based prototypes, similar to forecasting-based training objectives. During inference, the future window is unavailable, and both anomaly-aware injection and dynamic prototype updating are disabled. Therefore, no future information is used for test-time anomaly scoring.

E. Uncertainty-Aware Multi-Hypothesis Future Modeling

Future evolution in anomaly prediction is not uniquely determined by the historical window. Even similar histories may evolve into normal absorption, gradual degradation, or abrupt abnormal changes. Therefore, a single deterministic representation is insufficient to characterize future anomaly risk. To address this issue, EVOLVE introduces uncertainty-aware latent modeling and dual-structured decoding to generate multiple plausible future trajectories.

1) *Uncertainty-Aware Latent Modeling*: The *Uncertainty-Aware Latent Modeling (UALM)* module models multiple semantic hypotheses from historical representations. As described in Section III-C, EVOLVE constructs two historical views during training: the normal historical view H_T^n and the anomaly-aware historical view H_T^a . UALM adopts shared parameters for both views so that normal and anomaly-aware semantics are represented in a unified latent space.

For notation simplicity, we denote the input to UALM as a generic historical representation H_T , where $H_T \in \{H_T^n, H_T^a\}$ during training. UALM first parameterizes a latent Gaussian distribution:

$$[\mu, \rho] = \text{MLP}_z(H_T), \quad (11)$$

where $\mu \in \mathbb{R}^{L_z}$ denotes the latent mean and $\rho \in \mathbb{R}^{L_z}$ is transformed into a positive standard deviation:

$$\sigma = \text{softplus}(\rho) + \epsilon_0, \quad (12)$$

with a small constant ϵ_0 for numerical stability. Then K latent semantic hypotheses are sampled through reparameterization:

$$Z^{(k)} = \mu + \eta^{(k)} \odot \sigma, \quad \eta^{(k)} \sim \mathcal{N}(0, I), \quad k = 1, \dots, K. \quad (13)$$

The sampled latent set is denoted as

$$\mathbf{Z}_K = \{Z^{(k)}\}_{k=1}^K. \quad (14)$$

When the input is H_T^n or H_T^a , the corresponding latent sets are denoted as $\mathbf{Z}_K^n$ and $\mathbf{Z}_K^a$, respectively. Each latent sample represents one plausible interpretation of the historical semantics.

In addition to latent sampling, UALM estimates the evolution-mode distribution for each latent hypothesis. For each $Z^{(k)}$, UALM concatenates it with the historical representation and outputs a two-dimensional logit vector:

$$W^{(k)} = \text{MLP}_w([Z^{(k)}; H_T]), \quad W^{(k)} \in \mathbb{R}^2. \quad (15)$$

The two entries of $W^{(k)}$ correspond to the smooth evolution hypothesis and the deviation-amplifying evolution hypothesis. We then apply softmax along this two-hypothesis dimension:

$$p^{(k)} = \text{softmax}(W^{(k)}) = [p_{\text{smooth}}^{(k)}, p_{\text{dev}}^{(k)}], \quad (16)$$

where $p_{\text{smooth}}^{(k)} + p_{\text{dev}}^{(k)} = 1$. Importantly, the softmax is applied over the two evolution hypotheses for each latent sample, rather than across the K sampled paths. For normal and anomaly-aware historical views, the corresponding evolution-mode distributions are denoted as $\mathbf{P}_K^n$ and $\mathbf{P}_K^a$.

UALM differs from generic VAE-style stochastic forecasting in both objective and usage. It is not designed to

maximize the likelihood of future observations or to directly use latent variance as anomaly scores. Instead, UALM samples multiple semantic hypotheses and associates each hypothesis with an evolution-mode distribution. These hypotheses are decoded into multiple future trajectories, and their predictive disagreement is later used to estimate future anomaly risk. Therefore, the uncertainty modeled by UALM is task-specific and risk-oriented for anomaly prediction.

2) *Dual-Structured Decoder*: Given the latent hypotheses $\mathbf{Z}_K$ and their evolution-mode probabilities $\mathbf{P}_K$, the *Dual-Structured Decoder (DSD)* decodes each latent hypothesis into a plausible future trajectory. Different from deterministic forecasting decoders, DSD implements two complementary evolution hypotheses through two decoding branches: a smooth evolution hypothesis and a deviation-amplifying evolution hypothesis. The former describes stable future evolution where deviations or perturbations are absorbed by the system, while the latter describes the amplification of local deviations into potential abnormal dynamics.

For notation clarity, we describe the decoding process for a single latent hypothesis $Z^{(k)}$, while all K hypotheses are processed in parallel in practice.

Shared latent transformation. Each latent hypothesis is first projected into the future hidden space:

$$\tilde{H}_F^{(k)} = \text{Transformer}(\text{MLP}(Z^{(k)})), \quad (17)$$

where $\tilde{H}_F^{(k)}$ denotes the shared future hidden representation. This shared transformation captures common temporal dependencies and provides the basis for the following two evolution hypotheses.

Smooth evolution hypothesis. The smooth evolution hypothesis assumes that potential deviations or perturbations are absorbed by the system and the future dynamics remain regular. It applies a SmoothTransformer block to the shared future hidden representation. For notation simplicity, let $H \equiv \tilde{H}_F^{(k)}$. The smooth branch is computed as

$$H' = \text{LN}(H + \text{MHA}(H, H, H)), \quad (18)$$

$$H'' = \text{LN}(H' + \text{FFN}(H')), \quad (19)$$

$$H_{\text{smooth}}^{(k)} = H'' + \text{DWConv}(H''), \quad (20)$$

where $\text{MHA}(\cdot)$ captures global temporal dependencies, $\text{FFN}(\cdot)$ refines the hidden representation, and $\text{DWConv}(\cdot)$ denotes depth-wise temporal convolution. The depth-wise convolution suppresses high-frequency fluctuations and encourages stable and coherent future evolution.

Deviation-amplifying evolution hypothesis. The deviation-amplifying evolution hypothesis models the case where local deviations are enlarged and evolve into potential abnormal dynamics. It processes the shared future representation through a dilated convolution block followed by a contrastive enhancement block. Let $H \equiv \tilde{H}_F^{(k)}$. The dilated convolution block is formulated as

$$H' = \text{LN}(H + \text{PWConv}(\text{DConv}(H))), \quad (21)$$

where $\text{DConv}(\cdot)$ denotes depth-wise dilated temporal convolution and $\text{PWConv}(\cdot)$ denotes point-wise convolution for channel mixing. This block enlarges the temporal receptive field to capture sparse and long-range abnormal patterns.

To emphasize irregular components, the contrastive enhancement block first extracts a smoothed representation:

$$H_{\text{low}} = \text{SmoothConv}(H'), \quad (22)$$

and then computes high-frequency residuals:

$$H_{\text{high}} = \text{LN}(H' - H_{\text{low}}). \quad (23)$$

The final deviation-amplifying representation is obtained as

$$H_{\text{dev}}^{(k)} = H' + H_{\text{high}}. \quad (24)$$

This module does not generate perturbations by itself; rather, it models how already injected or latent deviations may be amplified during future evolution.

Evolution-mode fusion. For each latent hypothesis, the two decoded representations are fused according to the evolution-mode probabilities produced by UALM:

$$H_F^{(k)} = p_{\text{smooth}}^{(k)} \cdot H_{\text{smooth}}^{(k)} + p_{\text{dev}}^{(k)} \cdot H_{\text{dev}}^{(k)}. \quad (25)$$

The fused future hidden representation is then projected to the observation space through a lightweight prediction head:

$$\hat{X}_{T+1:T+F}^{(k)} = \text{Head}(H_F^{(k)}). \quad (26)$$

The set of decoded future trajectories is denoted as

$$\hat{\mathbf{X}}_F = \{\hat{X}_{T+1:T+F}^{(k)}\}_{k=1}^K. \quad (27)$$

During training, DSD is applied to both the normal historical view and the anomaly-aware historical view, producing $\hat{\mathbf{X}}_F^n$ and $\hat{\mathbf{X}}_F^a$, respectively. The normal view is encouraged to produce consistent future trajectories, while the anomaly-aware view is encouraged to expose deviation-sensitive future evolution. During inference, only the normal historical view is used, and future anomaly scores are computed from the predictive variance among $\hat{\mathbf{X}}_F^n$.

F. Training Objective and Adversarial Optimization

EVOLVE is trained with a two-stage alternating optimization strategy. The first stage optimizes the predictor to learn discriminative future evolution from both the normal historical view and the anomaly-aware historical view. The second stage optimizes the perturbation generator to produce hard but plausible anomaly-aware views. This separation clarifies the roles of the predictor and the anomaly injection module during adversarial learning.

During training, EVOLVE constructs two historical views from the same input window: the normal historical view H_T^n and the anomaly-aware historical view H_T^a . The normal view is expected to produce consistent future trajectories that match the observed future window, while the anomaly-aware view is expected to expose deviation-sensitive future evolution induced by variable-conditioned perturbations. Let $\hat{\mathbf{X}}_F^n = \{(\hat{X}_F^n)^{(k)}\}_{k=1}^K$ and $\hat{\mathbf{X}}_F^a = \{(\hat{X}_F^a)^{(k)}\}_{k=1}^K$ denote the

K decoded future trajectories from the normal and anomaly-aware historical views, respectively.

Normal-view consistency and forecasting losses. For the normal historical view, multiple decoded future trajectories should be mutually consistent and close to the observed future window $X_{T+1:T+F}$. We define the predictive variance across the K trajectories as

$$\text{Var}_K(\hat{\mathbf{X}}_F^n) = \frac{1}{FC} \sum_{t=T+1}^{T+F} \sum_{c=1}^C \text{Var} \left((\hat{x}_{t,c}^n)^{(1)}, \dots, (\hat{x}_{t,c}^n)^{(K)} \right). \quad (28)$$

The normal-view consistency loss is then defined as

$$\mathcal{L}_{\text{con}}^n = \text{Var}_K(\hat{\mathbf{X}}_F^n). \quad (29)$$

Meanwhile, the mean prediction should reconstruct the observed future:

$$\mathcal{L}_{\text{rec}}^n = \text{MSE} \left(\frac{1}{K} \sum_{k=1}^K (\hat{\mathbf{X}}_F^n)^{(k)}, X_{T+1:T+F} \right). \quad (30)$$

The normal-view objective is

$$\mathcal{L}_n = \mathcal{L}_{\text{con}}^n + \mathcal{L}_{\text{rec}}^n. \quad (31)$$

Anomaly-aware diversity and deviation losses. For the anomaly-aware historical view, the injected perturbation should induce futures that are distinguishable from stable normal evolution. Therefore, the decoded trajectories are encouraged to have sufficiently large predictive diversity and deviation from the observed future. Instead of reusing the injection strength symbol, we define a separate abnormal margin m :

$$m = \lambda_m \cdot \text{sg}(\gamma^2), \quad (32)$$

where γ is the injection strength in Eq. (7), λ_m is a scaling coefficient, and $\text{sg}(\cdot)$ denotes the stop-gradient operation. Detaching the margin prevents trivial solutions where the model reduces the margin to satisfy the loss.

The anomaly-aware diversity loss is defined as

$$\mathcal{L}_{\text{div}}^a = \text{ReLU} \left(m - \text{Var}_K(\hat{\mathbf{X}}_F^a) \right), \quad (33)$$

where $\text{Var}_K(\hat{\mathbf{X}}_F^a)$ is computed in the same way as the normal-view predictive variance. The anomaly-aware deviation loss is defined as

$$\mathcal{L}_{\text{dev}}^a = \text{ReLU} \left(m - \text{MSE} \left(\frac{1}{K} \sum_{k=1}^K (\hat{\mathbf{X}}_F^a)^{(k)}, X_{T+1:T+F} \right) \right). \quad (34)$$

The anomaly-aware objective is

$$\mathcal{L}_a = \mathcal{L}_{\text{div}}^a + \mathcal{L}_{\text{dev}}^a. \quad (35)$$

Evolution-mode alignment loss. UALM outputs a two-dimensional evolution-mode logit vector $W^{(k)}$ for each latent hypothesis, corresponding to the smooth evolution hypothesis and the deviation-amplifying evolution hypothesis. Let $W_n^{(k)}$ and $W_a^{(k)}$ denote the logits produced from the normal and anomaly-aware historical views, respectively. The normal view

is encouraged to align with the smooth evolution hypothesis, while the anomaly-aware view is encouraged to align with the deviation-amplifying evolution hypothesis:

$$\mathcal{L}_{\text{align}} = \frac{1}{K} \sum_{k=1}^K \left[\text{CE}(W_n^{(k)}, 0) + \text{CE}(W_a^{(k)}, 1) \right], \quad (36)$$

where class 0 denotes the smooth evolution hypothesis and class 1 denotes the deviation-amplifying evolution hypothesis. Equivalently, the one-hot targets are $[1, 0]$ for the normal view and $[0, 1]$ for the anomaly-aware view.

Predictor optimization. In the predictor optimization stage, the anomaly injection module is fixed, and the predictor, including the encoder, UALM, DSD, and prediction head, is optimized by minimizing

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_n + \mathcal{L}_a + \mathcal{L}_{\text{align}}. \quad (37)$$

This objective encourages the normal historical view to produce consistent and accurate future trajectories, while encouraging the anomaly-aware historical view to reveal deviation-sensitive future evolution.

Perturbation generator optimization. In the perturbation generator optimization stage, the predictor is fixed and only the anomaly injection module is updated. The goal is to prevent the generator from producing overly obvious or trivial perturbations. Instead, the generator is encouraged to construct hard but plausible anomaly-aware views whose evolution-mode distributions are close to those of normal views.

For each latent hypothesis, we first compute the evolution-mode probabilities:

$$p_n^{(k)} = \text{softmax}(W_n^{(k)}), \quad p_a^{(k)} = \text{softmax}(W_a^{(k)}). \quad (38)$$

Here, softmax is applied along the two-hypothesis dimension of each logit vector, rather than across the K sampled future paths. The adversarial loss for the perturbation generator is then defined as

$$\mathcal{L}_{\text{adv}}^G = \frac{1}{K} \sum_{k=1}^K \text{KL} \left(p_a^{(k)} \parallel \text{sg}(p_n^{(k)}) \right), \quad (39)$$

where $\text{sg}(\cdot)$ stops gradients from flowing into the predictor. The generator is optimized by minimizing

$$\mathcal{L}_G = \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}^G, \quad (40)$$

where λ_{adv} controls the strength of adversarial perturbation learning.

The discriminator is not implemented as an additional binary classifier. Instead, the predictor's evolution-mode estimator plays the role of discrimination by assigning different smooth/deviation-amplifying probabilities to normal and anomaly-aware historical views.

The overall training alternates between predictor optimization and perturbation generator optimization. Through this adversarial process, EVOLVE learns variable-conditioned perturbations that are challenging enough to improve anomaly sensitivity, while maintaining plausible evolution patterns for unsupervised anomaly prediction.

TABLE I
ANOMALY PREDICTION PERFORMANCE OF ALL MODELS ON SIX DATASETS

Dataset	MSL			PSM			SMD			SWAN			GECCO			Genisis		
Metric	Aiff-F1	V-ROC	V-PR	Aiff-F1	V-ROC	V-PR	Aiff-F1	V-ROC	V-PR	Aiff-F1	V-ROC	V-PR	Aiff-F1	V-ROC	V-PR	Aiff-F1	V-ROC	V-PR
LODA	63.78	53.53	10.57	61.02	48.28	30.41	57.93	39.14	5.34	37.11	18.38	30.04	39.04	43.27	2.23	54.25	46.62	0.93
IForest	66.59	53.31	9.17	65.25	50.17	31.86	58.69	39.91	3.53	30.00	27.45	32.54	46.49	39.86	1.93	61.25	50.06	1.41
TranAD	65.72	52.90	16.77	64.01	43.13	34.31	63.43	42.55	4.53	51.30	37.07	30.44	41.13	22.76	1.66	63.26	64.33	0.94
DCdetector	59.41	49.68	10.69	66.08	37.49	27.02	65.51	40.43	4.69	52.66	32.92	32.99	52.62	41.21	1.71	67.56	53.82	1.25
ATransformer	54.57	48.66	13.81	53.20	43.56	30.09	47.70	47.28	4.08	40.99	27.51	30.84	49.58	43.54	0.84	45.26	42.20	0.35
CATCH	68.13	54.89	17.39	66.78	45.62	32.55	64.28	46.58	5.70	55.77	35.00	34.73	61.05	47.49	2.33	70.77	61.60	1.24
CrossAD	<u>68.35</u>	52.19	17.41	68.16	44.70	28.95	64.84	45.52	5.09	52.41	34.06	35.58	60.52	41.28	<u>2.42</u>	71.11	61.68	1.40
PAD	62.91	57.42	19.82	58.72	38.54	31.15	62.95	46.16	4.79	52.72	38.32	34.10	58.14	46.12	1.40	60.47	54.92	0.65
A2P	67.93	59.62	17.80	<u>67.46</u>	48.26	35.92	65.54	<u>49.97</u>	5.33	<u>58.85</u>	39.05	36.31	63.71	<u>49.69</u>	1.85	72.88	63.12	0.99
IGCL	66.33	<u>63.07</u>	20.95	63.79	<u>51.58</u>	36.03	67.39	49.88	<u>6.28</u>	55.11	35.20	34.88	60.04	45.14	2.11	<u>74.11</u>	<u>70.19</u>	1.23
RED-F	64.06	60.63	<u>21.21</u>	69.97	45.71	34.35	<u>67.94</u>	42.20	6.20	58.47	<u>40.11</u>	36.08	61.41	47.51	2.07	68.97	61.69	1.36
TAP	67.39	62.17	19.34	67.29	49.15	<u>36.04</u>	67.16	49.16	6.16	55.56	35.09	<u>37.73</u>	<u>64.46</u>	45.71	1.93	66.91	69.77	<u>1.76</u>
FCM	66.26	57.51	17.62	60.66	49.30	32.58	64.71	48.87	5.87	55.94	39.49	35.62	61.30	46.99	2.13	60.86	59.99	1.43
EVOLVE	72.70	70.23	23.81	73.30	52.70	36.12	72.88	57.77	6.93	65.08	42.47	37.81	65.81	65.56	3.98	76.94	72.00	2.08

G. Inference and Anomaly Scoring

During inference, EVOLVE estimates future anomaly risk using only the normal historical view. Anomaly-aware perturbation injection, Anomaly Pattern Analysis (APA), and dynamic prototype updating are disabled. Therefore, inference does not use future observations and does not construct anomaly-aware historical views, avoiding any test-time access to future information.

Given a historical window $X_{1:T}$, EVOLVE first applies instance normalization, embedding, and the time series encoder to obtain the normal historical representation H_T^n . The representation is then fed into UALM, which samples K latent semantic hypotheses:

$$\mathbf{Z}_K^n = \{(\mathbf{Z}^n)^{(k)}\}_{k=1}^K. \quad (41)$$

For each latent hypothesis, UALM also produces an evolution-mode probability over the smooth evolution hypothesis and the deviation-amplifying evolution hypothesis. These latent hypotheses are decoded by DSD to generate K plausible future trajectories:

$$\hat{\mathbf{X}}_F^n = \{(\hat{\mathbf{X}}_F^n)^{(k)}\}_{k=1}^K. \quad (42)$$

Unlike training, the true future window $X_{T+1:T+F}$ is unavailable during inference, so reconstruction errors cannot be used for anomaly scoring. Instead, EVOLVE estimates anomaly risk from the predictive disagreement among the K decoded future trajectories. Specifically, the anomaly score at each future time step is computed by the predictive variance defined in Eq. (2). A low predictive variance indicates that multiple plausible latent hypotheses lead to consistent future trajectories, suggesting stable future evolution. In contrast, a high predictive variance indicates that plausible future trajectories diverge, suggesting uncertain evolution and higher anomaly risk.

Finally, binary future anomaly labels are obtained by thresholding the anomaly scores:

$$\hat{y}_t = \mathbb{I}(s_t \geq \tau), \quad t = T+1, \dots, T+F, \quad (43)$$

where τ is the decision threshold and $\mathbb{I}(\cdot)$ denotes the indicator function.

IV. EXPERIMENTS

This section evaluates EVOLVE on six real-world multivariate time series benchmarks. We compare EVOLVE with both adapted anomaly detection methods and anomaly prediction methods, and analyze its effectiveness, efficiency, and component contributions. The implementation and supplemental experimental materials are publicly available online.¹

A. Experimental Setup

1) *Datasets and Protocol*: We evaluate EVOLVE on six real-world multivariate time series datasets: MSL [25], PSM [26], SMD [14], SWAN [27], GECCO [27], and Genisis [28]. These datasets cover diverse industrial and cyber-physical scenarios, including spacecraft telemetry, server machines, space weather monitoring, water treatment systems, and cyber-physical production systems. Following previous work [29], only the first continuous segment of MSL is retained. Detailed dataset statistics, including the number of variables, training length, test length, and anomaly ratio, are provided in the supplemental materials.

For each dataset, we construct samples using a historical window of length T and predict anomaly risks over a future horizon of length F . Unless otherwise specified, we set $T = 96$ and evaluate four prediction horizons $F \in \{32, 64, 128, 192\}$. During training, future observations are used only as self-supervised signals, while future anomaly labels are not used. During inference, only the historical window is available, and anomaly scores are computed from the predictive variance among decoded future trajectories. Missing values, if any, are handled by the standard preprocessing protocol used in prior time series anomaly detection benchmarks, and all input windows are normalized before being fed into the model.

2) *Baselines*: We compare EVOLVE with representative methods from two categories.

Adapted anomaly detection methods. Following the adaptation protocols used in previous anomaly prediction stud-

¹<https://github.com/jianxinjin038-sketch/Evolve>

ies [7], [8], we include widely used anomaly detection methods and adapt them to the anomaly prediction setting. These baselines include statistical and density-based methods, i.e., LODA [30] and IForest [31]; reconstruction- or forecasting-based deep models, i.e., TranAD [32], Anomaly Transformer [33], CATCH [34], and CrossAD [24]; and contrastive learning-based models, i.e., DCdetector [35].

Anomaly prediction methods. We further compare EVOLVE with methods specifically designed for anomaly prediction, including PAD [20], A2P [36], IGCL [37], RED-F [10], TAP [7], and FCM [21]. These methods cover precursor-based, perturbation-based, contrastive forecasting, and future-context modeling paradigms for anomaly prediction. For all baselines, we use official implementations when available and follow the recommended configurations. When direct anomaly prediction outputs are unavailable, we adapt the methods following protocols established by previous AP studies.

3) *Evaluation Metrics:* We report Affiliation-F1 (Aff-F1) [38], VUS-ROC, and VUS-PR [39], [40] as the primary metrics in the main tables, since anomaly prediction is highly imbalanced and temporally structured. Aff-F1 evaluates the temporal affiliation between predicted and ground-truth anomaly events, making it more suitable than point-wise F1 when predictions are slightly shifted in time. VUS-ROC and VUS-PR provide range-aware evaluation over different temporal ranges and thresholds, with VUS-PR being particularly informative under severe class imbalance.

4) *Implementation Details:* All experiments are implemented in PyTorch and conducted on an NVIDIA GeForce RTX 3090 GPU with 24GB memory. We use the Adam optimizer [41] with a learning rate of 0.001 for training. The random seed is fixed to 2026 for reproducibility. Following the testing protocol proposed in TFB [42], we disable the `drop_last` operation during evaluation to avoid evaluation bias caused by inconsistent test batch sizes.

For EVOLVE, the size of the dynamic prototype memory is set to 100. The number of latent hypotheses K , the number of clusters in APA, and the batch size are treated as hyperparameters. The initial batch size is set to 64 and is halved when an out-of-memory error occurs, with a minimum batch size of 8. The dynamic prototype bank is initialized as an empty FIFO queue and is updated only during training. During inference, anomaly-aware perturbation injection and dynamic prototype updating are disabled.

Unless otherwise specified, equal weights are assigned to the loss terms in the predictor objective. For all baseline methods, we use their official or open-source implementations whenever available, and follow the configurations recommended in the original papers. All methods are evaluated under the same hardware environment and testing protocol for fair comparison.

B. Main Results

We conduct a systematic evaluation on six multivariate time series datasets under four prediction horizons, i.e., $F \in$

$\{32, 64, 128, 192\}$. The historical window is fixed as $T = 96$ for all datasets and methods. For clarity, Table I reports the average performance across the four prediction horizons, while full horizon-specific results are provided in the supplemental materials.

As shown in Table I, EVOLVE achieves the best average performance on all six datasets across the three primary metrics, including Aff-F1, VUS-ROC, and VUS-PR. This demonstrates that EVOLVE can better localize future anomaly events and more reliably rank future anomaly risks, which is essential for early-warning systems in real-world applications.

Compared with adapted anomaly detection methods, EVOLVE shows consistent advantages across diverse datasets. Although methods such as CATCH, CrossAD, DCdetector, and Anomaly Transformer achieve competitive performance in some cases, they are primarily designed to detect anomalies from observed windows rather than predict future abnormal evolution. Thus, directly adapting them to AP can be insufficient when future anomaly risks depend on uncertain and heterogeneous evolution patterns.

Compared with anomaly prediction baselines, including PAD, A2P, IGCL, RED-F, TAP, and FCM, EVOLVE also achieves stronger performance. This indicates that existing precursor-based, perturbation-based, and future-context modeling methods still have limitations in capturing variable-conditioned anomaly responses and uncertain future evolution. In contrast, EVOLVE constructs anomaly-aware historical views through adaptive prototype retrieval and estimates anomaly risk from the predictive disagreement among multiple decoded future trajectories, enabling more reliable future anomaly prediction.

C. Ablation Studies

We conduct ablation studies on all six datasets to evaluate the contribution of each major component in EVOLVE. Due to space constraints, Tables II–IV report representative results on MSL, SMD, and SWAN, while the complete six-dataset ablation results are provided in the supplemental materials. These three datasets cover different application domains and anomaly characteristics, including spacecraft telemetry, server-machine monitoring, and space weather monitoring, providing a compact yet diverse evaluation of the component contributions. The full results in the supplemental materials show consistent trends across all datasets.

Table II studies the effectiveness of the dual-structured decoder. We compare four variants: removing both evolution hypotheses, using only the smooth evolution hypothesis, using only the deviation-amplifying evolution hypothesis, and using the full dual-structured decoder. The results show that using either hypothesis alone is insufficient to consistently achieve the best performance. The smooth evolution hypothesis helps model stable future dynamics, while the deviation-amplifying evolution hypothesis captures potential abnormal dynamics induced by local deviations. Combining them leads to the best overall performance, demonstrating that the two evolution

hypotheses provide complementary information for multi-hypothesis future modeling.

TABLE II
ABLATION OF THE DUAL-STRUCTURED DECODER.

Dataset		MSL			SMD			SWAN		
Smooth	Alert	AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR
$\times$	$\times$	70.21	65.41	20.88	72.65	54.65	6.85	63.40	40.02	35.96
$\checkmark$	$\times$	68.75	64.56	20.55	70.97	56.40	6.76	63.36	39.26	35.16
$\times$	$\checkmark$	68.78	63.59	18.55	72.05	55.97	6.73	61.10	42.08	36.52
$\checkmark$	$\checkmark$	72.70	70.23	23.81	72.88	57.77	6.93	65.08	42.47	37.81

Table III evaluates the contribution of key components in EVOLVE. Removing adaptive anomaly injection degrades performance, indicating that variable-conditioned perturbation modeling is important for capturing heterogeneous anomaly responses. Removing Anomaly Pattern Analysis also leads to performance degradation, showing that dynamically updated residual-based prototypes provide useful context-dependent anomaly patterns beyond fixed priors. When Uncertainty-Aware Latent Modeling is removed, the performance decreases, verifying the importance of sampling multiple latent semantic hypotheses for uncertain future evolution. Removing adversarial optimization further weakens the model, suggesting that hard but plausible perturbations help improve the discrimination between stable and deviation-sensitive future evolution. Overall, the full EVOLVE model achieves the best or competitive results, confirming the complementary roles of these components.

TABLE III
ABLATION OF KEY COMPONENTS.

Dataset		MSL			SMD			SWAN		
Metric		AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR
w/o Dynamically Adaptive Anomalies		70.51	64.63	21.24	70.57	51.31	6.26	59.50	41.58	36.55
w/o Anomaly Pattern Analysis		70.57	65.99	20.98	71.54	56.85	6.62	60.21	40.62	36.06
w/o Adaptive Latent Modeling		69.55	66.09	21.55	71.07	56.03	6.65	61.92	42.12	36.51
w/o Adversarial Anomaly Generation		70.21	63.35	20.84	70.78	57.06	6.50	62.68	39.94	36.06
EVOLVE		72.70	70.23	23.81	72.88	57.77	6.93	65.08	42.47	37.81

TABLE IV
ABLATION OF TRAINING LOSSES.

Dataset		MSL			SMD			SWAN		
Metric		AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR	AFF-F1	V-ROC	V-PR
w/o MSE Loss		70.44	65.66	21.01	71.96	55.68	6.61	64.88	40.47	35.67
w/o VAR Loss		69.54	64.51	20.17	70.90	56.23	6.36	60.80	41.98	35.24
w/o Weights Loss		69.03	66.05	20.11	70.55	54.45	6.19	62.52	41.25	36.60
EVOLVE		72.70	70.23	23.81	72.88	57.77	6.93	65.08	42.47	37.81

Table IV analyzes the effect of different training objectives. Removing the MSE-based objectives weakens the alignment between decoded trajectories and observed future dynamics. Removing the variance-based objectives reduces the model’s ability to distinguish stable future evolution from uncertain or deviation-sensitive evolution. Removing the evolution-mode alignment loss also leads to performance degradation, indicating that explicitly aligning normal historical views with smooth evolution and anomaly-aware historical views with deviation-amplifying evolution is beneficial. These results demonstrate that the proposed training objectives jointly

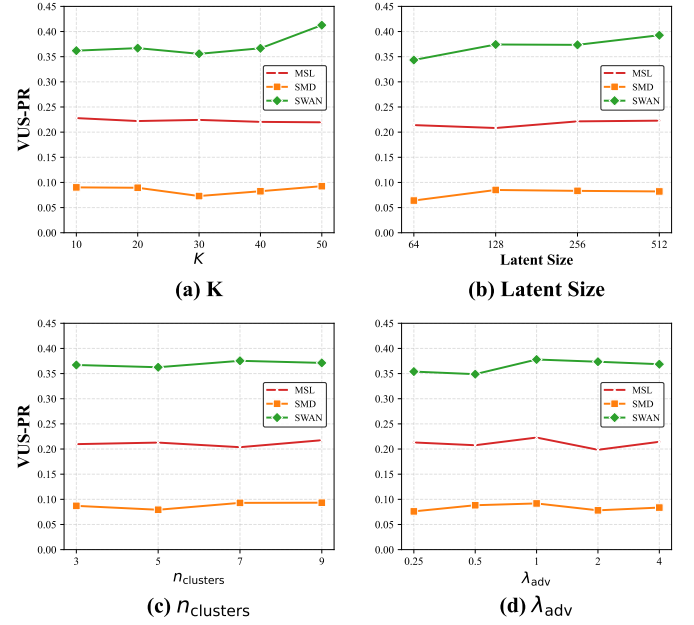


Fig. 3. Parameter sensitivity analysis of EVOLVE on MSL, SMD, and SWAN using VUS-PR.

contribute to accurate, diverse, and semantically consistent anomaly prediction.

D. Parameter Sensitivity

We further analyze the sensitivity of EVOLVE to several key hyperparameters, including the number of latent hypotheses K , the latent dimension, the number of anomaly prototype clusters $n_{clusters}$, and the adversarial loss weight λ_{adv} . Experiments are conducted on MSL, SMD, and SWAN using VUS-PR as the evaluation metric.

As shown in Figure 3, EVOLVE maintains relatively stable performance across different hyperparameter settings. Varying K does not lead to drastic performance changes, indicating that a moderate number of latent hypotheses is sufficient to capture future uncertainty. The model is also not overly sensitive to the latent dimension or the number of anomaly prototype clusters, suggesting that UALM and APA remain effective under different representation capacities and clustering granularities. Similarly, changing λ_{adv} only causes limited fluctuations, indicating that adversarial perturbation learning improves anomaly-aware representation learning without destabilizing training.

Overall, these results show that EVOLVE is not highly sensitive to the tested hyperparameters and can maintain stable anomaly prediction performance without extensive dataset-specific tuning.

E. Efficiency and Scalability Analysis

We further evaluate the computational efficiency and scalability of EVOLVE on the MSL dataset, with the historical window fixed as $T = 96$ and the prediction horizon fixed as $F = 32$. Table V compares EVOLVE with representative

TABLE V
EFFICIENCY AND EFFECTIVENESS COMPARISON ON MSL.

Method	VUS-ROC	Training Time	Inference Time	Memory Used
IGCL	66.74	14.09s	0.19s	34.12MB
PAD	60.79	99.20s	0.86s	281.13MB
A2P	62.01	34.87s	1.36s	559.77MB
RED-F	54.96	78.54s	2.15s	567.22MB
TAP	71.04	97.33s	1.85s	496.63MB
FCM	57.04	75.56s	1.97s	443.53MB
EVOLVE	76.17	67.21s	0.37s	145.98MB

TABLE VI
SCALABILITY ANALYSIS WITH RESPECT TO THE ANOMALY PROTOTYPE
MEMORY SIZE N_{mem} .

N_{mem}	Training Time	Memory Used	VUS-ROC
50	65.57s	135MB	75.64
100	67.21s	146MB	76.17
200	71.56s	160MB	77.81
300	73.43s	176MB	76.10
400	76.20s	205MB	75.54
500	77.99s	229MB	74.78

anomaly prediction baselines in terms of training time, inference time, memory consumption, and prediction performance measured by VUS-ROC.

As shown in Table V, EVOLVE achieves the best VUS-ROC among all compared methods, outperforming the strongest baseline TAP by 5.13 absolute points. Although EVOLVE requires more training time than lightweight methods such as IGCL and A2P, its training cost is lower than PAD, RED-F, TAP, and FCM. More importantly, EVOLVE maintains low inference latency and moderate memory consumption, with substantially faster inference than PAD, A2P, RED-F, TAP, and FCM. These results indicate that EVOLVE provides a favorable trade-off between predictive performance and computational cost.

The additional training cost mainly comes from two-stage alternating optimization, adaptive prototype updating, and adversarial perturbation learning. During inference, anomaly-aware perturbation injection and dynamic prototype updating are disabled, so the additional inference overhead remains limited. A backbone-controlled comparison is provided in the supplemental materials, showing that the improvement of EVOLVE does not solely come from the encoder backbone.

Table VI analyzes the impact of the maximum size N_{mem} of the anomaly prototype memory. Unless otherwise specified, we set the default memory size to $N_{\text{mem}} = 100$ in all main experiments. As N_{mem} increases, training time and memory consumption grow gradually, while VUS-ROC remains relatively stable. This suggests that EVOLVE is not overly sensitive to memory capacity and can scale to larger prototype memories without causing excessive computational burden. The results also indicate that a compact memory is sufficient to provide useful anomaly prototypes for adaptive perturbation modeling.

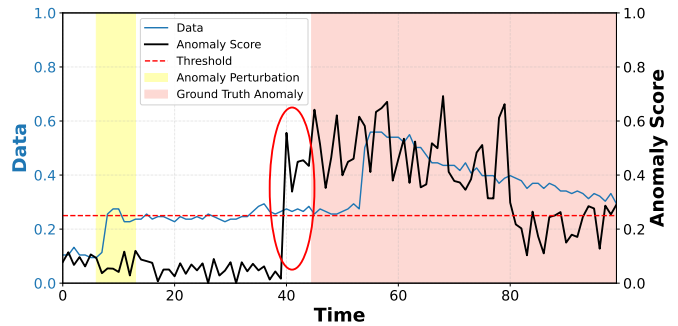


Fig. 4. Visualization of predictive anomaly scores on the SMD dataset.

F. Visualization of Predictive Anomaly Scores

To further illustrate the behavior of the proposed anomaly scoring criterion, Figure 4 visualizes one variable from the SMD dataset together with the predictive variance-based anomaly scores produced by EVOLVE. A mild disturbance appears around $t \approx 10$, while the anomaly score remains relatively stable. This indicates that EVOLVE does not simply react to instantaneous fluctuations in the observed history.

More importantly, the anomaly score rises sharply around $t \approx 40$, before the ground-truth anomaly interval begins at approximately $t \approx 45$. This early increase shows that the proposed scoring criterion can capture future anomaly risk before the abnormal segment fully manifests. After the anomaly develops, the anomaly score remains consistently high, reflecting sustained predictive uncertainty during abnormal evolution.

Overall, the visualization demonstrates the predictive nature of EVOLVE. Instead of relying on post-hoc reconstruction errors from observed anomalies, EVOLVE estimates future anomaly risk from the predictive disagreement among multiple decoded future trajectories, enabling early warning before anomalies become fully observable.

V. CONCLUSION

In this paper, we propose EVOLVE, a unified framework for unsupervised multivariate time series anomaly prediction. EVOLVE models anomaly prediction through variable-conditioned perturbation modeling and uncertainty-aware multi-hypothesis future reasoning. To capture heterogeneous anomaly responses, EVOLVE introduces an adaptive anomaly prototype memory that combines fixed anomaly priors with dynamically updated prototypes, and retrieves context-specific perturbations to construct anomaly-aware historical views. To model uncertain future evolution, EVOLVE samples multiple latent semantic hypotheses and decodes them through smooth and deviation-amplifying evolution hypotheses, using the predictive disagreement among decoded future trajectories to estimate future anomaly risk. Extensive experiments on six real-world benchmark datasets demonstrate that EVOLVE outperforms state-of-the-art anomaly prediction baselines and adapted anomaly detection baselines, showing its effectiveness for early warning and proactive risk mitigation.

REFERENCES

- [1] S. Schmidl, F. Naumann, and T. Papenbrock, “Autotsad: unsupervised holistic anomaly detection for time series data,” *Proceedings of the VLDB Endowment*, vol. 17, no. 11, pp. 2987–3002, 2024.
- [2] A. Zhang, S. Deng, D. Cui, Y. Yuan, and G. Wang, “An experimental evaluation of anomaly detection in time series,” *Proceedings of the VLDB Endowment*, vol. 17, no. 3, pp. 483–496, 2023.
- [3] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, “Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding,” in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 3220–3230.
- [4] Q. Wen, L. Yang, T. Zhou, and L. Sun, “Robust time series analysis and applications: An industrial perspective,” in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 4836–4837.
- [5] P. Boniol, M. Meftah, E. Remy, B. Didier, and T. Palpanas, “dcnn/dcam: anomaly precursors detection in multivariate time series with deep convolutional neural networks,” *Data-Centric Engineering*, vol. 4, p. e30, 2023.
- [6] V. Cerqueira, L. Torgo, and C. Soares, “Early anomaly detection in time series: a hierarchical approach for predicting critical health episodes,” *Machine Learning*, vol. 112, no. 11, pp. 4409–4430, 2023.
- [7] S. Hu, K. Zhao, C. Guo, X. Qiu, Y. Shu, J. Hu, C. S. Jensen, and B. Yang, “Tap: Time series anomaly prediction via adaptive period modeling and dual representation learning,” *IEEE Transactions on Knowledge and Data Engineering*, 2026.
- [8] K. Zhao, Z. Zhuang, C. Guo, H. Miao, C. S. Jensen, Y. Cheng, and B. Yang, “Unsupervised time series anomaly prediction with importance-based generative contrastive learning,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 3945–3956.
- [9] M.-Y. Park, W.-J. Lee, S. T. Kim, and G.-M. Park, “When will it fail?: Anomaly to prompt for forecasting future anomalies in time series,” *arXiv preprint arXiv:2506.23596*, 2025.
- [10] P. Chen, X. Shi, Y. Chang, Y. Sun, and S. K. Das, “Red-f: Reconstruction-elimination based dual-stream contrastive forecasting for multivariate time series anomaly prediction,” *arXiv preprint arXiv:2511.20044*, 2025.
- [11] Z. Xing, J. Pei, and S. Y. Philip, “Early prediction on time series: A nearest neighbor approach,” in *IJCAI*, 2009, pp. 1297–1302.
- [12] X.-X. Yin, Y. Miao, and Y. Zhang, “Time series based data explorer and stream analysis for anomaly prediction,” *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 5885904, 2022.
- [13] B. Lim and S. Zohren, “Time-series forecasting with deep learning: a survey,” *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, vol. 379, no. 2194, 2021.
- [14] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, “Robust anomaly detection for multivariate time series through stochastic recurrent neural network,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.
- [15] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, “Stochastic latent residual video prediction,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 3233–3246.
- [16] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.
- [17] A. Hassan, T. Kumar, A. Mishra, S. Serebryakov, S. K. Mopur, P. Koganti, M. Chelankuri, R. Vogety, S. Bhattacharya, and M. Foltin, “Forecast2anomaly (f2a): Adapting multivariate time series foundation models for anomaly prediction,” *arXiv preprint arXiv:2511.03149*, 2025.
- [18] D. Xu, W. Cheng, J. Ni, D. Luo, M. Natsumeda, D. Song, B. Zong, H. Chen, and X. Zhang, “Deep multi-instance contrastive learning with dual attention for anomaly precursor detection,” in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 91–99.
- [19] J. You, A. Cela, R. Natowicz, J. Ouanounou, and P. Siarry, “Anomaly prediction: A novel approach with explicit delay and horizon,” in *2024 IEEE 20th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2024, pp. 1–8.
- [20] S. Y. Jhin, J. Lee, and N. Park, “Precursor-of-anomaly detection for irregular time series,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 917–929.
- [21] S. Zhao, W. Wang, H. Xu, Z. Yu, Q. Wen, G. Wang, G. Pang *et al.*, “Abnormality forecasting: Time series anomaly prediction via future context modeling,” *arXiv preprint arXiv:2410.12206*, 2024.
- [22] B. Cai, S. Yang, L. Gao, and Y. Xiang, “Hybrid variational autoencoder for time series forecasting,” *Knowledge-Based Systems*, vol. 281, p. 111079, 2023.
- [23] Q. Shentu, B. Li, K. Zhao, Y. Shu, Z. Rao, L. Pan, B. Yang, and C. Guo, “Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders,” *arXiv preprint arXiv:2405.15273*, 2024.
- [24] B. Li, Q. Shentu, Y. Shu, H. Zhang, M. Li, N. Jin, B. Yang, and C. Guo, “Crossad: Time series anomaly detection with cross-scale associations and cross-window modeling,” *arXiv preprint arXiv:2510.12489*, 2025.
- [25] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.
- [26] A. Abdulaal, Z. Liu, and T. Lancewicki, “Practical approach to asynchronous multivariate time series anomaly detection and localization,” in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2485–2494.
- [27] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, “Revisiting time series outlier detection: Definitions and benchmarks,” in *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*, 2021.
- [28] A. von Birgelen and O. Niggemann, “Anomaly detection and localization for cyber-physical production systems with self-organizing maps,” in *IMPROVE-Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency: Intelligent Methods for the Factory of the Future*. Springer Berlin Heidelberg Berlin, Heidelberg, 2018, pp. 55–71.
- [29] C. Wang, Z. Zhuang, Q. Qi, J. Wang, X. Wang, H. Sun, and J. Liao, “Drift doesn’t matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection,” *Advances in neural information processing systems*, vol. 36, pp. 10758–10774, 2023.
- [30] T. Pevný, “Loda: Lightweight on-line detector of anomalies,” *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.
- [31] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth IEEE international conference on data mining*. IEEE, 2008, pp. 413–422.
- [32] S. Tuli, G. Casale, and N. R. Jennings, “Tranad: Deep transformer networks for anomaly detection in multivariate time series data,” *arXiv preprint arXiv:2201.07284*, 2022.
- [33] J. Xu, H. Wu, J. Wang, and M. Long, “Anomaly transformer: Time series anomaly detection with association discrepancy,” *arXiv preprint arXiv:2110.02642*, 2021.
- [34] X. Wu, X. Qiu, Z. Li, Y. Wang, J. Hu, C. Guo, H. Xiong, and B. Yang, “Catch: Channel-aware multivariate time series anomaly detection via frequency patching,” *arXiv preprint arXiv:2410.12261*, 2024.
- [35] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, “Dcdetector: Dual attention contrastive representation learning for time series anomaly detection,” in *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2023, pp. 3033–3045.
- [36] M.-Y. Park, W.-J. Lee, S. T. Kim, and G.-M. Park, “When will it fail?: Anomaly to prompt for forecasting future anomalies in time series,” *arXiv preprint arXiv:2506.23596*, 2025.
- [37] K. Zhao, Z. Zhuang, C. Guo, H. Miao, C. S. Jensen, Y. Cheng, and B. Yang, “Unsupervised time series anomaly prediction with importance-based generative contrastive learning,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 3945–3956.
- [38] A. Huet, J. M. Navarro, and D. Rossi, “Local evaluation of time series anomaly detection algorithms,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 635–645.
- [39] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin, “Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection,” *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2774–2787, 2022.
- [40] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. J. Elmore, and M. J. Franklin, “Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection,” *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 2774–2787, 2022.

- [41] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [42] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng *et al.*, "Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods," *arXiv preprint arXiv:2403.20150*, 2024.