NORMALITY CONSTRAINT LEARNING: ADAPTING FOUNDATION MODELS FOR TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

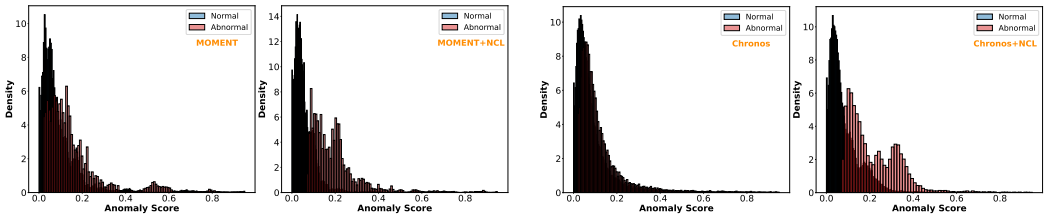
Time Series Foundation Models (TSFMs) achieve strong generalization by learning to reconstruct or forecast broad temporal patterns from large-scale time series during pre-training. Yet this strength can become a weakness for anomaly detection: TSFMs may model rare anomalous patterns as effectively as normal ones, allowing anomalies to be accurately reconstructed or forecasted and thus diminishing their reconstruction/forecasting error-based anomaly scores. This paper proposes **Normality Constraint Learning** (**NCL**), a lightweight plug-and-play framework that adapts pre-trained TSFMs for accurate anomaly detection without modifying their pre-trained parameters. Our key insight is to constrain the broad pattern space of TSFMs to the normal structure of a target time series, preventing their broad modeling capability from obscuring abnormal deviations. Specifically, NCL constructs a compact normality subspace from a few normal patch features and adaptively steers each patch feature toward normality within this subspace, guided by contrastive constraints that form compact and discriminative normality manifolds. The calibrated features are aggregated to reinforce normal components and fused with the original TSFM output, amplifying the discrepancy between normal and abnormal observations for the reconstruction/forecasting error-based anomaly scoring. Extensive experiments across diverse TSFM families and benchmarks show that NCL consistently improves anomaly detection performance, providing a generalizable framework for adapting TSFMs to anomaly detection.

1 INTRODUCTION

Time series foundation models (TSFMs) have recently delivered substantial gains in the landscape of time series analysis. Pre-trained on massive and diverse corpora, TSFMs such as Aurora (Wu et al., 2026), ROSE (Wang et al., 2025), Chronos (Ansari et al., 2024), and MOMENT (Goswami et al., 2024) encode rich universal representations of temporal dynamics and demonstrate strong zero-shot and few-shot generalization across various downstream tasks. Time series anomaly detection (TSAD) is one of the key downstream tasks of TSFMs. It aims to identify deviations from expected patterns, playing a critical role in many real-world applications such as industrial control (Wang et al., 2022), disease identification (Pereira & Silveira, 2019), and fraud detection (Anandakrishnan et al., 2018).

Although TSFMs have strong expressive power, their design is fundamentally reconstruction- or forecasting-centric, where internal features are shaped for broad pattern modeling. Since such general-purpose representation capability can often efficiently model both normal and abnormal patterns, the existing common practice of directly repurposing the reconstruction/forecasting errors of TSFMs for anomaly detection is questionable. As illustrated in Figure 1, both reconstruction-based and forecasting-based TSFMs can effectively reconstruct or forecast most of the abnormal observations, resulting in small reconstruction/forecasting errors for not only normal data but also anomalies. This phenomenon fundamentally undermines the widely-used TSAD intuition that anomalies are harder to reconstruct or forecast than normal data (Pang et al., 2021; Zamanzadeh Darban et al., 2024; Liu & Paparrizos, 2024; Zhu et al., 2026). We also provide a theoretical analysis of this issue in Appendix A.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Anomaly scores for reconstruction-based TSFMs. (b) Anomaly scores for forecasting-based TSFMs.

Figure 1: (a) and (b) respectively show the anomaly score distributions of normal and abnormal observations for reconstruction-based and forecasting-based TSFMs and their NCL-augmented counterparts on the PSM dataset (Abdulaal et al., 2021). NCL enables both types of TSFMs to produce substantially more discriminative anomaly scores.

To address this challenge, a natural idea is to adapt TSFMs for anomaly detection through fine-tuning strategies (Hu et al., 2022; Qiu et al., 2023; Park & Yoon, 2026). However, fine-tuning pre-trained foundation models risks catastrophic forgetting of the rich temporal knowledge, on which the anomaly scoring mechanisms in the fine-tuned TSFMs still heavily rely when computing the reconstruction or forecasting errors. This raises a critical question: *How can we effectively leverage the powerful representational capacity of frozen TSFMs while steering their outputs toward a normality regime so that abnormal deviations stand out more clearly?*

In this paper, we propose **Normality Constraint Learning** (**NCL**), a lightweight plug-and-play framework that can augment diverse TSFMs for accurate anomaly detection without modifying their pre-trained parameters. Our key insight is that, while TSFMs can capture a broad spectrum of temporal patterns, reconstruction/forecasting-error-based anomaly detection requires identifying and emphasizing the relevant normal patterns only; we therefore first construct a normality subspace from this broad pattern space and then adaptively steer each patch feature toward normality within the subspace, guided by compact and discriminative manifolds learned from a few normal patch features.

To this end, NCL is instantiated with three main components, namely normal subspace construction (**NormSpace**), patch-adaptive normality transformation (**PatchNT**), and contrastive normality constraint (**ConNC**). NormSpace constructs a compact normality subspace via singular value decomposition (SVD) of patch features from a few training subsequences encoded by a frozen TSFM. Within this subspace, PatchNT adaptively calibrates patch features toward a learnable normality manifold along each principal direction, while ConNC guides this adaptive transformation by encouraging the formation of compact and discriminative normality manifolds. The calibrated features across all normality directions are then aggregated into consensus features that reinforce normal components. Finally, these features are decoded and fused with the original TSFM output to enhance the normal patterns, thereby amplifying the discrepancy between normal and abnormal data. As shown in Figure 1, augmenting TSFMs with NCL yields more discriminative anomaly score distributions than vanilla TSFMs (see Appendix K for additional results). In summary, our contributions are as follows:

- We propose NCL, a novel plug-and-play framework that augments frozen TSFMs with normality constraints to learn calibrated normality features, thereby substantially sharpening the reconstruction/forecasting error-based anomaly scores of the TSFMs.
- We further instantiate NCL with three components: NormSpace, PatchNT, and ConNC. NormSpace constructs a normality subspace, while PatchNT adaptively transforms patch features toward the normality regime. This transformation is guided by ConNC to encourage the formation of compact yet discriminative normality manifolds within the regime.
- Extensive experiments on popular TSAD datasets demonstrate that NCL consistently and significantly improves the anomaly detection performance of diverse families of TSFMs.

2 Related Work

2.1 Time Series Anomaly Detection

Conventional TSAD approaches rely on statistical techniques and machine learning methods, which detect anomalies based on deviation from statistical properties or density estimates (Breunig et al.,

2000; Yairi et al., 2001; Liu et al., 2008; Hariri et al., 2019). Recently, the advent of deep learning has significantly advanced TSAD capabilities. Reconstruction-based methods (Li et al., 2025; Wu et al., 2025; Zhang et al., 2026a; Hu et al., 2026) learn to reconstruct normal patterns and identify anomalies through large reconstruction errors, while forecasting-based approaches (Liu et al., 2025; Cho et al., 2025; Zhou et al., 2026a) detect anomalies by measuring discrepancies between predicted and actual future observations. Meanwhile, contrastive learning (Luo & Wang, 2026; Mou et al., 2026; Park & Kang, 2026) has emerged as a powerful paradigm for learning discriminative representations.

Despite these advances, existing methods typically require training a separate model for each new dataset or application , *i.e.*, they are trained from scratch on large-scale domain-specific normal data and evaluated on the testing dataset from the same domain. This strong dependence on abundant in-domain data severely limits their applicability to scenarios where such data are scarce or costly to obtain (Shentu et al., 2025; Lan et al., 2026; He et al., 2026).

2.2 Time Series Foundation Models for Anomaly Detection

The recent emergence of TSFMs provides a generalization path, modeling universal temporal representations through large-scale time series pre-training. Most existing TSFMs (Gao et al., 2024; Shi et al., 2025; Wu et al., 2026) are task-general, which are pre-trained on massive data with general objectives (*e.g.*, reconstruction or forecasting), and are applied to various downstream tasks such as forecasting and anomaly detection. However, directly applying TSFMs to anomaly detection presents fundamental challenges. As noted by recent studies (Liu & Paparrizos, 2024; Zhu et al., 2026), task-general TSFMs effectively model both normal and anomalous patterns. This undermines the widely-used TSAD intuition that anomalies are harder to reconstruct or forecast than normal data.

More recently, TSAD-specific foundation models have also been explored (Shentu et al., 2025; Lan et al., 2026; He et al., 2026). For instance, DADA (Shentu et al., 2025) utilizes injected anomalies to explicitly distinguish between normal and abnormal patterns, while TimeRCD (Lan et al., 2026) conducts relative context discrepancy pre-training on synthetic anomalies. TimeRadar (He et al., 2026) is focused on leveraging these synthetic anomalies to enforce contextual deviations. These methods require mass synthetic anomalies to enrich the pre-training corpus. However, Low-quality anomalies may cause the foundation models to specialize in artificial deviations that do not transfer to real-world distributions (Xu et al., 2024; Zhou et al., 2026b), thereby leading to high detection errors.

3 Preliminaries

Time Series Anomaly Detection (TSAD). Given a time series $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{D \times T}$ with D channels and T ordered observations, the task of TSAD is to calculate an anomaly score $s_t \in \mathbb{R}$ for each $\mathbf{x}_t \in \mathbb{R}^D$ at timestamp t. The optimal goal is to assign each anomalous observation a larger anomaly score than all the normal observations.

Time Series Foundation Models for Anomaly Detection. Most TSFMs rely on data reconstruction and/or forecasting objectives in their training, and furthermore, reconstruction and forecasting errors are widely-used effective anomaly measures in TSAD. Thus, these models are the backend TSFMs in this work. Formally, a TSFM $\mathcal{F}$ typically regards a time series subsequence $\mathbf{X}_{t-L:t} = (\mathbf{x}_{t-L}, \mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_{t-1}) \in \mathbb{R}^{D \times L}$ as a sample input, and produces either a reconstruction of the input $\hat{\mathbf{X}}_{t-L:t}$ or a forecast of future observations $\hat{\mathbf{X}}_{t:t+F}$ over a horizon F. For notational brevity, we unify both paradigms by letting $\mathbf{Y}_t$ denote the prediction target, *i.e.*, $\mathbf{Y}_t = \mathbf{X}_{t-L:t}$ under reconstruction and $\mathbf{Y}_t = \mathbf{X}_{t:t+F}$ under forecasting. The corresponding model output is formulated as $\hat{\mathbf{Y}}_t = \mathcal{F}(\mathbf{X}_{t-L:t})$. The pointwise anomaly score is then derived from the discrepancy between $\mathbf{Y}_t$ and $\hat{\mathbf{Y}}_t$, commonly measured by the squared error $s_t = \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2$, where $\mathbf{y}_t \in \mathbf{Y}_t$ and $\hat{\mathbf{y}}_t \in \hat{\mathbf{Y}}_t$.

4 Methodology

4.1 Overview of NCL: Toward Normality-Constrained TSFMs

TSFMs are pre-trained on large corpora to learn a variety of universal temporal patterns. However, this generality allows them to reconstruct or forecast normal and anomalous data similarly well,

so anomalies may not yield larger errors. To mitigate this issue, we introduce the plug-and-play **normality constraint learning (NCL)** module $\mathcal{G}$ to achieve normality-constrained TSFMs by steering the representational capacity of a TSFM $\mathcal{F}$ toward the normal regime out of its broad pattern space.

As illustrated in Figure 2, NCL consists of three main components. Given a few normal training subsequences, its first component, ① **Normal Subspace Construction (NormSpace)**, identifies k principal normal directions via SVD on their patch features $\mathbf{Z}$ extracted from the frozen TSFM $\mathcal{F}$. The ② **Patch-Adaptive Normality Transformation (PatchNT)** component is then devised to enforce a lightweight yet patch-adaptive alignment of each feature with a learnable normality manifold along each principal direction. The ③ **Contrastive Normality Constraint (ConNC)** component is further proposed to constrain the transformed features tightly cluster around their respective normality modes. In doing so, NCL amplifies the discrepancy signal between normal and abnormal observations that the superior fitting capacity of TSFMs would otherwise diminish. All these three components are done without changing any of the pre-trained parameters of $\mathcal{F}$ itself. Let $\mathcal{F}'$ denote the frozen TSFM augmented with the proposed NCL module, the predictions can then be defined as $\tilde{\mathbf{Y}}_t = \mathcal{F}'(\mathbf{X}_{t-L:t})$.

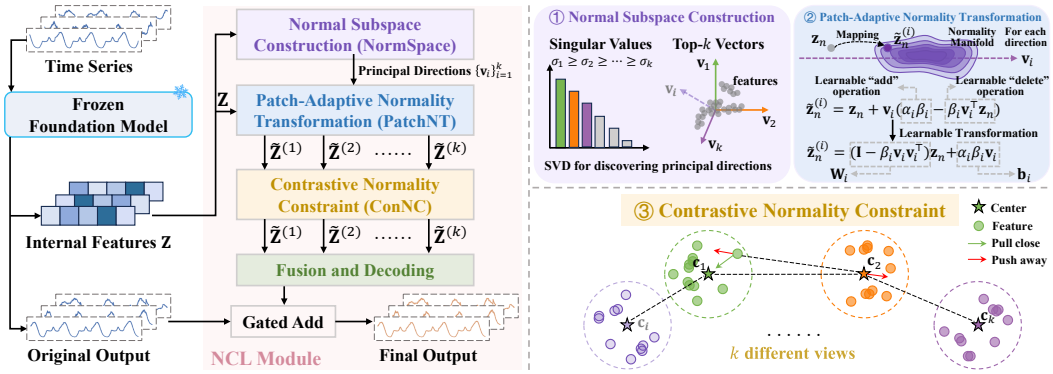


Figure 2: Overview of NCL. ① Given few-shot normal training subsequences, a frozen TSFM is used to extract their patch features $\mathbf{Z}$, upon which NormSpace constructs a normal subspace of k principal directions via SVD. ② PatchNT is then trained to perform patch-adaptive transformation of each feature onto a learnable normality manifold along each principal direction. ③ ConNC is further leveraged to calibrate the transformed features via constraining their distributions tightly cluster around the normal prototypes $\{\mathbf{c}_i\}_{i=1}^k$. Finally, these calibrated features are synthesized with the original TSFM output to amplify the error discrepancy between normal and abnormal data.

4.2 NormSpace: Normal Subspace Construction via Few-Shot SVD

Let $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]^\top \in \mathbb{R}^{N \times d}$ be the matrix of normal patch features extracted from $\mathcal{F}$, where N is the total number of patches and d is the feature dimension. To identify the most informative directions of variation within the normal feature space, we perform SVD on $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{1}$$

where $\mathbf{U}_r \in \mathbb{R}^{N \times r}$, $\mathbf{\Sigma}_r = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, and $\mathbf{V}_r \in \mathbb{R}^{d \times r}$, with $r = \mathrm{rank}(\mathbf{Z})$. The columns of $\mathbf{V}_r$, denoted $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r\}$, are the right singular vectors and correspond to the principal directions of $\mathbf{Z}$.

The singular vectors with the largest singular values capture the dominant and stable patterns shared across all N normal patch features, *i.e.*, the directions along which normal temporal structure is most concentrated. Therefore, we construct a normal-pattern-focused space by retaining the top-k right singular vectors:

$$\{\mathbf{v}_i\}_{i=1}^k, \quad k \ll d, \tag{2}$$

which collectively span a k-dimensional *normal subspace*. These vectors are stored as running buffers during inference, ensuring that the normal subspace is estimated from the training distribution.

We theoretically show in Appendix B that **(i)** performing SVD on the patch features from few-shot normal samples is sufficient to construct an informative normality space and **(ii)** choosing the top-k vectors to span a k-dimensional normal subspace is most effective.

4.3 PATCHNT: LIGHTWEIGHT PATCH-ADAPTIVE NORMALITY TRANSFORMATION

The normal subspace spanned by $\{\mathbf{v}_i\}_{i=1}^{k}$ captures the principal directions of normal temporal patterns, but SVD projection alone only preserves feature components along these directions without explicitly aligning them with normality. Moreover, different patches may deviate from normality in different ways and to different degrees, requiring patch-specific calibration. We therefore introduce patch-adaptive normality transformation (PatchNT), which uses only a few learnable parameters to selectively add or delete information from each patch feature, steering its component along $\mathbf{v}_i$ via these learnable "add" and "delete" operations toward a normality-consistent representation. Applying PatchNT separately along the k directions then yields k complementary views, each calibrating the patch feature toward the normal temporal structure captured along one direction.

Formally, for each patch feature $\mathbf{z}_n \in \mathbb{R}^d$, the i-th normality transformation proceeds as follows. First, the scalar projection of $\mathbf{z}_n$ onto $\mathbf{v}_i$ is computed:

$$p_n^{(i)} = \mathbf{v}_i^\top \mathbf{z}_n. \tag{3}$$

Next, inspired by deep delta learning (Zhang et al., 2026b) in residual update, two patch-adaptive vector parameters are defined to specify the learnable "add" and "delete" operations for steering patch feature along each direction $\mathbf{v}_i$. Specifically, the target value $\boldsymbol{\alpha}$ and the calibration strength $\boldsymbol{\beta}$ are produced by lightweight linear heads applied to $\mathbf{z}_n$ for the feature steering in all k directions:

$$\boldsymbol{\alpha} = \sigma(\mathbf{W}_\alpha \mathbf{z}_n + \mathbf{b}_\alpha) \in (0,1)^k, \quad \boldsymbol{\beta} = 2 \cdot \sigma(\mathbf{W}_\beta \mathbf{z}_n + \mathbf{b}_\beta) \in (0,2)^k, \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{W}_\alpha, \mathbf{W}_\beta \in \mathbb{R}^{k \times d}$ are the learnable weights. Each target value $\alpha_i \in \boldsymbol{\alpha}$ specifies where the calibrated feature should lie along $\mathbf{v}_i$, while $\beta_i \in \boldsymbol{\beta}$ controls the degree of calibration. Then we can obtain the calibrated feature as follows:

$$\tilde{\mathbf{z}}_n^{(i)} = \mathbf{z}_n + \beta_i \mathbf{v}_i(\alpha_i - p_n^{(i)}) = \mathbf{z}_n + \beta_i \mathbf{v}_i\big(\alpha_i - \mathbf{v}_i^\top \mathbf{z}_n\big) = \mathbf{z}_n + \mathbf{v}_i\big(\underbrace{\alpha_i \beta_i}_{\text{add}} - \underbrace{\beta_i \mathbf{v}_i^\top \mathbf{z}_n}_{\text{delete}}\big), \tag{5}$$

indicating that the introduced patch-adaptive parameters can achieve both information addition and deletion along the direction $\mathbf{v}_i$. However, it is still unclear to what extent the β_i will modify the relative distances of the feature from α_i in the direction $\mathbf{v}_i$. We further treat the calibration in Equation (5) as a rank-1 affine update:

$$\tilde{\mathbf{z}}_n^{(i)} = \mathbf{z}_n + \mathbf{v}_i\big(\alpha_i \beta_i - \beta_i \mathbf{v}_i^\top \mathbf{z}_n\big) = (\mathbf{I} - \beta_i \mathbf{v}_i \mathbf{v}_i^\top)\mathbf{z}_n + \alpha_i \beta_i \mathbf{v}_i = \mathbf{W}_i \mathbf{z}_n + \mathbf{b}_i, \tag{6}$$

where $\mathbf{W}_i = \mathbf{I} - \beta_i \mathbf{v}_i \mathbf{v}_i^\top$ and $\mathbf{b}_i = \alpha_i \beta_i \mathbf{v}_i$. Based on Equation (6), we can then characterize how β_i governs the geometric behavior of the transformation by analyzing the spectral structure of $\mathbf{W}_i$. We further propose Proposition 1 (see Appendix C for its proof).

Proposition 1 (Spectral and Geometric Property)**.** *The spectrum of* $\mathbf{W}_i = \mathbf{I} - \beta_i \mathbf{v}_i \mathbf{v}_i^\top$ *is* $\Lambda(\mathbf{W}_i) = \{1 - \beta_i, 1, \ldots, 1\}$ *with one eigenvalue* $(1 - \beta_i)$ *for eigenvector* $\mathbf{v}_i$ *and* $(d-1)$ *eigenvalues of* 1 *in* $\mathbf{v}_i^\perp = \{\mathbf{k} \in \mathbb{R}^d \mid \mathbf{v}_i^\top \mathbf{k} = 0\}$. *At* $\beta_i = 1$*, the affine update* $\tilde{\mathbf{z}}_n^{(i)} = \mathbf{W}_i \mathbf{z}_n + \mathbf{b}_i$ *exactly deletes the* $\mathbf{v}_i$*-component of* $\mathbf{z}_n$ *and adds* α_i*. At* $\beta_i < 1$ *the update is partial and at* $\beta_i > 1$ *it slightly overshoots, allowing the model to learn under- or over-corrective behaviors where beneficial.*

The proposition shows that PatchNT offers a brand-new affine transformation that requires only selective modification of the $\mathbf{v}_i$-component of $\mathbf{z}$, leaving the orthogonality of all components intact. Applying such transformation to all N patch features across all k directions, we obtain k normal views, each consisting of N transformed features:

$$\widetilde{\mathbf{Z}}^{(i)} = \big[\tilde{\mathbf{z}}_1^{(i)}, \tilde{\mathbf{z}}_2^{(i)}, \ldots, \tilde{\mathbf{z}}_N^{(i)}\big]^\top \in \mathbb{R}^{N \times d}, \quad i \in \{1, 2, \ldots, k\}, \tag{7}$$

where each $\tilde{\mathbf{z}}_n^{(i)}$ is adaptively calibrated toward normality captured along the $\mathbf{v}_i$ direction. Thus, $\widetilde{\mathbf{Z}}^{(i)}$ is effectively aligned with a direction-specific view of normality.

4.4 CONNC: CONTRASTIVE NORMALITY CONSTRAINT

In the transformed feature space, we aim to promote coherence among patch features calibrated along the same direction while preserving distinctions between features calibrated along different directions. This helps each view to accurately align with the direction-specific normality and reduces redundancy across views. Contrastive learning naturally supports this objective by encouraging within-view compactness and between-view separation. Inspired by self-supervised transformation classification (He et al., 2018; Bergman & Hoshen, 2020), we therefore introduce a contrastive normality constraint that uses the k transformation views as self-supervised labels, encouraging the transformed features to form k compact and discriminative normality manifolds.

Formally, let $\tilde{\mathbf{z}}_n^{(i)} \in \mathbb{R}^d$ denote the i-th transformation view of patch feature $\mathbf{z}_n$, the center $\mathbf{c}_i$ be the average feature over all N normal patch features for the i-th transformation, *i.e.*, $\mathbf{c}_i = \frac{1}{N}\sum_{n=1}^{N} \tilde{\mathbf{z}}_n^{(i)}$, and the contrastive normality constraint can then be formulated as follows:

$$\mathcal{L}_{\text{ConNC}} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{k}\sum_{i=1}^{k}\max(0,\ \|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_i\|_2 + \gamma - \min_{j\neq i}\|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_j\|_2), \tag{8}$$

where $\gamma > 0$ is a margin hyperparameter. The above loss encourages each transformation view $\tilde{\mathbf{z}}_n^{(i)}$ to be closer to its own cluster center $\mathbf{c}_i$ than to any other cluster center $\mathbf{c}_j$ ($j \neq i$) by a margin of at least γ. This ensures that the k normality-calibrated views are not only compact within each cluster but also mutually discriminative, forming a structured partition of the normal feature space that is difficult for anomalous features to satisfy.

4.5 TRAINING AND INFERENCE

Training. Given a normal patch feature $\mathbf{z}_n$, following the SVD projection and normality transformation stated above, we obtain its k calibrated features, which are then aggregated into a single normal feature by taking their element-wise mean, *i.e.*, $\tilde{\mathbf{z}}_n = \frac{1}{k}\sum_{i=1}^{k}\tilde{\mathbf{z}}_n^{(i)}$. Fusing over these k principal directions yields a consensus feature that reinforces the normality components. These fused patch features are then concatenated and passed through a linear head to produce the module prediction $\tilde{\mathbf{Y}}_t^{\text{ncl}}$. To retain the strong modeling capacity of the original TSFM while amplifying the normality properties, we fuse the module prediction with the frozen TSFM prediction $\hat{\mathbf{Y}}_t$ as follows:

$$\tilde{\mathbf{Y}}_t = \boldsymbol{\lambda} \odot \tilde{\mathbf{Y}}_t^{\text{ncl}} + (1 - \boldsymbol{\lambda}) \odot \hat{\mathbf{Y}}_t, \tag{9}$$

where $\odot$ denotes channel-wise broadcasting multiplication, and $\boldsymbol{\lambda} \in (0,1)^D$ is a learnable gated weight that captures the channel-specific reliance on the normality constraint versus the original TSFM output. This fusion mechanism is to increase reconstruction/forecasting error for abnormal observations through the normality-constrained component $\tilde{\mathbf{Y}}_t^{\text{ncl}}$. The key insight is that for abnormal input, the normality constraint pushes them away from the pre-training support $\mathcal{S}_j$, causing NCL-enhanced TSFMs to produce larger errors. This error amplification effect is theoretically justified via Theorem 1, with the proof provided in Appendix A.2.1.

Theorem 1 (Error Amplification)**.** *Let $\mathcal{F}^*$ be a frozen TSFM pre-trained on $\mathcal{S}_j$ with Lipschitz constant $L_{\mathcal{F}}$, and let $(\mathbf{X}_a, \mathbf{Y}_a) \in \text{Supp}(\mathcal{P}_a) \subseteq \mathcal{N}_\rho(\mathcal{S}_j)$ be an anomalous pair. Under assumption with linear expansion rate $\omega(r) = \kappa r$ where $\kappa \in (0, L_{\mathcal{F}})$, for the fused prediction $\tilde{\mathbf{Y}}_a$, we have:*

$$\|\mathbf{Y}_a - \tilde{\mathbf{Y}}_a\|_2^2 > \|\mathbf{Y}_a - \hat{\mathbf{Y}}_a\|_2^2, \tag{10}$$

demonstrating that output fusion amplifies the reconstruction or forecasting errors on anomalies.

Accordingly, the normality transformation is optimized by minimizing the following training loss that combines the squared error (for either reconstruction or forecasting) loss with the contrastive normality constraint:

$$\mathcal{L} = \mathcal{L}_{\text{Pred}} + \mathcal{L}_{\text{ConNC}}, \tag{11}$$

where $\mathcal{L}_{\text{Pred}}$ is the squared error between the fused prediction $\tilde{\mathbf{Y}}_t$ and the ground-truth target $\mathbf{Y}_t$. During the training process, only the parameters of the NCL module are updated, while all parameters of the TSFM $\mathcal{F}$ remain frozen. We also conduct detailed theoretical analysis in Appendix A.2.2 to

demonstrate that, in case there is abnormal information in the training normal patch features, $\mathcal{L}_{\text{ConNC}}$ can serve as a regularization term that prevents the NCL module from overfitting to it.

Inference. During inference, the anomaly score consists of two components. One is the amplified reconstruction/forecasting squared error between $\tilde{\mathbf{Y}}_t$ and $\mathbf{Y}_t$. Another anomaly scoring component is to leverage the misalignment between each test sample's normality-calibrated views and the centers. For patch features $\mathbf{Z}_{\text{test}} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M\}$ from a test sample, this misalignment-based score for the sample is defined as the maximum over all M patches of the average negative log-probability of each view being assigned to its own center:

$$s = \max_{m \le M} -\frac{1}{k} \sum_{i=1}^{k} \log \frac{e^{-\|\tilde{\mathbf{z}}_m^{(i)} - \mathbf{c}^{(i)}\|_2}}{\sum_{j=1}^{k} e^{-\|\tilde{\mathbf{z}}_m^{(i)} - \mathbf{c}^{(j)}\|_2}}. \tag{12}$$

For a normal sample, each calibrated view $\tilde{\mathbf{z}}_m^{(i)}$ lies close to its corresponding center $\mathbf{c}^{(i)}$. For an abnormal one, the normality transformation cannot align the anomalous feature to center equally well. The maximum over all M patches is used as the sample score, as the top anomalous patch conveys the most information about abnormality.

5 Experiments

5.1 Experimental Setup

Datasets. To evaluate the effectiveness of NCL, we conduct comprehensive experiments on ten TSAD datasets, including five univariate datasets—**IOPS** (Ren et al., 2019), **UCR** (Wu & Keogh, 2021), **NAB** (Ahmad et al., 2017), **MSL** (Benecki et al., 2021), and **SMAP** (Benecki et al., 2021)—and five multivariate datasets—**PSM** (Abdulaal et al., 2021), **SMD** (Su et al., 2019), **SWAT** (Mathur & Tippenhauer, 2016), **PUMP** (Feng & Tian, 2021), and **NYC** (Cui et al., 2016). Details about the datasets are included in Appendix D.1.

Baselines. We choose both forecasting-based and reconstruction-based TSFMs for anomaly detection as backbones. Forecasting-based TSFMs include Aurora (Wu et al., 2026), ROSE (Wang et al., 2025), Chronos (Ansari et al., 2024), and TTM (Ekambaram et al., 2024). For reconstruction-based TSFMs, MOMENT (Goswami et al., 2024) and UniTS (Gao et al., 2024) are selected. Additionally, we also include TSPulse (Ekambaram et al., 2026) and SEMPO (He et al., 2025), which employ both forecasting and reconstruction. Other popular TSFMs, such as Time-MoE (Shi et al., 2025) and TimesFM (Das et al., 2024), are also compared in Section 5.4 and Appendix I. NCL-augmented TSFMs are also compared with deep TSAD methods, simple one-liner baselines, and TSAD-specific foundation models to further demonstrate the advances made by our NCL, with the results presented in Appendix H and I. The detailed descriptions of these baselines can be found in Appendix D.2.

Implementation Details. NCL is trained for five epochs with only five normal samples randomly selected from the training set (see Appendix O for other few-shot sizes). All reported results are averaged over three independent runs. More implementation details are given in Appendix D.3.

Evaluation Metrics. Recent works have highlighted that point-adjusted metrics may exhibit severe overestimation issues (Liu & Paparrizos, 2024; Qiu et al., 2025), whereas VUS-PR (Paparrizos et al., 2022a) emerges as a more robust and fair evaluation measure. Therefore, we adopt VUS-PR as our primary metric. Several other PR and ROC metrics are reported in Appendix Q.

5.2 Main Results

Enabling Anomaly Detection of Diverse TSFMs. Table 1 presents the results of eight different TSFMs with/without using our proposed NCL across ten TSAD datasets. The full results (including several PR and ROC metric results with the standard deviations) are provided in Appendix Q. It is clear that NCL consistently and significantly improves the anomaly detection performance of diverse TSFMs, with an average improvement ratio of 41.49% across all backbone models. This dramatic enhancement demonstrates NCL's ability to address the fundamental limitation of TSFMs in distinguishing normal and abnormal patterns.

Comparison with Other TSFM Adaptation Methods. To demonstrate the superiority of NCL over alternative adaptation strategies, we compare it against two popular TSFM adaptation methods,

Table 1: VUS-PR results of vanilla and NCL-augmented TSFMs that use forecasting/reconstruction errors to detect anomalies on ten TSAD datasets. $\triangle_{\text{imp}}$ indicates improvement ratios gained by NCL.

Dataset	Aurora			ROSE			Chronos			TMM		
	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$
IOPS	0.1333	0.2528	+89.65	0.1307	0.1936	+48.13	0.2387	0.3614	+51.40	0.1877	0.3338	+77.84
UCR	0.0216	0.0247	+14.35	0.0242	0.0329	+35.95	0.0566	0.0691	+22.08	0.0400	0.0410	+2.50
NAB	0.3004	0.3401	+13.22	0.3365	0.3754	+11.56	0.2766	0.3756	+35.79	0.2894	0.3368	+16.38
MSL	0.2982	0.3681	+23.44	0.3604	0.4488	+24.53	0.3943	0.4361	+10.60	0.3653	0.4849	+32.74
SMAP	0.1889	0.2823	+49.44	0.2417	0.3484	+44.15	0.3106	0.3741	+20.44	0.2736	0.3686	+34.72
PSM	0.3592	0.4743	+32.04	0.3672	0.4107	+11.85	0.3474	0.4926	+41.80	0.3611	0.5039	+39.55
SMD	0.1070	0.1483	+38.60	0.1025	0.1331	+29.85	0.0980	0.1462	+49.18	0.1109	0.1591	+43.46
SWAT	0.1265	0.6484	+412.57	0.1638	0.5437	+231.93	0.5979	0.6281	+5.05	0.1185	0.6099	+414.68
PUMP	0.2027	0.2247	+10.85	0.2360	0.2392	+1.36	0.2097	0.2347	+11.92	0.2082	0.2297	+10.33
NYC	0.0341	0.0408	+19.65	0.0969	0.1431	+47.68	0.0482	0.0604	+25.31	0.0619	0.0806	+30.21
Mean	0.1772	0.2805	+58.28	0.2060	0.2869	+39.27	0.2578	0.3178	+23.29	0.2017	0.3148	+56.12
Dataset	**MOMENT**			**UniTS**			**TSPulse**			**SEMPO**		
	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$	w/o NCL	w/ NCL	$\triangle_{\text{imp}}(\%)$
IOPS	0.0657	0.1053	+60.27	0.1530	0.1861	+21.63	0.1919	0.3354	+74.78	0.1335	0.1822	+36.48
UCR	0.0796	0.0929	+16.71	0.0808	0.0832	+2.97	0.0274	0.0320	+16.79	0.0282	0.0363	+28.72
NAB	0.5210	0.5565	+6.81	0.4533	0.4962	+9.46	0.3201	0.3750	+17.15	0.3753	0.4084	+8.82
MSL	0.4726	0.5703	+20.67	0.5033	0.5657	+12.40	0.3060	0.4694	+53.40	0.4223	0.5081	+20.32
SMAP	0.4288	0.4716	+9.98	0.5263	0.5345	+1.56	0.2251	0.3687	+63.79	0.2830	0.4227	+49.36
PSM	0.3682	0.5279	+43.37	0.4203	0.4735	+12.66	0.3618	0.4840	+33.78	0.3624	0.4084	+12.69
SMD	0.0942	0.1312	+39.28	0.1296	0.1343	+3.63	0.1041	0.1246	+19.69	0.1080	0.1249	+15.65
SWAT	0.1895	0.5281	+178.68	0.1215	0.5346	+340.00	0.1190	0.6168	+418.32	0.1702	0.5287	+210.63
PUMP	0.2346	0.2764	+17.82	0.2370	0.2534	+6.92	0.1963	0.2324	+18.39	0.2389	0.2512	+5.15
NYC	0.0318	0.0327	+2.83	0.1495	0.1850	+23.75	0.0624	0.0670	+7.37	0.0862	0.1332	+54.52
Mean	0.2486	0.3293	+32.46	0.2775	0.3447	+24.22	0.1914	0.3105	+62.23	0.2208	0.3004	+36.06

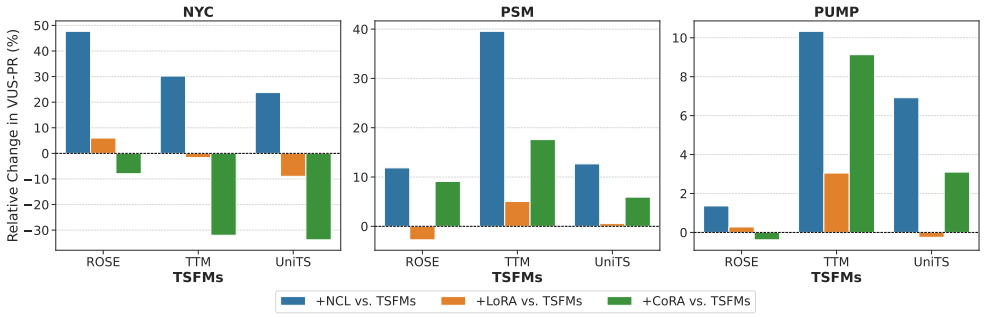


Figure 3: Performance changes relative to vanilla TSFMs for three different TSFMs.

including LoRA (Hu et al., 2022) and CoRA (Cheng et al., 2026). As shown in Figure 3, NCL consistently outperforms these methods across three representative TSFMs. LoRA can further utilize the limited normal samples to fine-tune TSFMs, while CoRA can enhance the modeling ability across channels; however, neither of them has an explicit normality constraint design for anomaly detection. In contrast, NCL constructs a sample-efficient normality space and constrains the features to align with its normality manifolds, thereby making abnormal deviations stand out more clearly.

5.3 Ablation Studies

To validate the contribution of each component in NCL, we conduct comprehensive ablation studies by systematically removing or replacing its individual components. Table 2 presents the results on IOPS and PSM datasets with three representative TSFMs (ROSE, TMM, UniTS).

Normality Grounded by NormSpace. As a NCL variant that uses only NormSpace, relying on the top-k singular vectors to perform reconstruction-based anomaly detection directly, NCL_{N} gains consistent improvement compared to all three vanilla TSFMs. This indicates that, although only a few patch features are used, NormSpace can achieve a strong sample-efficient normality grounding that effectively narrows down the broad pattern space of TSFMs onto the normality space. This also confirms Theorems 2 and 3 in Appendix B. Its significance can also be observed when **(i)** comparing NCL_{P} to NCL_{NP} or **(ii)** comparing NCL_{PC} to the full NCL method NCL_{NPC}.

Learnable Transformation to Calibrate Features via PatchNT. Adding the learnable transformation in PatchNT on top of NCL_N, NCL_{NP} further effectively calibrates the features in the normality space across all three TSFMs on both datasets. Even without using NormSpace, the adaptive transformation in PatchNT works well for the calibration as highlighted by the difference between NCL_C (which uses random affine transformation to replace PatchNT) and NCL_{PC}. Besides, the features transformed solely by NCL_P are also better than the vanilla TSFM features. Collectively, these results systematically justify the effectiveness of our learnable transformation design.

Table 2: Ablation study results of NCL and its variants.

Dataset				IOPS			PSM		
Variant	NormSpace	PatchNT	ConNC	ROSE	TMM	UniTS	ROSE	TMM	UniTS
Vanilla	✘	✘	✘	0.1307	0.1877	0.1530	0.3672	0.3611	0.4203
NCL_N	✔	✘	✘	0.1728	0.2821	0.1545	0.3829	0.4549	0.4479
NCL_P	✘	✔	✘	0.1429	0.2356	0.1589	0.3761	0.4409	0.4375
NCL_C	✘	✘	✔	0.1796	0.2874	0.1584	0.3789	0.4413	0.4358
NCL_{NP}	✔	✔	✘	0.1811	0.3092	0.1623	0.3912	0.4655	0.4507
NCL_{PC}	✘	✔	✔	0.1843	0.3059	0.1668	0.3835	0.4660	0.4512
NCL_{NPC}	✔	✔	✔	**0.1936**	**0.3338**	**0.1861**	**0.4107**	**0.5039**	**0.4735**

Sharpening the Feature Normality Further via ConNC. Adding ConNC on top of NCL_{NP}, our full method NCL_{NPC} gains consistently large improvements across all cases, showcasing the importance of the contrastive constraint for enhancing the discriminability of the calibrated features. This constraint can also work well solely (*i.e.*, NCL_C) compared to the vanilla TSFMs. Overall, these well justify the role of ConNC in guiding PatchNT to further sharpen the normality within the subspace.

5.4 Efficiency Analysis

NCL enjoys not only substantial detection improvements but also parameter and training efficiency. Figure 4 compares its training costs with LoRA, CoRA, and full fine-tuning TSFMs, with the results averaged over the five multivariate datasets. Our key observations are as follows.

Parameter Efficiency. NCL requires an average of only **14.2K** trainable parameters across different TSFMs, which is significantly fewer than both the parameter-efficient fine-tuning methods like LoRA (1.15M) and CoRA (8.05M) and the full fine-tuning (from 371.4K to 205.29M).

Training Speed. NCL exhibits remarkably fast training time, completing its full training within around **2 seconds**. The efficiency stems from two main factors: (i) NormSpace requires only few-shot samples to construct an informative normality space and (ii) only lightweight projection heads and fusion weights are learned while the TSFM remains frozen.

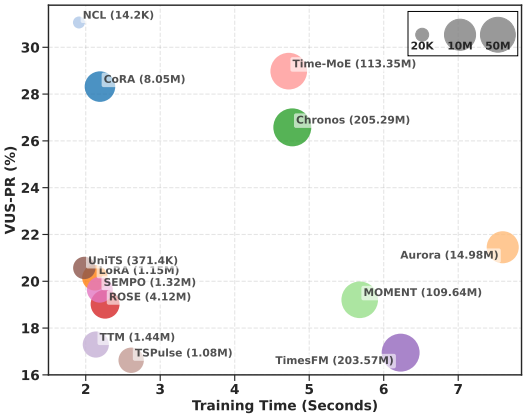


Figure 4: Effectiveness vs. efficiency trade-off.

Overall, represented by a small bubble in the upper-left of the plot (denoting a lower number of training parameters), NCL achieves substantially better detection effectiveness (higher on the y-axis) while also having a fast training time (further left on the x-axis). In contrast, most competitors concentrate in the lower-right region, requiring longer training times yet inferior VUS-PR. More detailed efficiency comparisons are in Appendix G.

6 Conclusion

We propose NCL, a lightweight plug-and-play framework that empowers frozen TSFMs to achieve accurate anomaly detection. NCL is implemented with NormSpace, PatchNT, and ConNC. NormSpace constructs a normality subspace, while PatchNT adaptively steers patch features toward normality within this subspace, guided by ConNC to encourage the formation of compact yet discriminative normality manifolds. We justify the effectiveness of NCL in constraining TSFMs into the normality space through both theoretical and empirical evidence. These results imply that further specialization/adaptation of pre-trained TSFMs is required for the downstream anomaly detection task, and NCL provides a generalizable framework and a new perspective on tackling this issue.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing of the paper. We have not used generative AI tools for formulating mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, or implementing methods; and the remaining required-disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. Specifically, all AI-assisted text was checked and revised by the authors to ensure fidelity to the intended meaning; the AI tools were used solely to improve grammar, clarity, and fluency, and did not generate new scientific content. No AI-generated text was incorporated without author verification. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work provides a methodological contribution and relies exclusively on publicly available, non-proprietary datasets used under their original licenses. It involves neither human subjects nor sensitive data. We believe our results advance the understanding of specialized foundation models and time series anomaly detection while raising no additional ethical concerns.

REPRODUCIBILITY STATEMENT

We provide the hyperparameter configurations and implementation details for our experiments in Appendix D. The core algorithmic component of our method is described in Appendix E. The theoretical results presented in this paper, including their proofs and assumptions, are detailed in Appendix A, B, and C. The source code has been submitted as supplemental material and will be made publicly available upon acceptance of this paper.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2485–2494, 2021.

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

Archana Anandakrishnan, Senthil Kumar, Alexander Statnikov, Tanveer Faruquie, and Di Xu. Anomaly detection in finance: editors' introduction. In *KDD 2017 Workshop on Anomaly Detection in Finance*, pp. 1–7. PMLR, 2018.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024, 2024.

Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. Detecting anomalies in spacecraft telemetry using evolutionary thresholding and lstms. In *Proceedings of the genetic and evolutionary computation conference companion*, pp. 143–144, 2021.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations (ICLR)*, 2020.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Guo Guo. Cora: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *International Conference on Learning Representations*, volume 2026, pp. 106657–106680, 2026.

Dongchan Cho, Jiho Han, Keumyeong Kang, Minsang Kim, Honggyu Ryu, and Namsoon Jung. Structured temporal causality for interpretable multivariate time series anomaly detection. *Advances in Neural Information Processing Systems*, 38:59257–59294, 2025.

Yuwei Cui, Chetan Surpur, Subutai Ahmad, and Jeff Hawkins. A comparative study of htm and other neural network models for online sequence learning with streaming data. In *IJCNN*, pp. 1530–1538, 2016.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International conference on learning representations*, volume 235, pp. 10148–10167, 2024.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H Nguyen, Wesley M Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.

Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley Gifford, and Jayant Kalagnanam. Tspulse: Tiny pre-trained models with disentangled representations for rapid time-series analysis. In *International Conference on Learning Representations*, volume 2026, pp. 62090–62133, 2026.

Cheng Feng and Pengwei Tian. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2858–2867, 2021.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. Extended isolation forest. *IEEE transactions on knowledge and data engineering*, 33(4):1479–1489, 2019.

Hui He, Kun Yi, Yuanchi Ma, Qi Zhang, Zhendong Niu, and Guansong Pang. Sempo: Lightweight foundation models for time series forecasting. *Advances in Neural Information Processing Systems*, 38:161846–161879, 2025.

Hui He, Hezhe Qiao, Yutong Chen, Kun Yi, and Guansong Pang. Timeradar: A domain-rotatable foundation model for time series anomaly detection. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1602–1613, 2026.

Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 1945–1954. Ieee, 2018.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Shiyan Hu, Tengxue Zhang, Jianxin Jin, Xiangfei Qiu, Bin Yang, and Chenjuan Guo. Teamwork: Multivariate time series anomaly detection via asymmetric role-aware channel modeling. In *Forty-third International Conference on Machine Learning*, 2026.

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

Beibu Li, Qichao Shentu, Yang Shu, Hui Zhang, Ming Li, Ning Jin, Bin Yang, and Chenjuan Guo. CrossAD: Time series anomaly detection with cross-scale associations and cross-window modeling. In *NeurIPS*, 2025.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model merging in the essential subspace. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31145–31154, 2026.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Qinghua Liu and John Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 37:108231–108261, 2024.

Zehao Liu, Mengzhou Gao, and Pengfei Jiao. Gcad: Anomaly detection in multivariate time series from the perspective of granger causality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19041–19049, 2025.

Donghao Luo and Xue Wang. Anomalytcn: Efficient contrastive-based time series anomaly detection with pure convolution structure. *Neural Networks*, pp. 108879, 2026.

Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

Xudong Mou, Rui Wang, Bo Li, Tianyu Wo, Jie Sun, Hui Wang, and Xudong Liu. Roca: Robust contrastive one-class time series anomaly detection with contaminated data. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S Tsay, Aaron Elmore, and Michael J Franklin. Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(11):2774–2787, 2022a.

John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. Tsb-uad: An end-to-end benchmark suite for univariate time-series anomaly detection. *Proc. VLDB Endow.*, 15(8):1697–1711, 2022b.

Jinju Park and Seokho Kang. Paano: Patch-based representation learning for time-series anomaly detection. In *Proceedings of International Conference on Learning Representations*, 2026.

Miseon Park and Kijung Yoon. A comparative study of adaptation strategies for time series foundation models in anomaly detection. In *International Conference on Pattern Recognition*, pp. 636–650. Springer, 2026.

Joao Pereira and Margarida Silveira. Learning representations from healthcare time series data for unsupervised anomaly detection. In *2019 IEEE international conference on big data and smart computing (BigComp)*, pp. 1–7. IEEE, 2019.

Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. TAB: Unified benchmarking of time series anomaly detection methods. In *Proc. VLDB Endow.*, 2025.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in neural information processing systems*, 36:79320–79362, 2023.

Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. Time-series anomaly detection service at microsoft. In *SIGKDD*, pp. 3009–3017, 2019.

Qichao Shentu, Beibu Li, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders. In *International Conference on Learning Representations*, volume 2025, pp. 81358–81381, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International conference on learning representations*, volume 2025, pp. 34635–34667, 2025.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, volume 2025, pp. 4501–4519, 2025.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and trends® in machine learning*, 8(1-2):1–230, 2015.

Yihang Wang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series forecasting model with unified representation and adaptive transfer. In *International Conference on Machine Learning*, 2025.

Yue Wang, Michael Perry, Dane Whitlock, and John W Sutherland. Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. *Journal of Manufacturing Systems*, 62:823–834, 2022.

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

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

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Xingjian Wu, Jianxin Jin, Wanghui Qiu, Peng Chen, Yang Shu, Bin Yang, and Guo Guo. Aurora: Towards universal generative multimodal time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 103239–103277, 2026.

Hongzuo Xu, Yijie Wang, Songlei Jian, Qing Liao, Yongjun Wang, and Guansong Pang. Calibrated one-class classification for unsupervised time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

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

Hanqi Yan, Lin Gui, Wenjie Li, and Yulan He. Addressing token uniformity in transformers via singular value transformation. In *Uncertainty in artificial intelligence*, pp. 2181–2191. PMLR, 2022.

Annan Yu, Danielle Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W Mahoney, and Bernie Wang. Understanding transformers for time series: Rank structure, flow-of-ranks, and compressibility. In *International Conference on Learning Representations*, volume 2026, pp. 42449–42489, 2026.

Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1):1–42, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Fan Zhang, Sinchee Chin, Jing-Hao Xue, and Wenming Yang. Icdiffad: Implicit conditioning diffusion model for time series anomaly detection. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Yifan Zhang, Yifeng Liu, Mengdi Wang, and Quanquan Gu. Deep delta learning. *arXiv preprint arXiv:2601.00417*, 2026b.

Xiaohui Zhou, Yijie Wang, Hongzuo Xu, and Yizhou Li. Modeling heterogeneous normality in time series anomaly detection. *Information Processing & Management*, 63(5):104644, 2026a.

Xiaohui Zhou, Yijie Wang, Hongzuo Xu, Weixuan Liang, Xiaoli Li, and Guansong Pang. Impact: Influence modeling for open-set time series anomaly detection. In *International Conference on Machine Learning*, 2026b.

Xiaokun Zhu, Louis Carpentier, and Mathias Verbeke. When foundation models are one-liners: Limitations and future directions for time series anomaly detection. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhihao Zhuang, Yingying Zhang, Kai Zhao, Chenjuan Guo, Bin Yang, Qingsong Wen, and Lunting Fan. Noise matters: Cross contrastive learning for flink anomaly detection. *Proceedings of the VLDB Endowment*, 18(4):1159–1168, 2024.

A THEORETICAL ANALYSIS OF APPLYING TSFMS TO ANOMALY DETECTION

In Section 1, we emphasize that both reconstruction-based and forecasting-based TSFMs can effectively reconstruct or forecast most of the abnormal observations, meaning that anomalies do not necessarily incur larger reconstruction or forecasting errors. We attribute this issue to two aspects: **(i)** TSFMs are typically pre-trained on large-scale time-series datasets encompassing diverse patterns and domains. Consequently, observations that constitute anomalies in a specific application domain may represent normal patterns in other domains within the pre-training corpus, enabling the model to reconstruct or forecast them with high fidelity; and **(ii)** the large parameter capacity of TSFMs allows them to further capture and internalize a wide range of general temporal patterns.

These two factors jointly lead to the extremely broad pattern modeling ability of TSFMs, making anomalies not inherently difficult to reconstruct or forecast. This undermines the fundamental assumption that anomalies should yield higher reconstruction or forecasting errors, thereby limiting the effectiveness of applying TSFMs to anomaly detection. In order to further understand the influence of these two aspects on the broad pattern modeling ability of TSFMs during the pre-training process, we conducted a detailed theoretical analysis in Appendix A.1.

A.1 PRE-TRAINING AND ANOMALY DETECTION ON THE TSFMS

Let $\{\mathcal{P}_i\}_{i=1}^s$ represent the collection of various time series distributions in a large-scale corpus. Then we know the empirical distribution of the entire pre-training data is $\mathcal{P}_{pre} = \sum_{i=1}^s \eta_i \mathcal{P}_i$, where $\sum_{i=1}^s \eta_i = 1$. Pre-training a reconstruction-based or forecasting-based TSFM $\mathcal{F}$ aims to minimize the following objective:

$$\mathcal{L}_{\text{Pred}}(\mathcal{F}(\cdot)) = \sum_{i=1}^{s} \eta_i \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{P}_i} \left[\|\mathbf{Y} - \mathcal{F}(\mathbf{X})\|_2^2 \right]. \tag{13}$$

By the universal approximation theorem (Hornik, 1991), for any continuous function f and any $\epsilon > 0$, there exists a neural network $\mathcal{F}$ with sufficient capacity such that

$$\sup_{\mathbf{X}\in\mathcal{K}} \|\mathcal{F}(\mathbf{X}) - f(\mathbf{X})\|_2^2 \leq \epsilon, \tag{14}$$

where $\mathcal{K}$ is a compact subset. For each distribution $\mathcal{P}_i$ in the pre-training corpus, let $f_i^*(\cdot)$ denote the optimal predictor that minimizes the expected squared error:

$$f_i^*(\mathbf{X}) = \mathbb{E}_{\mathbf{Y}|\mathbf{X}\sim\mathcal{P}_i}[\mathbf{Y}|\mathbf{X}]. \tag{15}$$

Given sufficient model capacity and training data, the pre-trained TSFM $\mathcal{F}$ approximates a weighted combination of these optimal predictors. Specifically, minimizing Equation (13) yields:

$$\mathcal{F}^*(\mathbf{X}) \approx \sum_{i=1}^{s} \eta_i f_i^*(\mathbf{X}) = \mathbb{E}_{\mathbf{Y}|\mathbf{X}\sim\mathcal{P}_{pre}}[\mathbf{Y}|\mathbf{X}], \tag{16}$$

where the approximation becomes exact as model capacity increases.

Then we consider a downstream anomaly detection task on a target distribution $\mathcal{P}_t$, where anomalies correspond to observations drawn from an anomalous distribution $\mathcal{P}_a$. The fundamental challenge arises when the support of $\mathcal{P}_a$ overlaps with or lies close to the support of pre-training distributions. Let $\mathcal{S}_j = \text{Supp}(\mathcal{P}_j)$ denote the support of the j-th pre-training distribution in the input-output space. For $\rho \geq 0$, we define its ρ-neighborhood as:

$$\mathcal{N}_\rho(\mathcal{S}_j) = \left\{ (\mathbf{X},\mathbf{Y}) : \inf_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}})\in\mathcal{S}_j} \left\| (\mathbf{X},\mathbf{Y}) - (\widetilde{\mathbf{X}},\widetilde{\mathbf{Y}}) \right\|_2 \leq \rho \right\}. \tag{17}$$

Suppose the support of the anomalous distribution is contained within this neighborhood, *i.e.*, $\text{Supp}(\mathcal{P}_a) \subseteq \mathcal{N}_\rho(\mathcal{S}_j)$, for some $j \in \{1,\ldots,s\}$ and sufficiently small $\rho > 0$. This condition means that, although these observations are abnormal in the downstream domain, their temporal patterns are close to patterns that appeared in the pre-training corpus. Assume the pre-trained model achieves uniformly small prediction error on $\mathcal{S}_j$:

$$\sup_{(\mathbf{X},\mathbf{Y})\in\mathcal{S}_j} \|\mathbf{Y} - \mathcal{F}^*(\mathbf{X})\|_2 \leq \sup_{(\mathbf{X},\mathbf{Y})\in\mathcal{S}_j} \|\mathbf{Y} - f^*(\mathbf{X})\|_2 + \|f^*(\mathbf{X}) - \mathcal{F}^*(\mathbf{X})\|_2 \leq \delta + \epsilon_j, \tag{18}$$

where δ is the irreducible uncertainty error, and ϵ_j is the approximation error. Furthermore, assume $\mathcal{F}^*$ is $L_{\mathcal{F}}$-Lipschitz continuous on $\mathcal{N}_\rho(\mathcal{S}_j)$:

$$\|\mathcal{F}^*(\mathbf{X}_1) - \mathcal{F}^*(\mathbf{X}_2)\|_2 \leq L_{\mathcal{F}}\|\mathbf{X}_1 - \mathbf{X}_2\|_2. \tag{19}$$

For any anomalous pair $(\mathbf{X}_a, \mathbf{Y}_a) \in \mathrm{Supp}(\mathcal{P}_a)$, since $\mathrm{Supp}(\mathcal{P}_a) \subseteq \mathcal{N}_\rho(\mathcal{S}_j)$, there exists $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \in \mathcal{S}_j$ such that

$$\|\mathbf{X}_a - \widetilde{\mathbf{X}}\|_2 \leq \rho, \quad \|\mathbf{Y}_a - \widetilde{\mathbf{Y}}\|_2 \leq \rho. \tag{20}$$

By the triangle inequality:

$$\begin{aligned}\|\mathbf{Y}_a - \mathcal{F}^*(\mathbf{X}_a)\|_2 &\leq \|\mathbf{Y}_a - \widetilde{\mathbf{Y}}\|_2 + \|\widetilde{\mathbf{Y}} - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2 + \|\mathcal{F}^*(\widetilde{\mathbf{X}}) - \mathcal{F}^*(\mathbf{X}_a)\|_2 \\ &\leq \rho + \delta + \epsilon_j + L_{\mathcal{F}}\rho \\ &= \epsilon_j + \delta + (1 + L_{\mathcal{F}})\rho,\end{aligned} \tag{21}$$

where the second inequality follows from: (i) $\|\mathbf{Y}_a - \widetilde{\mathbf{Y}}\|_2 \leq \rho$ by Inequation (20); (ii) $\|\widetilde{\mathbf{Y}} - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2 \leq \delta + \epsilon_j$ since $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \in \mathcal{S}_j$ and the uniform error bound in Inequation (18) holds for all points in $\mathcal{S}_j$; and (iii) $\|\mathcal{F}^*(\widetilde{\mathbf{X}}) - \mathcal{F}^*(\mathbf{X}_a)\|_2 \leq L_{\mathcal{F}}\|\widetilde{\mathbf{X}} - \mathbf{X}_a\|_2 \leq L_{\mathcal{F}}\rho$ by the Lipschitz continuity in Inequation (19).

Consequently, the squared error satisfies:

$$\|\mathbf{Y}_a - \mathcal{F}^*(\mathbf{X}_a)\|_2^2 \leq \left[\epsilon_j + \delta + (1 + L_{\mathcal{F}})\rho\right]^2. \tag{22}$$

Taking expectations over $\mathcal{P}_a$:

$$\mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{P}_a}\left[\|\mathbf{Y} - \mathcal{F}^*(\mathbf{X})\|_2^2\right] \leq \left[\epsilon_j + \delta + (1 + L_{\mathcal{F}})\rho\right]^2. \tag{23}$$

This reveals the over-generalization phenomenon. In non-extremely unstable scenarios, δ is usually very small. When both ϵ_j and ρ are small, anomalies that fall near the support of pre-training data can also be reconstructed or forecasted with high accuracy, making reconstruction or forecasting error an unreliable anomaly score.

A.2 Theoretical Justification of Normality Constraint Learning

Having established that TSFMs suffer from over-generalization in Appendix A.1, we now demonstrate how the proposed normality constraint learning mitigates this issue through two complementary mechanisms: **(i)** error amplification on anomalies via output fusion, and **(ii)** regularization against anomaly overfitting via contrastive normality constraint.

A.2.1 Error Amplification via Output Fusion (Proof of Theorem 1)

Proof. Recall from Equation (9) that the final prediction is formed by:

$$\tilde{\mathbf{Y}}_t = \boldsymbol{\lambda} \odot \tilde{\mathbf{Y}}_t^{\mathrm{ncl}} + (1 - \boldsymbol{\lambda}) \odot \hat{\mathbf{Y}}_t, \tag{24}$$

where $\tilde{\mathbf{Y}}_t^{\mathrm{ncl}}$ is the normality-constrained prediction and $\hat{\mathbf{Y}}_t = \mathcal{F}^*(\mathbf{X}_t)$ is the frozen TSFM prediction. For notational simplicity, we denote the channel-wise fusion weight as $\lambda \in (0, 1)$ for a single-channel analysis.

For a normal pair $(\mathbf{X}_n, \mathbf{Y}_n) \in \mathrm{Supp}(\mathcal{P}_t)$, the normality constraint learning module is trained to minimize $\mathcal{L}_{\mathrm{Pred}}$ in Equation (11). Since the transformation $\tilde{\mathbf{z}}_n^{(i)} = \mathbf{W}_i\mathbf{z}_n + \mathbf{b}_i$ projects patch features onto the normal manifold spanned by $\{\mathbf{v}_i\}_{i=1}^k$, the module prediction $\tilde{\mathbf{Y}}_n^{\mathrm{ncl}}$ approximates the normal conditional expectation:

$$\tilde{\mathbf{Y}}_n^{\mathrm{ncl}} \approx \mathbb{E}_{\mathbf{Y}|\mathbf{X}_n\sim\mathcal{P}_t}[\mathbf{Y}|\mathbf{X}_n] = f_t^*(\mathbf{X}_n). \tag{25}$$

Assuming both $\hat{\mathbf{Y}}_n$ and $\tilde{\mathbf{Y}}_n^{\mathrm{ncl}}$ achieve small errors on normal data:

$$\|\mathbf{Y}_n - \hat{\mathbf{Y}}_n\|_2 \leq \epsilon_{\mathcal{F}}, \quad \|\mathbf{Y}_n - \tilde{\mathbf{Y}}_n^{\mathrm{ncl}}\|_2 \leq \epsilon_{\mathrm{ncl}}, \tag{26}$$

the fused prediction error satisfies:

$$\begin{aligned}\|\mathbf{Y}_n - \tilde{\mathbf{Y}}_n\|_2 &= \|\mathbf{Y}_n - \lambda\tilde{\mathbf{Y}}_n^{\text{ncl}} - (1-\lambda)\hat{\mathbf{Y}}_n\|_2 \\ &\leq \lambda\|\mathbf{Y}_n - \tilde{\mathbf{Y}}_n^{\text{ncl}}\|_2 + (1-\lambda)\|\mathbf{Y}_n - \hat{\mathbf{Y}}_n\|_2 \\ &\leq \lambda\epsilon_{\text{ncl}} + (1-\lambda)\epsilon_{\mathcal{F}}. \end{aligned} \tag{27}$$

Thus, the fused prediction maintains low error on normal data.

For an anomalous pair $(\mathbf{X}_a, \mathbf{Y}_a) \in \text{Supp}(\mathcal{P}_a)$, the frozen TSFM prediction $\hat{\mathbf{Y}}_a$ achieves a small error due to over-generalization (Inequation (21)):

$$\|\mathbf{Y}_a - \hat{\mathbf{Y}}_a\|_2 \leq \epsilon_j + \delta + (1 + L_{\mathcal{F}})\rho. \tag{28}$$

Recall that $\text{Supp}(\mathcal{P}_a) \subseteq \mathcal{N}_\rho(\mathcal{S}_j)$, meaning abnormal features lie within a ρ-neighborhood of the j-th pre-training distribution support. The normality constraint learning module transforms abnormal patch features $\mathbf{z}_a$ via:

$$\tilde{\mathbf{z}}_a^{(i)} = (\mathbf{I} - \beta_i \mathbf{v}_i \mathbf{v}_i^\top)\mathbf{z}_a + \alpha_i\beta_i\mathbf{v}_i, \tag{29}$$

where $\{\mathbf{v}_i\}_{i=1}^k$ are the principal directions of the normal distribution from $\mathcal{P}_t$. Since $\mathcal{P}_t \neq \mathcal{P}_j$ (the target normal distribution differs from the pre-training distribution), these transformations calibrate $\mathbf{z}_a$ toward the target normality manifolds rather than the pre-training support $\mathcal{S}_j$. Let $\tilde{\mathbf{X}}_a$ denote the λ-fusion of the original input and the input reconstructed from the transformed patch features $\{\tilde{\mathbf{z}}_a^{(i)}\}_{i=1}^k$, the transformation then shifts the abnormal input away from $\mathcal{N}_\rho(\mathcal{S}_j)$ toward the support of $\mathcal{P}_t$. Specifically, for any $(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \in \mathcal{S}_j$ that was within ρ distance of $(\mathbf{X}_a, \mathbf{Y}_a)$, the transformed input satisfies:

$$\|\tilde{\mathbf{X}}_a - \widetilde{\mathbf{X}}\|_2 \geq \|\mathbf{X}_a - \widetilde{\mathbf{X}}\|_2 + \tau, \tag{30}$$

where $\tau \gg \rho$ quantifies the displacement induced by the normality transformation. Consequently, the transformed abnormal input $\tilde{\mathbf{X}}_a$ now lies in a larger neighborhood $\mathcal{N}_{\rho+\tau}(\mathcal{S}_j)$ relative to its original position. Since the pre-trained model $\mathcal{F}^*$ was trained on $\mathcal{S}_j$ and achieves small error only within $\mathcal{N}_\rho(\mathcal{S}_j)$, the prediction error on the transformed input increases. Thus, we have Assumption 1.

Assumption 1. *There exists a strictly increasing $\omega : [0,\infty) \to [0,\infty)$ with $\omega(0) = 0$ such that for all $\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{N}_{\rho+\tau}(\mathcal{S}_j) \setminus \mathcal{N}_\rho(\mathcal{S}_j)$,*

$$\omega\big(\|\mathbf{X}_1 - \mathbf{X}_2\|_2\big) \;\leq\; \|\mathcal{F}^*(\mathbf{X}_1) - \mathcal{F}^*(\mathbf{X}_2)\|_2 \;\leq\; L_{\mathcal{F}}\|\mathbf{X}_1 - \mathbf{X}_2\|_2. \tag{31}$$

Under Assumption 1, applying the reverse triangle inequality with $\mathcal{F}^*(\widetilde{\mathbf{X}})$ where $\widetilde{\mathbf{X}}$ has ρ distance from $\mathbf{X}_a$, we have:

$$\begin{aligned}\|\mathbf{Y}_a - \tilde{\mathbf{Y}}_a\|_2 &= \|\mathbf{Y}_a - \mathcal{F}^*(\tilde{\mathbf{X}}_a)\|_2 \\ &\geq \|(\mathbf{Y}_a - \mathcal{F}^*(\widetilde{\mathbf{X}})) + (\mathcal{F}^*(\widetilde{\mathbf{X}}) - \mathcal{F}^*(\tilde{\mathbf{X}}_a))\|_2 \\ &\geq \|\mathcal{F}^*(\tilde{\mathbf{X}}_a) - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2 - \|\mathbf{Y}_a - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2. \end{aligned} \tag{32}$$

Since $\widetilde{\mathbf{X}} \in \mathcal{S}_j$ and $\mathcal{F}^*$ perform well in $\mathcal{N}_\rho(\mathcal{S}_j)$, we can simplify the above as:

$$\begin{aligned}\|\mathbf{Y}_a - \tilde{\mathbf{Y}}_a^{\text{ncl}}\|_2 &\geq \|\mathcal{F}^*(\tilde{\mathbf{X}}_a) - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2 - \|\mathbf{Y}_a - \mathcal{F}^*(\mathbf{X}_a)\|_2 \\ &\geq \omega\big(\|\tilde{\mathbf{X}}_a - \widetilde{\mathbf{X}}\|_2\big) - B_j \\ &\geq \omega\big(\rho + \tau\big) - B_j, \end{aligned} \tag{33}$$

where $B_j := \epsilon_j + \delta + (1 + L_{\mathcal{F}})\rho$ according to Inequation (28). By simultaneously discarding the uncertainty and approximation parts in both $\omega\big(\rho+\tau\big)$ and B_j, *i.e.*, ϵ_j and δ, we can approximately regard $\omega\big(\rho+\tau\big)$ as a linear function of $\rho + \tau$, *i.e.*, $\omega\big(\rho+\tau\big) = \kappa \cdot \big(\rho+\tau\big)$, where $0 < \kappa < L_{\mathcal{F}}$. Then we have:

$$\begin{aligned}\|\mathbf{Y}_a - \tilde{\mathbf{Y}}_a^{\text{ncl}}\|_2 - \|\mathbf{Y}_a - \hat{\mathbf{Y}}_a\|_2 &\geq \|\mathcal{F}^*(\tilde{\mathbf{X}}_a) - \mathcal{F}^*(\widetilde{\mathbf{X}})\|_2 - 2\|\mathbf{Y}_a - \mathcal{F}^*(\mathbf{X}_a)\|_2 \\ &\geq \omega\big(\rho+\tau\big) - 2B_j \\ &\approx \kappa \cdot \big(\rho+\tau\big) - (2 + 2L_{\mathcal{F}})\rho. \end{aligned} \tag{34}$$

Since ρ denotes a very small neighborhood radius and $\tau \gg \rho$, $\kappa \in (0, L_{\mathcal{F}})$ can easily satisfy $\kappa > \frac{2+2L_{\mathcal{F}}}{1+\frac{\tau}{\rho}} \to 0$, so that:

$$\|\mathbf{Y}_a - \tilde{\mathbf{Y}}_a\|_2^2 > \|\mathbf{Y}_a - \hat{\mathbf{Y}}_a\|_2^2. \tag{35}$$

This demonstrates that the output fusion mechanism directly amplifies the reconstruction or forecasting errors on anomalies while preserving low errors on normal data. □

A.2.2 Regularization Against Anomaly Overfitting

The contrastive normality constraint $\mathcal{L}_{\text{ConNC}}$ in Equation (8) also acts as a regularization term that prevents the NCL module from overfitting to abnormal patterns. Firstly, the contrastive loss enforces that the k transformation views $\{\tilde{\mathbf{z}}_n^{(i)}\}_{i=1}^k$ form k well-separated clusters in the normality subspace, with each cluster centered at $\mathbf{c}_i$. Formally, for any patch feature $\mathbf{z}_n$ from normal data:

$$\|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_i\|_2 + \gamma \leq \min_{j \neq i} \|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_j\|_2. \tag{36}$$

This margin-based separation ensures that normal features are tightly clustered around their respective centers, with inter-cluster distances exceeding intra-cluster distances by at least γ.

Then for anomalous patch features $\mathbf{z}_a$ derived from $(\mathbf{X}_a, \mathbf{Y}_a) \sim \mathcal{P}_a$, the transformation $\tilde{\mathbf{z}}_a^{(i)} = \mathbf{W}_i \mathbf{z}_a + \mathbf{b}_i$ attempts to project $\mathbf{z}_a$ onto the normality manifold. However, if $\mathbf{z}_a$ contains abnormal components orthogonal to all $\{\mathbf{v}_i\}_{i=1}^k$, the projection cannot fully eliminate these components. Let $\mathbf{z}_a = \mathbf{z}_a^{\parallel} + \mathbf{z}_a^{\perp}$, where $\mathbf{z}_a^{\parallel} \in \text{span}(\{\mathbf{v}_i\}_{i=1}^k)$ and $\mathbf{z}_a^{\perp} \in \text{span}(\{\mathbf{v}_i\}_{i=1}^k)^{\perp}$. The transformation only modifies $\mathbf{z}_a^{\parallel}$:

$$\tilde{\mathbf{z}}_a^{(i)} = \mathbf{W}_i(\mathbf{z}_a^{\parallel} + \mathbf{z}_a^{\perp}) + \mathbf{b}_i = \mathbf{W}_i \mathbf{z}_a^{\parallel} + \mathbf{b}_i + \mathbf{z}_a^{\perp}. \tag{37}$$

The residual component $\mathbf{z}_a^{\perp}$ causes $\tilde{\mathbf{z}}_a^{(i)}$ to deviate from the cluster centers:

$$\|\tilde{\mathbf{z}}_a^{(i)} - \mathbf{c}_i\|_2 \geq \|\mathbf{z}_a^{\perp}\|_2 - \|\mathbf{W}_i \mathbf{z}_a^{\parallel} + \mathbf{b}_i - \mathbf{c}_i\|_2. \tag{38}$$

If $\|\mathbf{z}_a^{\perp}\|_2$ is sufficiently large, $\tilde{\mathbf{z}}_a^{(i)}$ will not satisfy the margin constraint, leading to zero loss, which does not result in parameter updates and prevents the NCL from adapting to abnormal patterns.

B Theoretical Analysis for Normal Subspace Construction

To illustrate the effectiveness of normal subspace construction, we conduct theoretical analysis from two aspects: **(i)** few-shot normal samples are sufficient to construct an informative normality space (Appendix B.1); and **(ii)** top-k truncation effectively captures the essential normal information while suppressing noise (Appendix B.2).

B.1 Few-Shot SVD for Effective Estimation of the Normality Space

In this section, we demonstrate that the construction of an effective normal subspace does not require a large number of normal samples. Generally, normal patch features naturally lie in a low-dimensional subspace with a nontrivial spectral gap (Yu et al., 2026). Then we utilize standard matrix concentration and the Davis–Kahan theorem (Davis & Kahan, 1970) to prove that the leading singular subspace estimated from a finite set of normal patches converges to the population normal subspace, with estimation error decreasing as $O(N^{-1/2})$. Hence, patch features from a small number of normal samples can already provide a reliable estimate of the normality space.

Theorem 2 (Few-Shot Normal Subspace Construction)**.** *Let $\mathbf{z}_1, \ldots, \mathbf{z}_N \in \mathbb{R}^d$ be i.i.d. normal patch features with $\mathbb{E}[\mathbf{z}_i \mathbf{z}_i^\top] = \widehat{\mathbf{M}}$, where $\widehat{\mathbf{M}}$ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$, and nontrivial eigengap $\Delta_k = \lambda_k - \lambda_{k+1} > 0$. For sample estimation subspace $\mathbf{V}_k$ and population normal subspace $\widehat{\mathbf{V}}_k$, with probability at least $1 - \delta$,*

$$\left\| \sin\Theta(\mathbf{V}_k, \widehat{\mathbf{V}}_k) \right\|_2 = O\left(\frac{1}{\sqrt{N}}\right), \tag{39}$$

demonstrating that estimation error decreases as $O(1/\sqrt{N})$. Therefore, only a relatively small number of normal samples are sufficient to construct an informative normality space.

Proof. According to Equation (2), the sample estimation subspace is spanned by top-k right singular vectors of normal patch features $\mathbf{Z}$, formalized as:

$$\mathbf{V}_k = \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}. \tag{40}$$

Similarly, we can define the population top-k normal subspace as:

$$\widehat{\mathbf{V}}_k = \mathrm{span}\{\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \ldots, \widehat{\mathbf{v}}_k\}. \tag{41}$$

Let normal patch features $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N \in \mathbb{R}^d$ be i.i.d. drawn from the normal-feature distribution $\mathcal{Z}$, satisfying $\|\mathbf{z}_i\|_2^2 \le l$ almost surely. Then we can define the population and empirical second-moment matrices as:

$$\widehat{\mathbf{M}} = \mathbb{E}[\mathbf{z}\mathbf{z}^\top], \quad \mathbf{M} = \frac{1}{N}\mathbf{Z}^\top\mathbf{Z} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}_i\mathbf{z}_i^\top. \tag{42}$$

Since $\mathbf{Z} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^\top$, where $\mathbf{\Sigma}_r = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ with $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r \ge 0$, we have:

$$\mathbf{M} = \mathbf{V}_r\frac{\mathbf{\Sigma}_r^2}{N}\mathbf{V}_r^\top. \tag{43}$$

The r eigenvectors of $\mathbf{M}$ are exactly the r right singular vectors from $\mathbf{Z}$, and the corresponding eigenvalues are $\lambda_j = \frac{\sigma_j^2}{N}$, where $j \in \{1, 2, \ldots, r\}$. Then according to Davis–Kahan theorem (Davis & Kahan, 1970), we have:

$$\left\|\sin\Theta(\mathbf{V}_k, \widehat{\mathbf{V}}_k)\right\|_2 \le \frac{2\|\mathbf{M} - \widehat{\mathbf{M}}\|_2}{\Delta_k}, \tag{44}$$

where $\Theta(\mathbf{V}_k, \widehat{\mathbf{V}}_k) = \mathrm{diag}(\arccos\langle\mathbf{v}_1, \widehat{\mathbf{v}}_1\rangle, \ldots, \arccos\langle\mathbf{v}_k, \widehat{\mathbf{v}}_k\rangle)$, and $\Delta_k = \lambda_k - \lambda_{k+1} > 0$ is the eigengap. Then for each $i \in \{1, 2, \ldots, N\}$, we can define the random matrix as:

$$\mathbf{H}_i = \mathbf{z}_i\mathbf{z}_i^\top - \widehat{\mathbf{M}}. \tag{45}$$

By construction, $\mathbb{E}[\mathbf{H}_i] = \mathbf{0}$. The matrices $\{\mathbf{H}_i\}_{i=1}^N$ are independent and zero-mean.

Since $\mathbf{z}_i\mathbf{z}_i^\top$ is a rank-one positive semidefinite matrix with $\|\mathbf{z}_i\mathbf{z}_i^\top\|_2 = \|\mathbf{z}_i\|_2^2 \le l$, and $\widehat{\mathbf{M}} \preceq l \cdot \mathbf{I}$ (because $\widehat{\mathbf{M}} = \mathbb{E}[\mathbf{z}\mathbf{z}^\top]$ and $\|\mathbf{z}\|_2^2 \le l$), we have:

$$\|\mathbf{H}_i\|_2 = \left\|\mathbf{z}_i\mathbf{z}_i^\top - \widehat{\mathbf{M}}\right\|_2 \le \max\{\|\mathbf{z}_i\mathbf{z}_i^\top\|_2, \|\widehat{\mathbf{M}}\|_2\} \le l. \tag{46}$$

Then we compute:

$$\mathbb{E}[\mathbf{H}_i^2] = \mathbb{E}[(\mathbf{z}_i\mathbf{z}_i^\top - \widehat{\mathbf{M}}] = \mathbb{E}[(\mathbf{z}_i\mathbf{z}_i^\top)^2] - \widehat{\mathbf{M}}^2 \preceq \mathbb{E}[(\mathbf{z}_i\mathbf{z}_i^\top)^2]. \tag{47}$$

Since

$$(\mathbf{z}_i\mathbf{z}_i^\top)^2 = \mathbf{z}_i\mathbf{z}_i^\top\mathbf{z}_i\mathbf{z}_i^\top = \|\mathbf{z}_i\|_2^2\,\mathbf{z}_i\mathbf{z}_i^\top \preceq l \cdot \mathbf{z}_i\mathbf{z}_i^\top, \tag{48}$$

taking expectations, we have:

$$\mathbb{E}[\mathbf{H}_i^2] \preceq \mathbb{E}[(\mathbf{z}_i\mathbf{z}_i^\top)^2] \preceq l \cdot \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top] = l \cdot \widehat{\mathbf{M}}. \tag{49}$$

Therefore, we can obtain:

$$\left\|\sum_{i=1}^{N}\mathbb{E}[\mathbf{H}_i^2]\right\|_2 \le N \cdot l \cdot \|\widehat{\mathbf{M}}\|_2 = N \cdot l \cdot \widehat{\lambda}_1, \tag{50}$$

Where $\widehat{\lambda}_1 = \|\widehat{\mathbf{M}}\|_2$ is the largest eigenvalue of the population second-moment matrix. Then by the matrix Bernstein inequality (Tropp, 2015), for any $t \ge 0$, we have:

$$\mathbb{P}\left(\left\|\sum_{i=1}^{N}\mathbf{H}_i\right\|_2 \ge t\right) \le 2d \cdot \exp\left(\frac{-t^2/2}{N l\widehat{\lambda}_1 + lt/3}\right), \tag{51}$$

where d is the feature dimension. Therefore, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have:

$$\left\|\sum_{i=1}^{N}\mathbf{H}_i\right\|_2 \le \sqrt{2Nl\widehat{\lambda}_1 \ln\frac{2d}{\delta}} + \frac{2l}{3}\ln\frac{2d}{\delta}. \tag{52}$$

Dividing both sides by N, we obtain:

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{H}_i \right\|_2 \leq \sqrt{\frac{2l\widehat{\lambda}_1}{N} \log \frac{2d}{\delta}} + \frac{2l}{3N} \log \frac{2d}{\delta}. \tag{53}$$

According to Equation (45), since

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{H}_i = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i \mathbf{z}_i^\top - \widehat{\mathbf{M}}) = \frac{1}{N} \mathbf{Z}^\top \mathbf{Z} - \widehat{\mathbf{M}} = \mathbf{M} - \widehat{\mathbf{M}}, \tag{54}$$

we can establish the concentration inequality:

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_2 \leq \sqrt{\frac{2l\widehat{\lambda}_1}{N} \log \frac{2d}{\delta}} + \frac{2l}{3N} \log \frac{2d}{\delta}. \tag{55}$$

Combining this concentration bound with Inequality (44), with probability at least $1 - \delta$, we have:

$$\left\| \sin \Theta(\mathbf{V}_k, \widehat{\mathbf{V}}_k) \right\|_2 \leq \frac{2}{\Delta_k} \left[\sqrt{\frac{2l\widehat{\lambda}_1}{N} \log \frac{2d}{\delta}} + \frac{2l}{3N} \log \frac{2d}{\delta} \right]. \tag{56}$$

Thus, regarding $d, l, \delta, \widehat{\lambda}_1, \Delta_k$ as constants, asymptotically, we have:

$$\left\| \sin \Theta(\mathbf{V}_k, \widehat{\mathbf{V}}_k) \right\|_2 = O\left(\frac{1}{\sqrt{N}} \right). \tag{57}$$

The above analyses show that the error diminishes at a rate of $O(1/\sqrt{N})$ with a certain probability. For instance, when N is $5 \times p$ with only five normal samples, where p is the number of patches in a single sample, the error can be reduced to 0.1118 when $p = 16$; and it would be 0.0791 when $p = 32$.

Therefore, only a relatively small number of normal samples are sufficient to construct an informative normality space. The proof is completed. □

B.2 Effectiveness of the Top-k Truncation

Retaining only the top-k singular vectors from SVD to preserve dominant directions is a common practice in many tasks (Yan et al., 2022; Stoica et al., 2025; Li et al., 2026). This approach is intuitive and reasonable, since it removes many small singular value components that are not helpful for the current task, further highlighting the main patterns and eliminating potential interference.

However, there is still a lack of in-depth theoretical analysis to explore why the top-k truncation is effective. Focusing on our k-dimensional normal subspace construction, we conduct a detailed theoretical analysis from the following three aspects: **(i)** top-k truncated approximation is more effective than arbitrary k approximation; **(ii)** top-k approximation effectively retains the main information; **(iii)** top-k approximation effectively reduces noise interference. Then we derive Theorems 3, 4, and 5 (detailed proofs in Appendix B.2.1, B.2.2, and B.2.3) to explore the above three aspects.

Theorem 3. *Consider a well-trained decoder with weights $\boldsymbol{w}$ that can effectively recover patch feature matrix $\mathbf{Z}$ to the original series. Compared to the approximation of arbitrary k singular vectors $\bar{\mathbf{Z}} = \sum_{j=i_1}^{i_k} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top$, the top-k truncated approximation $\hat{\mathbf{Z}} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ can achieve lower output error:*

$$\mathbb{E}_{\mathbf{Z} \sim \mathcal{Z}} \left[\|\mathbf{Z}\boldsymbol{w} - \hat{\mathbf{Z}}\boldsymbol{w}\|_2^2 \right] \leq \mathbb{E}_{\mathbf{Z} \sim \mathcal{Z}} \left[\|\mathbf{Z}\boldsymbol{w} - \bar{\mathbf{Z}}\boldsymbol{w}\|_2^2 \right]. \tag{58}$$

Theorem 3 indicates that the top-k truncated approximation can retain more effective information than arbitrary k approximation, thus achieving lower output error. However, it remains unclear to what extent this information is being retained. Then we derive Theorem 4 to illustrate that for each patch feature, there is an upper bound to the difference between the top-k approximation information and the original information, thus effectively retaining the dominant patterns.

Theorem 4. *$\forall \mathbf{z}_n \in \mathbf{Z}$, $\exists \mathbf{z}'_n$ from the k-dimensional normal subspace, such that $\|\mathbf{z}_n - \mathbf{z}'_n\|_2 \leq \sigma_{k+1}$.*

Theorems 3 and 4 mainly prove the validity of the k-dimensional normal subspace spanned by the top-k right singular vectors from different perspectives of dominant information retention. Then we would like to investigate whether our top-k approximation can effectively eliminate the noise interference if there are noise disturbances in the original patch feature matrix $\mathbf{Z}$. For instance, the noise interference could cause once there is inherent noise in the original input or irrelevant information introduced during the encoding process. Finally, we derive Theorem 5 to demonstrate the effectiveness of using top-k approximation to reduce noise.

Theorem 5. *Suppose that $\mathbf{Z}$ is composed of a clean data matrix $\mathbf{E}$ and a white noise matrix $\mathbf{D}$, i.e., $\mathbf{Z} = \mathbf{E} + \mathbf{D}$, and the clean data is uncorrelated with the noise. To estimate $\mathbf{E}$, the best rank k matrix in the least-squares sense is $\hat{\mathbf{E}}_{\mathrm{LS}} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, which minimizes the following squared error:*

$$\min_{\hat{\mathbf{E}}} \|\hat{\mathbf{E}} - \mathbf{E}\|_F^2. \tag{59}$$

B.2.1 Proof of Theorem 3

Proof. The singular value decomposition of $\mathbf{Z} \in \mathbb{R}^{N \times d}$ is $\mathbf{Z} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, where $r = \mathrm{rank}(\mathbf{Z})$. We denote the sorted index subset obtained by random selection as $\mathcal{I} = \{i_1, i_2, \ldots, i_k\}$, where $i_1 < i_2 < \cdots < i_k$, and the remaining sorted index subset is $\hat{\mathcal{I}} = \{i_{k+1}, i_{k+2}, \ldots, i_r\}$. Then we can calculate the error matrix resulting from the arbitrary approximation:

$$\mathbf{Z} - \bar{\mathbf{Z}} = \sum_{j=i_{k+1}}^{i_r} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top. \tag{60}$$

Similarly, the error matrix resulting from the top-k truncated approximation is:

$$\mathbf{Z} - \hat{\mathbf{Z}} = \sum_{i=k+1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top. \tag{61}$$

Consider the well-trained decoder with weights $\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_s\} \in \mathbb{R}^{d \times s}$, where s is the patch size. It is worth noting that $\mathbf{Z}\boldsymbol{w}$ denotes the best restoration of the original series, thus, a lower error compared to $\mathbf{Z}\boldsymbol{w}$ implies that more effective information of the matrix $\mathbf{Z}$ has been approximately retained. For $\boldsymbol{w}_l \in \boldsymbol{w}$, the corresponding point errors for arbitrary approximation and top-k truncated approximation are as follows:

$$\begin{aligned}
(\mathbf{Z} - \bar{\mathbf{Z}})\boldsymbol{w}_l &= \left(\sum_{j=i_{k+1}}^{i_r} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top \right) \boldsymbol{w}_l = \sum_{j=i_{k+1}}^{i_r} \sigma_j \mathbf{u}_j (\mathbf{v}_j^\top \boldsymbol{w}_l), \\
(\mathbf{Z} - \hat{\mathbf{Z}})\boldsymbol{w}_l &= \left(\sum_{i=k+1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \right) \boldsymbol{w}_l = \sum_{i=k+1}^{r} \sigma_i \mathbf{u}_i (\mathbf{v}_i^\top \boldsymbol{w}_l).
\end{aligned} \tag{62}$$

Considering the above point-by-point error calculation where $\mathbf{v}_j^\top \boldsymbol{w}_l$ is a scalar, we can sum up the above to form the linear combination of orthogonal normalized vectors $\mathbf{u}_i$. Then we have:

$$\begin{aligned}
\|(\mathbf{Z} - \bar{\mathbf{Z}})\boldsymbol{w}\|_2^2 &= \left\| \sum_{l=1}^{s} \sum_{j=i_{k+1}}^{i_r} (\sigma_j \mathbf{v}_j^\top \boldsymbol{w}_l) \mathbf{u}_j \right\|_2^2 = \sum_{l=1}^{s} \sum_{j=i_{k+1}}^{i_r} (\sigma_j \mathbf{v}_j^\top \boldsymbol{w}_l)^2, \\
\|(\mathbf{Z} - \hat{\mathbf{Z}})\boldsymbol{w}\|_2^2 &= \left\| \sum_{l=1}^{s} \sum_{i=k+1}^{r} (\sigma_i \mathbf{v}_i^\top \boldsymbol{w}_l) \mathbf{u}_i \right\|_2^2 = \sum_{l=1}^{s} \sum_{i=k+1}^{r} (\sigma_i \mathbf{v}_i^\top \boldsymbol{w}_l)^2.
\end{aligned} \tag{63}$$

Since $\boldsymbol{w}$ is assumed to be well-trained, the contribution strength of the right singular vector $\mathbf{v}_i$ to the decoding output monotonically increases with the corresponding singular value σ_i. Thus we have: $c_i \triangleq \sum_{l=1}^{s} (\mathbf{v}_i^\top \boldsymbol{w}_l)^2$, where $c_1 \geq c_2 \geq \cdots \geq c_r \geq 0$. Then we can rewrite Equation (63) as follows:

$$\|(\mathbf{Z} - \bar{\mathbf{Z}})\boldsymbol{w}\|_2^2 = \sum_{j=i_{k+1}}^{i_r} \sigma_j^2 c_j, \quad \|(\mathbf{Z} - \hat{\mathbf{Z}})\boldsymbol{w}\|_2^2 = \sum_{i=k+1}^{r} \sigma_i^2 c_i. \tag{64}$$

Noting $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, and $\hat{\mathcal{I}} = \{i_{k+1}, \ldots, i_r\}$ represents indices from the arbitrary selection while $\{\sigma_{k+1}, \ldots, \sigma_r\}$ represents the smallest singular values, thus, we have: $\sigma_{i_r} \geq \sigma_r, \cdots, \sigma_{i_{k+1}} \geq \sigma_{k+1}$ and $c_{i_r} \geq c_r, \cdots, c_{i_{k+1}} \geq c_{k+1}$, with the fact $i_r \leq r, \cdots, i_{k+1} \leq k+1$. Then for each item in Equation (64), we know $\sigma_{i_m}^2 c_{i_m} \geq \sigma_m^2 c_m$ since $i_m \leq m$, and obtain:

$$\sum_{j=i_{k+1}}^{i_r} \sigma_j^2 c_j \geq \sum_{i=k+1}^{r} \sigma_i^2 c_i. \tag{65}$$

By taking the expectation over the feature distribution $\mathcal{Z}$, we can obtain the following inequality:

$$\mathbb{E}_{\mathbf{Z}\sim\mathcal{Z}}\left[\|\mathbf{Z}\boldsymbol{w} - \hat{\mathbf{Z}}\boldsymbol{w}\|_2^2\right] \leq \mathbb{E}_{\mathbf{Z}\sim\mathcal{Z}}\left[\|\mathbf{Z}\boldsymbol{w} - \bar{\mathbf{Z}}\boldsymbol{w}\|_2^2\right]. \tag{66}$$

□

B.2.2 Proof of Theorem 4

Proof. Since $\mathbf{Z}$ is a $N \times d$ matrix, *i.e.*, $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]^\top$, where $\mathbf{z}_n \in \mathbb{R}^d$ is the patch feature. After performing SVD on $\mathbf{Z}$, *i.e.*, $\mathbf{Z} = \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^\top$, we have:

$$\begin{bmatrix} \mathbf{z}_1^\top \\ \mathbf{z}_2^\top \\ \vdots \\ \mathbf{z}_N^\top \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{u}}_1 \\ \vec{\mathbf{u}}_2 \\ \vdots \\ \vec{\mathbf{u}}_N \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_r^\top \end{bmatrix}, \tag{67}$$

where $\mathbf{U}_r = [\vec{\mathbf{u}}_1^\top, \vec{\mathbf{u}}_2^\top, \ldots, \vec{\mathbf{u}}_N^\top]^\top \in \mathbb{R}^{N\times r}$, $\mathbf{V}_r = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r] \in \mathbb{R}^{d\times r}$. Then we expand each item u_{ij} in $\mathbf{U}_r$, and obtain:

$$\begin{bmatrix} \mathbf{z}_1^\top \\ \mathbf{z}_2^\top \\ \vdots \\ \mathbf{z}_N^\top \end{bmatrix} = \begin{bmatrix} \sigma_1 \cdot u_{11} & \sigma_2 \cdot u_{12} & \cdots & \sigma_r \cdot u_{1r} \\ \sigma_1 \cdot u_{21} & \sigma_2 \cdot u_{22} & \cdots & \sigma_r \cdot u_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1 \cdot u_{N1} & \sigma_2 \cdot u_{N2} & \cdots & \sigma_r \cdot u_{Nr} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_r^\top \end{bmatrix}. \tag{68}$$

For each patch feature $\mathbf{z}_n \in \mathbf{Z}$, we have:

$$\mathbf{z}_n^\top = \begin{bmatrix} \sigma_1 \cdot u_{n1} & \sigma_2 \cdot u_{n2} & \cdots & \sigma_r \cdot u_{nr} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \vdots \\ \mathbf{v}_r^\top \end{bmatrix}. \tag{69}$$

Then we can get the summation form:

$$\mathbf{z}_n^\top = \sum_{i=1}^{r} \sigma_i \cdot u_{ni}\mathbf{v}_i^\top = \sum_{i=1}^{k} \sigma_i \cdot u_{ni}\mathbf{v}_i^\top + \sum_{i=k+1}^{r} \sigma_i \cdot u_{ni}\mathbf{v}_i^\top. \tag{70}$$

Note that we can construct feature $\mathbf{z}'_n$ from the k-dimensional normal subspace, *i.e.*, $\mathbf{z}_n'^\top = \sum_{i=1}^{k} \sigma_i \cdot u_{ni}\mathbf{v}_i^\top$. Then we have:

$$\left\|\mathbf{z}_n^\top - \mathbf{z}_n'^\top\right\|_2 = \left\|\sum_{i=k+1}^{r} \sigma_i \cdot u_{ni}\mathbf{v}_i^\top\right\|_2 = \sqrt{\sum_{i=k+1}^{r} (\sigma_i \cdot u_{ni})^2}. \tag{71}$$

Since all the vectors in $\mathbf{U}_r$ are mutually orthogonal and normalized, we know $\|\vec{\mathbf{u}}_n\| = 1$, and $\left\|\vec{\mathbf{u}}_n^{[k+1,r]}\right\| \leq 1$, where $\vec{\mathbf{u}}_n^{[k+1,r]}$ is the sub-vectors of $\vec{\mathbf{u}}_n$ from $k+1$-th to r-th dimensions. Noting that the singular values are decreasing, *i.e.*, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, then we have the following upper bound:

$$\left\|\mathbf{z}_n^\top - \mathbf{z}_n'^\top\right\|_2 \leq \sigma_{k+1}\sqrt{\sum_{i=k+1}^{r} u_{ni}^2} \leq \sigma_{k+1}. \tag{72}$$

□

B.2.3 PROOF OF THEOREM 5

Proof. Since clean data is assumed to be uncorrelated with the white noise, we know that the clean data matrix $\mathbf{E}$ is orthogonal to the noise matrix $\mathbf{D}$, *i.e.*, $\mathbf{E}^\top\mathbf{D} = 0$. We also have $\mathbf{D}^\top\mathbf{D} = \delta^2\mathbf{I}$ since noise is white and uncorrelated, where δ^2 is the noise variance. Our goal is to minimize $\|\hat{\mathbf{E}} - \mathbf{E}\|_F^2$. Because of $\mathbf{Z} = \mathbf{E} + \mathbf{D}$, we have:

$$\|\mathbf{Z} - \hat{\mathbf{E}}\|_F^2 = \|\mathbf{E} - \hat{\mathbf{E}} + \mathbf{D}\|_F^2 = \|\mathbf{E} - \hat{\mathbf{E}}\|_F^2 + \|\mathbf{D}\|_F^2 + 2\mathrm{tr}[(\mathbf{E} - \hat{\mathbf{E}})^\top\mathbf{D}]. \tag{73}$$

Since $\mathbf{E}^\top\mathbf{D} = 0$ and the estimated signal is uncorrelated with noise in the expectation sense, thus $\mathrm{tr}[(\mathbf{E} - \hat{\mathbf{E}})^\top\mathbf{D}] = 0$, and we have $\|\mathbf{D}\|_F^2 = \mathrm{tr}(\mathbf{D}^\top\mathbf{D}) = \mathrm{tr}(\delta^2\mathbf{I}) = \delta^2 \cdot d$, then we can obtain:

$$\|\mathbf{Z} - \hat{\mathbf{E}}\|_F^2 = \|\mathbf{E} - \hat{\mathbf{E}}\|_F^2 + \delta^2 \cdot d, \tag{74}$$

where $\delta^2 \cdot d$ is a constant term. Therefore, the minimization of $\|\hat{\mathbf{E}} - \mathbf{E}\|_F^2$ is equivalent to the minimization of $\|\mathbf{Z} - \hat{\mathbf{E}}\|_F^2$. Suppose that clean data matrix $\mathbf{E}$ is rank deficient, *i.e.*, $\mathrm{rank}(\mathbf{E}) = k$, then we also restrict $\hat{\mathbf{E}}$ to the matrix with a rank not exceeding k. Therefore, minimizing $\|\hat{\mathbf{E}} - \mathbf{E}\|_F^2$ becomes:

$$\min_{\mathrm{rank}(\hat{\mathbf{E}}) \le k} \|\mathbf{Z} - \hat{\mathbf{E}}\|_F^2. \tag{75}$$

According to the Eckart–Young–Mirsky theorem (Eckart & Young, 1936), for any matrix $\mathbf{Z}$, the optimal approximation with a rank no greater than k is the truncation of top-k singular values of $\mathbf{Z}$. Therefore, the best least-squares estimate of the clean data matrix $\mathbf{E}$ is the top-k truncated approximation of $\mathbf{Z}$:

$$\hat{\mathbf{E}}_{\mathrm{LS}} = \hat{\mathbf{Z}} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{76}$$

where σ_i, $\mathbf{u}_i$, and $\mathbf{v}_i$ denote the singular values, left singular vectors, and right singular vectors of $\mathbf{Z}$ respectively. We complete the proof. □

C PROOF OF PROPOSITION 1 (SPECTRAL AND GEOMETRIC PROPERTY)

Proof. For any $\mathbf{k} \in \mathbf{v}_i^\perp$ (*i.e.*, $\mathbf{v}_i^\top\mathbf{k} = 0$), we have:

$$\mathbf{W}_i\mathbf{k} = (\mathbf{I} - \beta_i\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{k} = \mathbf{k} - \beta_i\mathbf{v}_i(\mathbf{v}_i^\top\mathbf{k}) = \mathbf{k}. \tag{77}$$

Thus each vector in $\mathbf{v}_i^\perp$ is an eigenvector with eigenvalue 1. Since $\mathbf{v}_i \in \mathbb{R}^d$, we can select the remaining $d-1$ directions that are orthogonal to $\mathbf{v}_i$ and combine them to form $\mathbf{v}_i^\perp$. Then we can know that $\mathbf{W}_i$ has $(d-1)$ eigenvalues of 1 since $\dim(\mathbf{v}_i^\perp) = d - 1$.

Similarly, for the vector $\mathbf{v}_i$ itself, we have:

$$\mathbf{W}_i\mathbf{v}_i = (\mathbf{I} - \beta_i\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{v}_i = \mathbf{v}_i - \beta_i\mathbf{v}_i(\mathbf{v}_i^\top\mathbf{v}_i) = (1 - \beta_i)\mathbf{v}_i. \tag{78}$$

Therefore, $\mathbf{W}_i$ has an eigenvector $\mathbf{v}_i$ with eigenvalue $1 - \beta_i$. Overall, The spectrum of $\mathbf{W}_i$ is $\Lambda(\mathbf{W}_i) = \{1 - \beta_i, 1, \ldots, 1\}$. This indicates that since $\mathbf{W}_i$ is a rank-1 perturbation of the identity, it leaves all components orthogonal to $\mathbf{v}_i$ unchanged and only rescales the projection along $\mathbf{v}_i$ by a factor of $(1 - \beta_i)$.

To more intuitively present how β_i governs the geometric behavior of the transformation $\tilde{\mathbf{z}}_n^{(i)} = \mathbf{W}_i\mathbf{z}_n + \mathbf{b}_i$, we compute the corresponding projection of $\tilde{\mathbf{z}}_n^{(i)}$ onto $\mathbf{v}_i$:

$$\begin{aligned}
\mathbf{v}_i^\top\tilde{\mathbf{z}}_n^{(i)} &= \mathbf{v}_i^\top(\mathbf{I} - \beta_i\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{z}_n + \mathbf{v}_i^\top\alpha_i\beta_i\mathbf{v}_i \\
&= (1 - \beta_i)\mathbf{v}_i^\top\mathbf{z}_n + \alpha_i\beta_i \\
&= \alpha_i + (1 - \beta_i)(\mathbf{v}_i^\top\mathbf{z}_n - \alpha_i).
\end{aligned} \tag{79}$$

Equation (79) reflects explicit geometric regimes: **(i) identity mapping** ($\beta_i < 1, \beta_i \to 0$): the update tends to retain partial old information, and remains unchanged when $\beta_i = 0$. **(ii) complete replacement** ($\beta_i = 1$): the updated projection is exactly replaced by α_i to reflect normality. **(iii) over-calibration** ($\beta_i > 1$): the update tends to further reversely calibrate the old information.

Since each patch feature involves varying degrees of normality, our learnable parameters α_i and β_i can adaptively adjust each feature to the normality manifold. This allows the model to learn under-, moderate, and over-corrective behaviors where beneficial. □

D EXPERIMENTAL DETAILS

D.1 DATASETS

We conduct comprehensive experiments on ten TSAD datasets, including five univariate and five multivariate datasets. Table 3 summarizes the detailed statistics of these datasets. The characteristics of each dataset are introduced below:

- **IOPS** (Ren et al., 2019): A large-scale dataset containing Key Performance Indicators (KPIs) from Internet companies, with time series collected from real-world IT operations. The dataset includes various anomaly patterns such as spikes, level shifts, and gradual changes.
- **UCR** (Wu & Keogh, 2021): A collection of univariate time series from diverse domains including medicine, power demand, air temperature, and so on. The anomalies are typically point anomalies or short subsequence anomalies.
- **NAB** (Ahmad et al., 2017): The Numenta Anomaly Benchmark consists of labeled real-world and artificial time series with point and contextual anomalies. It includes data from various sources such as online advertisement clicks, machine temperature sensors, and traffic data.
- **MSL** (Benecki et al., 2021): Mars Science Laboratory dataset containing telemetry data from spacecraft monitoring systems. It consists of time series with anomalies caused by various spacecraft operational issues. Since only one feature represents a sensor measurement and the remaining features are binary-encoded commands, we transform the original multivariate time series into univariate ones, which is consistent with TSB-UAD (Paparrizos et al., 2022b) and TAB (Qiu et al., 2025).
- **SMAP** (Benecki et al., 2021): Soil Moisture Active Passive satellite dataset containing spacecraft telemetry data. Anomalies represent spacecraft component failures and unusual sensor behaviors. Like the processing of MSL, the multivariate time series are transformed into univariate ones.
- **PSM** (Abdulaal et al., 2021): Pooled Server Metrics dataset is collected internally from multiple application server nodes as eBay, containing monitoring server performance metrics such as CPU usage, memory consumption, and network traffic. The dataset includes complex multivariate anomaly patterns.
- **SMD** (Su et al., 2019): Server Machine Dataset is collected from a large Internet company's server monitoring system. It contains various types of server anomalies, including hardware failures and software bugs. Anomalies are annotated by domain experts.
- **SWAT** (Mathur & Tippenhauer, 2016): Secure Water Treatment dataset is collected from a real-world water treatment testbed. Anomalies are caused by attack scenarios targeting critical infrastructure, making it particularly challenging.
- **PUMP** (Feng & Tian, 2021): A dataset collected from a water pumping system with multiple sensors monitoring pressure, flow rate, and motor status. The anomalies are caused by mechanical failures and sensor malfunctions.
- **NYC** (Cui et al., 2016): New York City taxi demand dataset containing traffic volume measurements. Anomalies correspond to unusual traffic patterns caused by special events or incidents.

For all univariate time series, we follow the data filtering strategy proposed by TAB benchmark (Qiu et al., 2025) to ensure data quality. Finally, we include a total of 361 univariate and multivariate time series. For all datasets, we follow the standard train-test split provided by the original sources (Qiu et al., 2025). Only five normal samples from the training set are utilized to learn the normality constraint, while the test set includes both normal and anomalous observations for evaluation.

D.2 BASELINES

We evaluate NCL on ten state-of-the-art foundation models, including six forecasting-based, two reconstruction-based, and two forecasting-reconstruction-combined TSFMs:

- **Aurora** (Wu et al., 2026): A multimodal foundation model pre-trained on a cross-domain multimodal time series corpus with generative probabilistic forecasting. It demonstrates strong zero-shot forecasting capabilities on both unimodal and multimodal scenarios.

Table 3: Statistics of ten datasets (Dim: dimensionality).

Dataset	Domain	Dim	Avg Train Length	Avg Test Length	Avg Anomaly Ratio (%)	Series Count
IOPS	Web	1	10,142.4	91,288.3	2.15	11
UCR	Multiple	1	18,608.6	47,294	0.58	243
NAB	Multiple	1	3,557.8	3,557	9.84	45
MSL	Spacecraft	1	2,269.8	2,896.8	3.85	22
SMAP	Spacecraft	1	2,608.8	7,929.7	2.20	35
PSM	Server Machine	25	132,481	87,841	11.07	1
SMD	Server Machine	38	708,405	708,420	2.08	1
SWAT	Water Treatment	51	495,000	449,919	5.78	1
PUMP	Water Treatment	44	76,901	143,401	6.54	1
NYC	Transport	3	13,104	4,416	0.57	1

- **ROSE** (Wang et al., 2025): A foundation model pre-trained with both reconstruction and forecasting tasks. For downstream tasks, only the pre-trained prediction head is used to fine-tune the model.
- **Time-MoE** (Shi et al., 2025): A mixture-of-experts foundation model that uses a sparse gating mechanism to select specialized experts for different temporal patterns. Pre-trained on 300 billion time points with multi-resolution forecasting objectives.
- **Chronos** (Ansari et al., 2024): A probabilistic forecasting model based on language modeling techniques. It tokenizes time series into discrete bins and applies autoregressive forecasting pre-training on massive time series datasets.
- **TimesFM** (Das et al., 2024): A decoder-only foundation model pre-trained on Google's internal time series datasets with point forecasting objectives.
- **TTM** (Ekambaram et al., 2024): A lightweight foundation model with channel-mixing and time-mixing layers. Despite its compact size, it achieves competitive performance through efficient architectural design.
- **MOMENT** (Goswami et al., 2024): A masked autoencoder foundation model pre-trained on the Time Series Pile dataset. It uses a transformer encoder to reconstruct randomly masked time series patches, learning robust temporal representations.
- **UniTS** (Gao et al., 2024): A unified multi-task foundation model pre-trained with masked reconstruction. It demonstrates strong generalization across various downstream tasks.
- **TSPulse** (Ekambaram et al., 2026): An ultra-lightweight foundation model with disentangled representation learning. It pre-trains with masked reconstruction augmented by explicit disentanglement across temporal, spectral, and semantic spaces, and employs lightweight post-hoc fusers to selectively combine these views for task-specific adaptation.
- **SEMPO** (He et al., 2025): A lightweight foundation model featuring energy-aware spectral decomposition and mixture-of-prompts enabled Transformer. It models both high-energy and low-energy informative frequency signals for efficient pre-training, and employs dataset-specific prompts with adaptive routing for parameter-efficient cross-domain adaptation.

For anomaly detection, all TSFMs are applied in a zero-shot manner without fine-tuning their pre-trained parameters. We use the forecasting error (for forecasting-based TSFMs), reconstruction error (for reconstruction-based TSFMs), or combined error of forecasting and reconstruction (for forecasting-reconstruction-combined TSFMs) as the baseline anomaly score. NCL is then applied as a plug-and-play framework to enhance their anomaly detection capabilities.

Additionally, we compare NCL-augmented TSFMs with several deep TSAD methods, simple one-liner baselines, and TSAD-specific foundation models in Appendix H, I, and J, respectively.

Conventional deep TSAD methods (*i.e.*, methods trained from scratch on each dataset) include CroAD (Li et al., 2025), ConAD (Zhuang et al., 2024), TsNet (Wu et al., 2023), and Dlinear (Zeng et al., 2023).

One-liner baselines (*i.e.*, simple baseline methods requiring no training) include LOF (Local Outlier Factor), Last-1 (last value as prediction), Last-3 (average of last 3 values), Var-100 (100-window moving variance), and Centered-5 (average of the 2 neighboring values before and after).

TSAD-specific foundation models include DADA (Shentu et al., 2025), TimeRCD (Lan et al., 2026), and TimeRadar (He et al., 2026). These approaches aim to learn discriminative representations by pre-training on large-scale datasets augmented with synthetic anomalies. DADA employs dual adversarial decoders to explicitly distinguish normal from abnormal patterns using injected anomalies, while TimeRCD conducts relative context discrepancy pre-training on synthetic anomalies. Pre-training on the anomaly-injected time series corpus released by DADA, TimeRadar identifies diverse anomalies in a domain-rotatable time–frequency space.

D.3 Implementation Details

We utilize the TAB (Qiu et al., 2025) code repository for unified evaluation and follow its default parameter configuration. For all ten TSFMs, we rely on code from:

- **Aurora**: `https://github.com/decisionintelligence/Aurora`
- **ROSE**: `https://github.com/decisionintelligence/ROSE`
- **Time-MoE**: `https://github.com/Time-MoE/Time-MoE`
- **Chronos**: `https://github.com/amazon-science/chronos-forecasting`
- **TimesFM**: `https://github.com/google-research/timesfm`
- **TTM**: `https://github.com/ibm-granite/granite-tsfm/tree/main/tsfm_public/models/tinytimemixer`
- **MOMENT**: `https://github.com/moment-timeseries-foundation-model/moment`
- **UniTS**: `https://github.com/mims-harvard/UniTS`
- **TSPulse**: `https://github.com/ibm-granite/granite-tsfm`
- **SEMPO**: `https://github.com/mala-lab/SEMPO`

Several deep TSAD methods (CroAD, ConAD, TsNet, and Dlinear) are adopted from `https://github.com/decisionintelligence/TAB`. LOF is adapted from the pyod package `https://github.com/yzhao062/pyod`. Last-1, Last-3, Var-100, and Centered-5 are adopted from `https://gitlab.kuleuven.be/m-group-campus-brugge/dtai_public/publications/iclr2026_timeseriesfoundationmodelsad/`. DADA is from the official code repository `https://github.com/iambowen/DADA`. TimeRCD is from `https://github.com/thu-sail-lab/Time-RCD`. TimeRadar is from `https://github.com/mala-lab/TimeRadar`.

For NCL hyperparameters, the number of principal directions (top-k singular vectors) is set to $k = 16$ for all experiments, and we also conduct a comprehensive sensitivity analysis in Appendix P. The margin γ in the contrastive normality constraint (Equation (8)) is set to 1.0. The learning rate is 1e-4 with the Adam optimizer. We train NCL for 5 epochs with only 5 normal samples randomly selected from the training set. During training, we freeze all parameters of the pre-trained TSFM and only optimize the parameters in NCL. All experiments are conducted on a workstation with an Intel(R) Xeon(R) Gold 6426Y CPU and a single NVIDIA A100 GPU (80GB memory).

E ALGORITHMS DETAILS

Algorithm 1 details the complete training procedure of NCL, including normal subspace construction, patch-adaptive normality transformation, contrastive normality constraint, and output fusion. Only the NCL module parameters are updated while the TSFM $\mathcal{F}$ remains frozen. The training procedure emphasizes five key aspects: **(i)** The TSFM $\mathcal{F}$ is kept frozen throughout training, preserving its pre-trained knowledge; **(ii)** The SVD identifies the principal directions of normal variation in the feature space; **(iii)** Transforming each feature onto the normality manifold along each direction; **(iv)** The contrastive normality constraint encourages that transformed features form well-separated clusters corresponding to distinct normal directions; and **(v)** The gated fusion mechanism adaptively combines NCL predictions with original TSFM outputs to amplify normal properties.

Algorithm 1 Training Procedure of NCL

Input: Frozen TSFM $\mathcal{F}$, normal training samples $\{\mathbf{X}_t, \mathbf{Y}_t\}_{t=1}^{N_{\text{train}}}$, learning rate η, margin γ, number of epochs E, number of principal directions k, trainable parameters $\mathbf{W}_\alpha, \mathbf{W}_\beta, \mathbf{b}_\alpha, \mathbf{b}_\beta$, decoder weights, fusion weights $\boldsymbol{\lambda}$

Output: Trained NCL parameters, top-k right singular vectors $\{\mathbf{v}_i\}_{i=1}^k$, cluster centers $\{\mathbf{c}_i\}_{i=1}^k$

Initialize learnable parameters: $\mathbf{W}_\alpha, \mathbf{W}_\beta \in \mathbb{R}^{k\times d}$, $\mathbf{b}_\alpha, \mathbf{b}_\beta \in \mathbb{R}^k$
Initialize decoder head and fusion weights $\boldsymbol{\lambda} \in (0,1)^D$
for epoch $e = 1$ to E **do**
  Select all the samples from $\{\mathbf{X}_t, \mathbf{Y}_t\}_{t=1}^{N_{\text{train}}}$ to form batch $\{(\mathbf{X}_t, \mathbf{Y}_t)\}_{t\in\mathcal{B}}$
  $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N = \mathcal{F}(\mathbf{X}_\mathcal{B})$ {Extract patch features from frozen TSFM}
  $\hat{\mathbf{Y}}_\mathcal{B} = \text{Decoder}_\mathcal{F}(\mathbf{Z})$ {Obtain TSFM predictions}
  $\mathbf{Z} = \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^\top$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ {Perform SVD on feature matrix}
  $\{\mathbf{v}_i\}_{i=1}^k \leftarrow$ columns 1 to k of $\mathbf{V}_r$ {Retain top-k right singular vectors}
  for each patch feature $\mathbf{z}_n$ in $\mathbf{Z}$ **do**
    $\boldsymbol{\alpha}_n = \sigma(\mathbf{W}_\alpha\mathbf{z}_n + \mathbf{b}_\alpha) \in (0,1)^k$
    $\boldsymbol{\beta}_n = 2\cdot\sigma(\mathbf{W}_\beta\mathbf{z}_n + \mathbf{b}_\beta) \in (0,2)^k$
    for each principal direction $i = 1$ to k **do**
      $\tilde{\mathbf{z}}_n^{(i)} = (\mathbf{I} - \beta_i\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{z}_n + \alpha_i\beta_i\mathbf{v}_i$ {Normality Transformation, Equation (6)}
    end for
    $\tilde{\mathbf{z}}_n = \frac{1}{k}\sum_{i=1}^k \tilde{\mathbf{z}}_n^{(i)}$ {Aggregate views}
  end for
  for each direction $i = 1$ to k **do**
    $\mathbf{c}_i = \frac{1}{N}\sum_{n=1}^N \tilde{\mathbf{z}}_n^{(i)}$
  end for
  Initialize $\mathcal{L}_{\text{ConNC}} = 0$
  for each patch $n = 1$ to N **do**
    for each direction $i = 1$ to k **do**
      $d_{\text{pos}} = \|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_i\|_2$ {Compute positive pair distance}
      $d_{\text{neg}} = \min_{j\neq i}\|\tilde{\mathbf{z}}_n^{(i)} - \mathbf{c}_j\|_2$ {Compute negative pair distances}
      $\mathcal{L}_{\text{ConNC}} \leftarrow \mathcal{L}_{\text{ConNC}} + \max(0, d_{\text{pos}} + \gamma - d_{\text{neg}})$ {Accumulate triplet loss}
    end for
  end for
  $\mathcal{L}_{\text{ConNC}} = \mathcal{L}_{\text{ConNC}}/(N\cdot k)$ {Equation (8)}
  $\tilde{\mathbf{Y}}_\mathcal{B}^{\text{ncl}} = \text{Decoder}_{\text{ncl}}([\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_N])$ {Decode calibrated features}
  $\tilde{\mathbf{Y}}_\mathcal{B} = \boldsymbol{\lambda}\odot\tilde{\mathbf{Y}}_\mathcal{B}^{\text{ncl}} + (1-\boldsymbol{\lambda})\odot\hat{\mathbf{Y}}_\mathcal{B}$ {Fuse with TSFM output, Equation (9)}
  $\mathcal{L}_{\text{Pred}} = \frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}\|\tilde{\mathbf{Y}}_t - \mathbf{Y}_t\|_2^2$ {Compute prediction loss}
  $\mathcal{L} = \mathcal{L}_{\text{Pred}} + \mathcal{L}_{\text{ConNC}}$ {Total loss, Equation (11)}
  $\theta_{\text{NCL}} \leftarrow \theta_{\text{NCL}} - \eta\nabla_{\theta_{\text{NCL}}}\mathcal{L}$ {Update NCL parameters via gradient descent}
end for
return Trained NCL parameters, $\{\mathbf{v}_i\}_{i=1}^k$, $\{\mathbf{c}_i\}_{i=1}^k$

F Computational Complexity Analysis

We further analyze the computational complexity of NCL. Given N patches from few-shot normal training samples, feature dimension d, and k principal directions, the normal subspace construction via SVD requires $\mathcal{O}(N^2 \cdot d)$ one-time cost. Then for per-iteration complexity, TSFM feature extraction requires $\mathcal{O}(N \cdot d \cdot C_{\mathcal{F}})$ where $C_{\mathcal{F}}$ is TSFM complexity (frozen, no gradients); normality transformation is $\mathcal{O}(N \cdot k \cdot d)$ for all k views; $\mathcal{O}(N \cdot k^2 \cdot d)$ is needed for distance computations; and lightweight projection requires $\mathcal{O}(k \cdot d)$. Since $k \ll d$ and the TSFM is frozen, we can effectively conduct NCL training in approximate $\mathcal{O}(N \cdot k^2 \cdot d)$ time, which is linear with dimensionality d. The comparisons in Section 5.4 and Appendix G empirically validate the computational efficiency of NCL.

G Detailed Efficiency Comparison with Other TSFM Adaptation Methods

Figure 5 provides a detailed efficiency comparison of NCL against other TSFM adaptation methods across different TSFMs on multivariate time series. The results consistently demonstrate NCL's superior efficiency. By freezing the TSFM backbone and only optimizing lightweight projection heads, NCL requires minimal parameter update overhead during training. In contrast, LoRA must store gradients for updating key TSFM parameters, and CoRA needs more parameters to model the cross-channel association. It is worth noting that NCL has slightly more trainable parameters compared to LoRA on reconstruction-based TSFMs (MOMENT and UniTS). This is because the reconstruction is performed on the entire series, and the final linear head of NCL thus introduces more parameters. Beyond parameter overhead, NCL also exhibits the fastest training times across nearly all TSFMs. For MOMENT, NCL completes training in approximately 2.59 seconds, while LoRA requires 2.71 seconds and CoRA needs 3.32 seconds. Similar patterns are observed for other TSFMs including Chronos, ROSE, and SEMPO. The efficiency advantage makes NCL particularly attractive for practical deployment scenarios where frequent model updates are required.

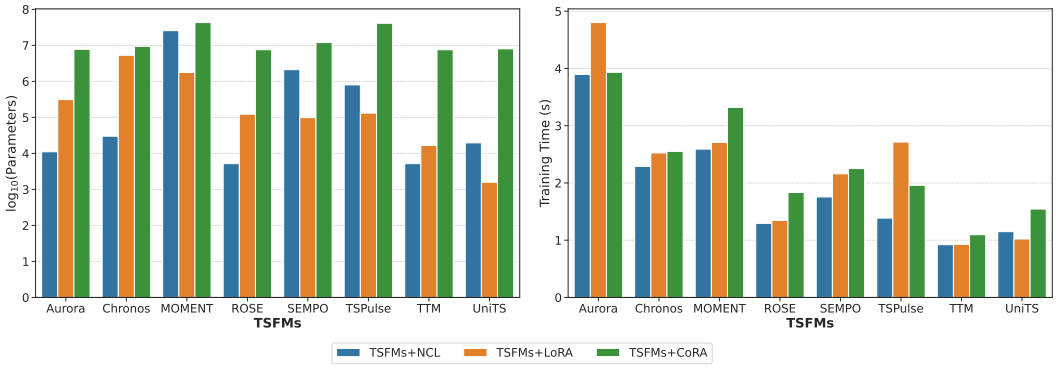


Figure 5: Efficiency comparison of NCL and other TSFM adaptation methods w.r.t. different TSFMs.

H Comparison with Deep TSAD and One-Liner Methods

To further demonstrate the effectiveness of NCL, we compare it against advanced deep TSAD methods (trained from scratch on each dataset) and simple one-liner baselines (requiring no training). Table 4 presents comprehensive precision-recall-based metrics including the Area Under the Precision-Recall Curve (A-P), the Range Area Under the Precision-Recall Curve (R-A-P), and the Volume Under the Surface of Precision-Recall (VUS-PR, abbreviated as V-PR) across all ten datasets. Table 5 reports comprehensive receiver-operating-characteristic-based metrics including the Area under the Receiver Operating Characteristics Curve (A-R), the Range Area under the Receiver Operating Characteristics Curve (R-A-R), and the Volume Under the Surface of Receiver Operating Characteristic (VUS-ROC, abbreviated as V-ROC) across all ten datasets.

Comparison with Deep TSAD Methods. NCL consistently outperforms sophisticated deep TSAD methods such as CroAD, ConAD, TsNet, and Dlinear. Specifically, UniTS+NCL achieves the best

Table 4: PR comparison with deep TSAD methods and one-liner baselines. The best ones are **boldfaced** and the second best are underlined.

Dataset	Metric	CroAD	ConAD	TsNet	Dlinear	LOF	Last-1	Last-3	Var-100	TMM	+NCL	UniTS	+NCL
IOPS	A-P	0.3019	0.0226	0.1486	0.2188	0.0574	0.2147	0.2320	0.2331	0.2093	**0.3393**	0.1810	0.3133
	R-A-P	0.2855	0.0285	0.1447	0.2261	0.0617	0.1764	0.2334	0.1715	0.1886	**0.3349**	0.1524	0.1874
	V-PR	0.2827	0.0285	0.1437	0.2231	0.0634	0.1763	0.2282	0.1722	0.1877	**0.3338**	0.1530	0.1861
UCR	A-P	0.0269	0.0089	0.0404	0.0452	0.0586	0.0727	0.0530	**0.0777**	0.0446	0.0443	0.0735	0.0766
	R-A-P	0.0268	0.0198	0.0433	0.0430	0.0587	0.0635	0.0412	**0.0846**	0.0348	0.0356	0.0827	0.0843
	V-PR	0.0272	0.0198	0.0481	0.0468	0.0586	0.0654	0.0457	0.0817	0.0400	0.0410	0.0808	**0.0832**
NAB	A-P	0.2775	0.1330	0.2486	0.2402	0.1552	0.1655	0.1730	0.4235	0.2054	0.2684	0.3890	**0.4398**
	R-A-P	0.3623	0.2376	0.3417	0.3246	0.2795	0.2570	0.2629	0.4929	0.2956	0.3402	0.4528	**0.4933**
	V-PR	0.3626	0.2356	0.3392	0.3250	0.2801	0.2565	0.2624	0.4960	0.2894	0.3368	0.4533	**0.4962**
MSL	A-P	0.3856	0.0697	0.3542	0.2923	0.1310	0.1341	0.1707	**0.4831**	0.3243	0.4374	0.4317	0.4511
	R-A-P	0.4522	0.1267	0.4312	0.3468	0.2761	0.2603	0.2649	**0.5769**	0.3665	0.4884	0.5169	0.5410
	V-PR	0.4464	0.1264	0.4224	0.3424	0.2758	0.2594	0.2646	0.5591	0.3653	0.4849	0.5033	**0.5657**
SMAP	A-P	0.3452	0.0299	0.4350	0.2801	0.0950	0.0928	0.1332	**0.4681**	0.2411	0.3411	0.4669	0.4617
	R-A-P	0.3373	0.0582	0.4853	0.2698	0.2887	0.1992	0.2102	**0.5457**	0.2733	0.3722	0.5385	0.5376
	V-PR	0.3366	0.0581	0.4747	0.2693	0.2905	0.2006	0.2122	0.5326	0.2736	0.3686	0.5263	**0.5345**
PSM	A-P	0.4044	0.3202	0.3551	0.3481	0.3132	0.3320	0.3437	0.4387	0.3618	0.5116	0.4183	**0.5244**
	R-A-P	0.4019	0.3166	0.3591	0.3500	0.3136	0.3316	0.3426	0.4140	0.3611	**0.5014**	0.4191	0.4747
	V-PR	0.4029	0.3166	0.3600	0.3497	0.3139	0.3316	0.3430	0.4159	0.3611	**0.5039**	0.4203	0.4735
SMD	A-P	0.1612	0.0520	0.1712	0.1307	0.0717	0.0946	0.0938	**0.2122**	0.1134	0.1601	0.1804	0.1898
	R-A-P	0.1602	0.0558	**0.1689**	0.1351	0.0737	0.0827	0.0892	0.1237	0.1111	0.1593	0.1291	0.1338
	V-PR	0.1494	0.0560	**0.1687**	0.1349	0.0737	0.0828	0.0891	0.1242	0.1109	0.1591	0.1296	0.1343
SWAT	A-P	0.1614	0.1265	0.1104	0.1967	0.1303	0.0895	0.0828	0.1064	0.0837	0.6787	0.0862	**0.7237**
	R-A-P	0.2075	0.1534	0.1861	0.2215	0.1587	0.1199	0.1140	0.1690	0.1211	**0.6363**	0.1258	0.5750
	V-PR	0.2011	0.1522	0.1758	0.2185	0.1572	0.1178	0.1115	0.1591	0.1185	**0.6099**	0.1215	0.5346
PUMP	A-P	0.1514	0.1223	0.1191	0.1223	0.1108	0.1372	0.1309	0.1068	0.1228	0.1971	0.1389	**0.2046**
	R-A-P	0.1981	0.1645	0.2217	0.1776	0.1555	0.1879	0.1823	0.1652	0.2107	0.2322	0.2415	**0.2525**
	V-PR	0.1952	0.1636	0.2156	0.1771	0.1564	0.1882	0.1821	0.1681	0.2082	0.2297	0.2370	**0.2534**
NYC	A-P	0.0420	0.0179	0.0244	0.0325	0.0228	0.0216	0.0225	0.1513	0.0359	0.0547	0.1447	**0.2093**
	R-A-P	0.0650	0.0293	0.0489	0.0482	0.0857	0.0595	0.0647	**0.2614**	0.0652	0.0860	0.1496	0.1846
	V-PR	0.0622	0.0289	0.0483	0.0478	0.0755	0.0548	0.0577	**0.2423**	0.0619	0.0806	0.1495	0.1850
Average Rank		5.13	11.23	6.27	6.93	9.07	8.93	8.63	4.00	7.47	3.77	4.40	**2.13**

average rank of **2.13** across all PR metrics and datasets, significantly better than CroAD (5.13) and TsNet (6.27). The key advantage of NCL over deep TSAD methods is its generalization capability. Conventional deep TSAD methods must be trained from scratch for each new dataset, while NCL requires only a few normal samples to adapt pre-trained TSFMs. This makes NCL significantly more practical for real-world scenarios where training data is scarce or expensive to obtain.

Comparison with One-Liner Baselines. Interestingly, simple moving-window variance baselines (Var-100) achieve competitive performance with an average rank of 4.00, even outperforming sophisticated deep learning methods. For instance, on UCR and MSL datasets, Var-100 achieves the best A-P scores. This phenomenon has been noted in recent studies (Liu & Paparrizos, 2024; Zhu et al., 2026), suggesting that many existing methods fail to substantially improve over simple statistical baselines. However, NCL-enhanced TSFMs (TMM+NCL and UniTS+NCL with average ranks of 3.77 and 2.13 in PR metrics, and average ranks of 3.47 and 2.73 in ROC metrics) consistently surpass these baselines, demonstrating that normality constraint learning successfully addresses the fundamental limitation of TSFMs in anomaly detection.

Significant improvement for TSFMs. As analyzed above, NCL-enhanced TSFMs achieve better performance compared to deep TSAD and one-liner methods. More intuitively, in terms of the PR metrics, the average rank of the original TMM was only 7.47, significantly lower than 5.13 for CroAD and 4.00 for Var-100. After adding NCL, the rank improves to 3.77. It is the gains brought by NCL that enable TSFMs to achieve the best ranks compared to the baselines. *e.g.*, 4.40 $\rightarrow$ 2.13 (UniTS on PR metrics), 7.23 $\rightarrow$ 3.47 (TMM on ROC metrics), 4.93 $\rightarrow$ 2.73 (UniTS on ROC metrics). The consistent improvement of NCL across diverse metrics and datasets validates that the proposed normality constraint effectively steers frozen TSFMs toward enhanced anomaly discrimination.

I Comparison with TSAD-Specific Foundation Models

We further compare NCL against state-of-the-art TSAD-specific foundation models DADA (Shentu et al., 2025), TimeRCD (Lan et al., 2026), and TimeRadar (He et al., 2026) in both zero-shot

Table 5: ROC comparison with deep TSAD methods and one-liner baselines. The best ones are **boldfaced** and the second best are underlined.

Dataset	Metric	CroAD	ConAD	TsNet	Dlinear	LOF	Last-1	Last-3	Var-100	TMM	+NCL	UniTS	+NCL
IOPS	A-R	0.7448	0.4979	0.6825	0.6935	0.5474	0.6084	0.6393	0.7727	0.6868	**0.8292**	0.7446	0.8155
	R-A-R	0.7589	0.5396	0.7291	0.7435	0.5585	0.6431	0.6891	0.7417	0.7158	**0.8437**	0.7388	0.7453
	V-ROC	0.7582	0.5335	0.7237	0.7390	0.5534	0.6400	0.6835	0.7426	0.7117	**0.8426**	0.7375	0.7439
UCR	A-R	0.5251	0.5008	0.5262	0.5753	0.5827	0.5291	0.5314	0.5387	0.5484	**0.5851**	0.5506	0.5566
	R-A-R	0.6025	0.6078	0.6030	0.6633	0.6424	0.6177	0.6245	0.5941	0.6379	**0.6442**	0.6055	0.6116
	V-ROC	0.5984	0.5968	0.5960	0.6356	0.6371	0.6112	0.6179	0.5915	0.6295	**0.6403**	0.6021	0.6083
NAB	A-R	0.6623	0.5057	0.6235	0.6070	0.5200	0.5251	0.5297	0.7034	0.5530	0.5828	0.6841	**0.7097**
	R-A-R	0.7372	0.6372	0.7240	0.7051	0.5998	0.6332	0.6393	0.7556	0.6496	0.6598	0.7458	**0.7625**
	V-ROC	0.7338	0.6260	0.7148	0.6983	0.5958	0.6259	0.6317	0.7553	0.6417	0.6550	0.7434	**0.7613**
MSL	A-R	0.7532	0.5017	0.7559	0.6975	0.5451	0.5829	0.5964	**0.8432**	0.6386	0.7326	0.8067	0.8216
	R-A-R	0.8143	0.6212	0.8325	0.7786	0.6150	0.6396	0.6515	**0.8860**	0.7195	0.7737	0.8394	0.8560
	V-ROC	0.8077	0.6093	0.8236	0.7699	0.6076	0.6334	0.6451	**0.8677**	0.7091	0.7677	0.8333	0.8487
SMAP	A-R	0.6618	0.5004	**0.7914**	0.6671	0.5232	0.5193	0.5305	0.7205	0.6070	0.6333	0.7654	0.7603
	R-A-R	0.7314	0.6236	**0.8450**	0.7344	0.5486	0.5622	0.5812	0.7839	0.6938	0.6962	0.8042	0.8036
	V-ROC	0.7255	0.6125	**0.8379**	0.7286	0.5475	0.5586	0.5771	0.7796	0.6853	0.6911	0.8000	0.7985
PSM	A-R	0.6053	0.5752	0.5585	0.5557	0.5290	0.5414	0.5546	0.6317	0.5697	0.6904	0.6387	**0.7100**
	R-A-R	0.5979	0.4726	0.5610	0.5562	0.5258	0.5412	0.5514	0.6012	0.5691	**0.6648**	0.6174	0.6402
	V-ROC	0.5983	0.4695	0.5614	0.5525	0.5221	0.5390	0.5501	0.6025	0.5665	**0.6677**	0.6160	0.6381
SMD	A-R	**0.7708**	0.5773	0.7668	0.7028	0.5696	0.6541	0.6712	0.7573	0.6831	0.7293	0.7590	0.7697
	R-A-R	**0.7927**	0.5780	0.7855	0.7256	0.5957	0.6684	0.6881	0.6295	0.6987	0.7508	0.6875	0.6696
	V-ROC	**0.7905**	0.5779	0.7837	0.7242	0.5953	0.6678	0.6867	0.6294	0.6972	0.7489	0.6854	0.6681
SWAT	A-R	0.6280	0.5380	0.2933	0.6142	0.5198	0.3530	0.2785	0.2358	0.2506	0.8245	0.2339	**0.8274**
	R-A-R	0.6818	0.5833	0.4064	0.6563	0.5688	0.4345	0.3733	0.3456	0.3625	**0.8127**	0.3523	0.7533
	V-ROC	0.6728	0.5797	0.3967	0.6518	0.5649	0.4277	0.3654	0.3325	0.3537	**0.7933**	0.3417	0.7148
PUMP	A-R	0.6669	0.5968	0.4481	0.5702	0.5267	0.6082	0.6045	0.5325	0.5245	0.7904	0.5598	**0.7914**
	R-A-R	0.7335	0.6554	0.6103	0.6574	0.6076	0.6790	0.6739	0.6386	0.6660	**0.7929**	0.6904	0.7904
	V-ROC	0.7273	0.6483	0.5999	0.6517	0.6028	0.6748	0.6692	0.6356	0.6546	**0.7912**	0.6828	0.7873
NYC	A-R	0.6663	0.4055	0.5379	0.6383	0.5157	0.4749	0.4924	0.7374	0.6249	0.6824	0.8186	**0.8620**
	R-A-R	0.7293	0.5039	0.6679	0.6835	0.6834	0.6193	0.6333	**0.8904**	0.7546	0.8142	0.8354	0.8513
	V-ROC	0.7145	0.4888	0.6559	0.6767	0.6500	0.6021	0.6178	**0.8634**	0.7332	0.7846	0.8264	0.8434
Average Rank		4.57	10.07	6.53	5.73	9.60	9.10	8.37	5.67	7.23	3.47	4.93	**2.73**

Table 6: VUS-PR comparison with TSAD-specific foundation models. The best ones are **boldfaced** and the second best are underlined.

Model	**Dataset**									
	IOPS	UCR	NAB	MSL	SMAP	PSM	SMD	SWAT	PUMP	NYC
$\text{DADA}_{\text{zero}}$	0.1837	0.0511	0.4574	0.4395	0.4165	0.4732	0.1239	0.5988	0.2097	0.0553
$\text{DADA}_{\text{five}}$	0.1453	0.0783	0.4692	0.4671	0.3656	0.4549	0.1197	0.4262	**0.2541**	**0.3744**
$\text{TimeRCD}_{\text{zero}}$	0.0691	0.0819	0.3410	0.3328	0.3324	0.3126	0.0691	0.5721	0.1960	0.0690
$\text{TimeRCD}_{\text{five}}$	0.0584	0.0786	0.3033	0.2996	0.3056	0.3117	0.0670	0.5356	0.1916	0.0650
$\text{TimeRadar}_{\text{zero}}$	0.1111	0.0467	0.3221	0.4617	0.3207	0.4231	0.0968	0.5802	0.2263	0.0985
$\text{TimeRadar}_{\text{five}}$	0.1048	0.0483	0.4424	0.4697	0.3869	0.4259	0.1001	0.5769	0.2211	0.1012
$\text{Time-MoE}_{\text{zero}}$	0.2163	0.0568	0.2875	0.3618	0.2182	0.4017	0.0930	0.6238	0.1986	0.0392
+NCL	**0.3358**	0.0780	0.3712	0.4342	0.3324	0.4575	0.1452	**0.6384**	0.2329	0.0677
$\text{TimesFM}_{\text{zero}}$	0.1902	0.0543	0.2804	0.3928	0.3366	0.3497	0.1123	0.1132	0.1792	0.0444
+NCL	0.3244	0.0600	0.3728	0.4360	0.3862	0.4794	0.1498	0.5117	0.2380	0.0483
TMM_{zero}	0.1877	0.0400	0.2894	0.3653	0.2736	0.3611	0.1109	0.1185	0.2082	0.0619
+NCL	0.3338	0.0410	0.3368	0.4849	0.3686	**0.5039**	**0.1591**	0.6099	0.2297	0.0806
$\text{UniTS}_{\text{zero}}$	0.1530	0.0808	0.4533	0.5033	0.5263	0.4203	0.1296	0.1215	0.2370	0.1495
+NCL	0.1861	**0.0832**	**0.4962**	**0.5657**	**0.5345**	0.4735	0.1343	0.5346	0.2534	0.1850

and few-shot settings. We apply NCL to four representative general-purpose TSFMs: Time-MoE, TimesFM, TMM, and UniTS, and the results are presented in Table 6. NCL demonstrates substantial improvements when integrated with general-purpose TSFMs, enabling them to achieve competitive or superior performance compared to TSAD-specific foundation models. Specifically, UniTS+NCL

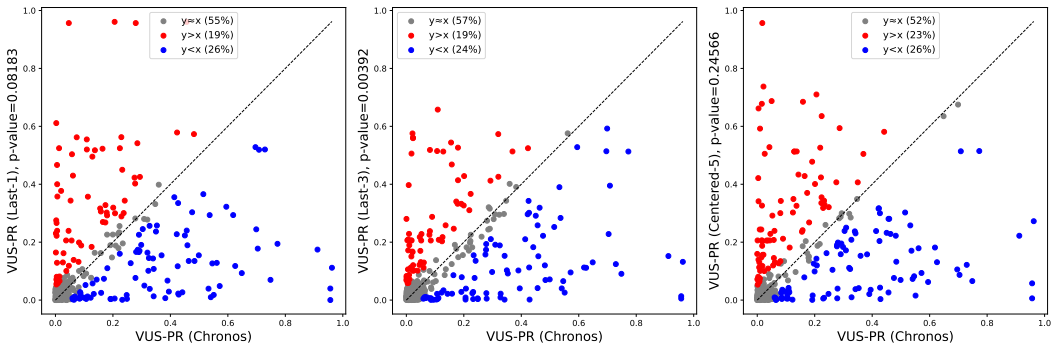
achieves the best performance on 4 out of 10 datasets, while TMM+NCL excels on 2 datasets. Both Time-MoE+NCL and TimesFM+NCL are competitive on datasets such as PSM and SMD. This indicates that NCL effectively transforms them into powerful anomaly detectors.

While DADA, TimeRCD, and TimeRadar are specifically engineered for anomaly detection, they show inconsistent performance across datasets. $\text{DADA}_{\text{five}}$ performs well on PUMP (0.2541) and NYC (0.3744), but struggles on other datasets like IOPS (0.1453) and SWAT (0.4262). $\text{TimeRCD}_{\text{zero}}$ achieves notable performance on SWAT (0.5721) but falls short on most other benchmarks. $\text{TimeRadar}_{\text{zero}}$ achieves competitive performance on certain datasets such as SWAT (0.5802), yet underperforms on other datasets. In contrast, NCL-augmented TSFMs demonstrate more robust performance across diverse domains, achieving the higher average performance across all datasets. This suggests that general-purpose TSFMs, when properly adapted through NCL, can leverage their broad pre-trained knowledge more effectively than models trained specifically for anomaly detection from scratch.

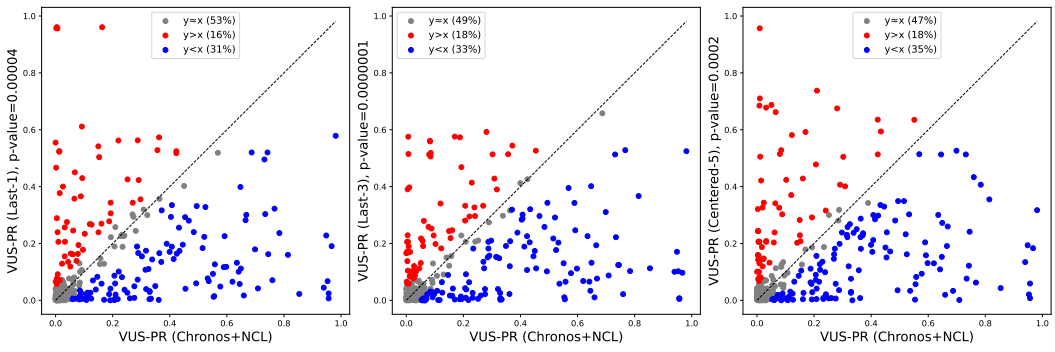
J One-to-One Comparison with One-Liner Baselines

To provide a more granular comparison against simple one-liner baselines, we conduct one-to-one comparisons across all 361 time series from the ten datasets. Figure 6 and Figure 7 visualize the win-loss records of Chronos and MOMENT against three representative baselines, respectively.

Chronos vs. Squared-Difference Baselines. Figure 6a shows that the vanilla Chronos struggles against simple squared-difference baselines (Last-1, Last-3, Centered-5), winning only 26% against Last-1, 24% against Last-3, and 26% against Centered-5. More importantly, the p-value between Chronos and Centered-5 is 0.246, indicating that the statistics are not significant. This confirms the observation in recent studies (Zhu et al., 2026) that foundation models without proper normality con-



(a) Comparison between Chronos and the squared-difference baselines.



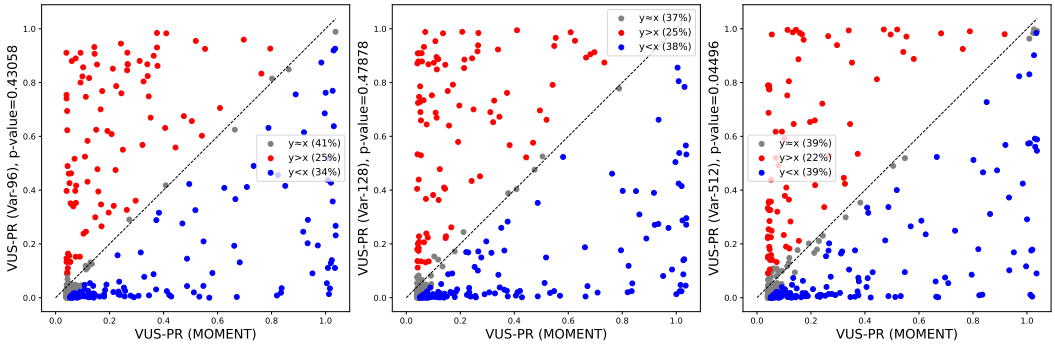
(b) Comparison between Chronos+NCL and the squared-difference baselines.

Figure 6: One-to-one comparison between Chronos (without and with NCL) and three squared-difference baselines across 361 time series from ten TSAD datasets.

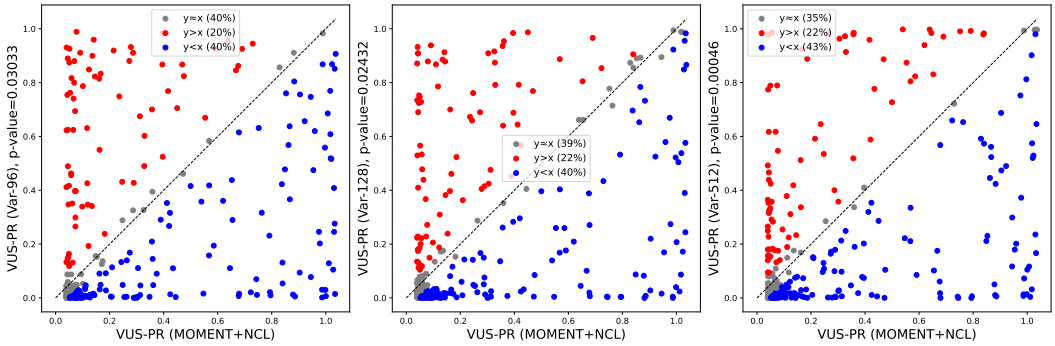
straints often fail to outperform trivial baselines. By contrast, as shown in Figure 6b, Chronos+NCL dramatically improves the win rate, achieving **31%** against Last-1, **33%** against Last-3, and **35%** against Centered-5. In the meantime, the p-value between Chronos+NCL and Centered-5 is 0.0002, indicating a significant difference. This substantial improvement demonstrates NCL's effectiveness in steering the foundation model toward normality-biased predictions that amplify anomalies.

MOMENT vs. Moving-Window Variances. Figure 7a reveals a similar pattern for reconstruction-based TSFMs. Vanilla MOMENT wins only 34% against Var-96, 38% against Var-128, and 39% against Var-512, and the p-value between MOMENT and Var-96 is 0.4306. After applying NCL, as shown in Figure 7b, MOMENT+NCL significantly improves to **40%** against Var-96, **40%** against Var-256, and **43%** against Var-512; and reduces the p-value between MOMENT and Var-96 to 0.0303. The improvements are particularly pronounced against the stronger Var-96 baseline, indicating that NCL effectively captures fine-grained normal patterns that simple variance measures cannot.

The consistent improvements across hundreds of individual time series provide strong statistical evidence for NCL's effectiveness. While NCL substantially improves overall performance, there are time series where simple baselines outperform NCL-enhanced TSFMs. These cases stem from two main aspects: (i) time series with extremely simple periodic patterns where one-liner methods naturally excel; and (ii) time series with non-stationary training samples where the normal subspace construction becomes unstable. Future work could explore stationary processing strategies to address the latter limitation.



(a) Comparison between MOMENT and the moving-window variances.



(b) Comparison between MOMENT+NCL and the moving-window variances.

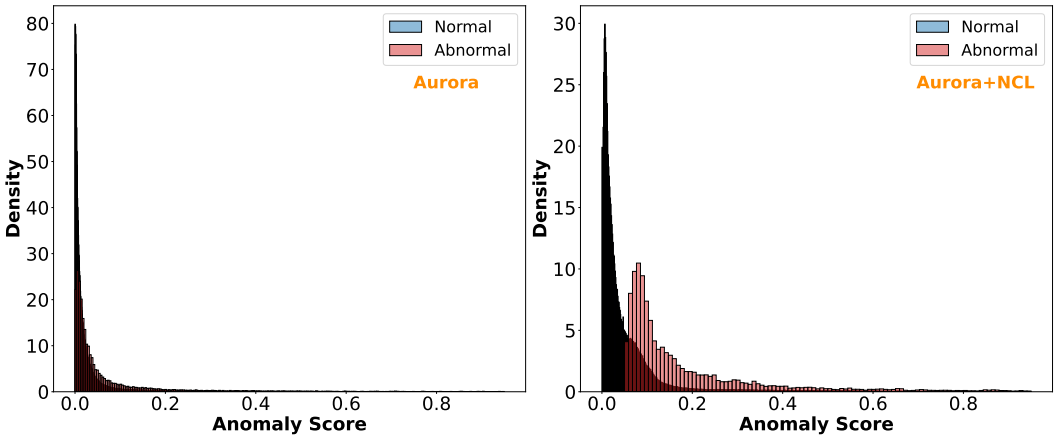
Figure 7: One-to-one comparison between MOMENT (without and with NCL) and three moving-window variances across 361 time series from ten TSAD datasets.

K Distinguishable Distributions of Anomaly Scores

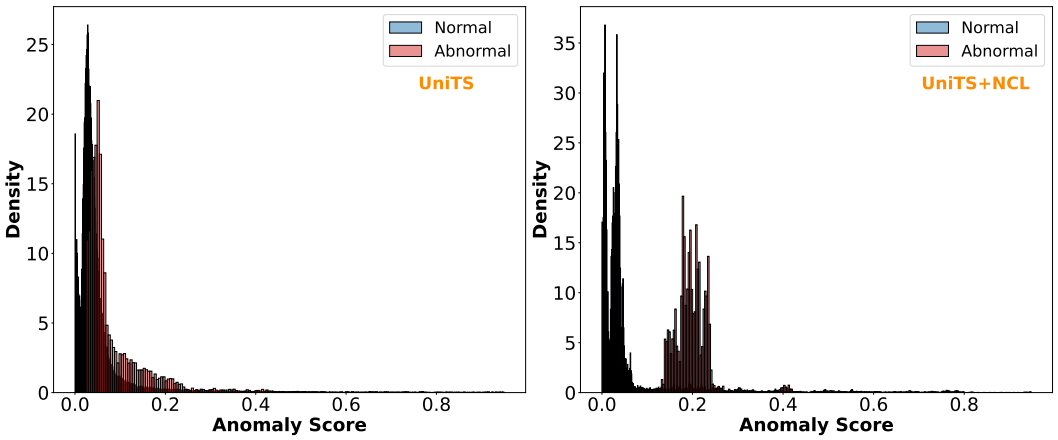
Figure 8 provides additional evidence for NCL's effectiveness by visualizing the anomaly score distributions on different TSFMs and datasets. The results consistently show that NCL successfully widens the gap between normal and abnormal score distributions. As shown in Figure 8a, vanilla

Aurora produces highly overlapping distributions between normal and abnormal observations, with substantial overlap in the 0-0.2 score range. This overlap makes it extremely difficult to set an appropriate threshold for anomaly detection. After applying NCL, the distributions become well-separated: normal observations concentrate around 0-0.1, while anomalous observations shift toward 0.1-0.4 with a longer tail extending to higher scores, creating a clear separation margin. The reduced overlap translates directly to improved detection performance with fewer false positives and false negatives. Figure 8b reveals a similar pattern for reconstruction-based TSFMs. Vanilla UniTS exhibits significant distribution overlap with both normal and abnormal scores concentrated in the 0-0.2 range, making discrimination challenging. With NCL, the normal distribution becomes more compact and concentrated near zero, while the abnormal distribution shifts rightward.

The consistent distributional improvements across different TSFMs (forecasting-based Aurora and reconstruction-based UniTS) and datasets (multivariate SMD and PUMP) demonstrate that NCL's normality constraint is universally effective. Regardless of the TSFM architecture or data characteristics, NCL successfully steers the model toward normality-biased predictions that amplify the normal-abnormal gap.



(a) Anomaly scores for Aurora on the SMD dataset.



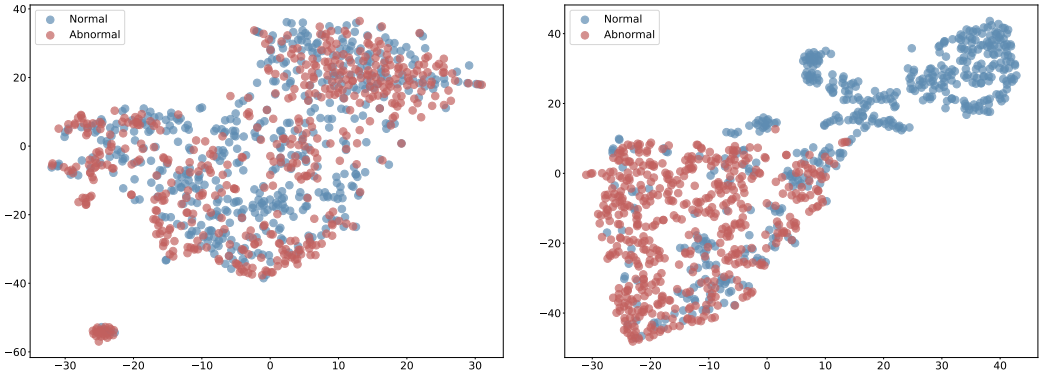
(b) Anomaly scores for UniTS on the PUMP dataset.

Figure 8: Anomaly score distributions for different TSFMs and datasets.

L Visualization of Feature Space

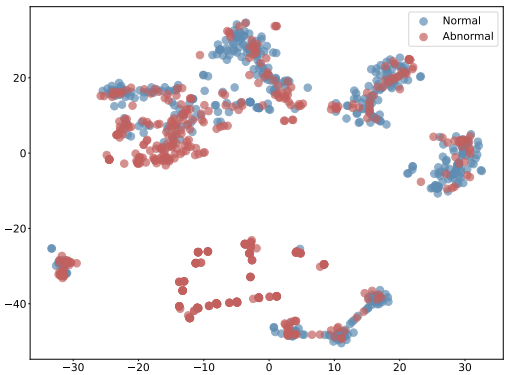
To understand how NCL sharpens the feature normality within the feature space, we utilize t-SNE dimensionality reduction to visualize patch features from vanilla and NCL-augmented TSFMs.

Figure 9 presents the feature space visualization on two representative TSFM-dataset combinations. As shown in Figure 9a, vanilla TMM produces a feature space where normal and abnormal patches are heavily intermingled with no clear boundary. The normal features are scattered across a large region, and many abnormal features fall within the normal region, making discrimination extremely difficult. After applying NCL (as shown in Figure 9b), the feature space undergoes a dramatic transformation. The abnormal features are clearly separated from the normal clusters. Figure 9c and 9d show a similar pattern for reconstruction-based TSFMs. The feature visualizations provide intuitive geometric understanding of how NCL improves anomaly detection. By explicitly constructing an informative normality subspace and constraining features to align with the normality manifolds, NCL transforms the originally broad pattern space into one where normal and abnormal patterns occupy geometrically distinct regions. This spatial separation makes subsequent reconstruction/forecasting error-based anomaly scoring much more discriminative.

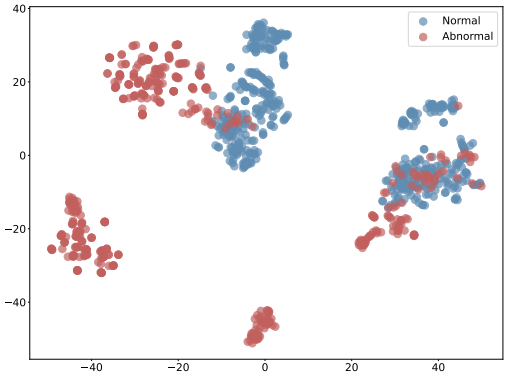


(a) Vanilla TMM on the PUMP dataset.

(b) NCL-augmented TMM on the PUMP dataset.



(c) Vanilla UniTS on the PSM dataset.



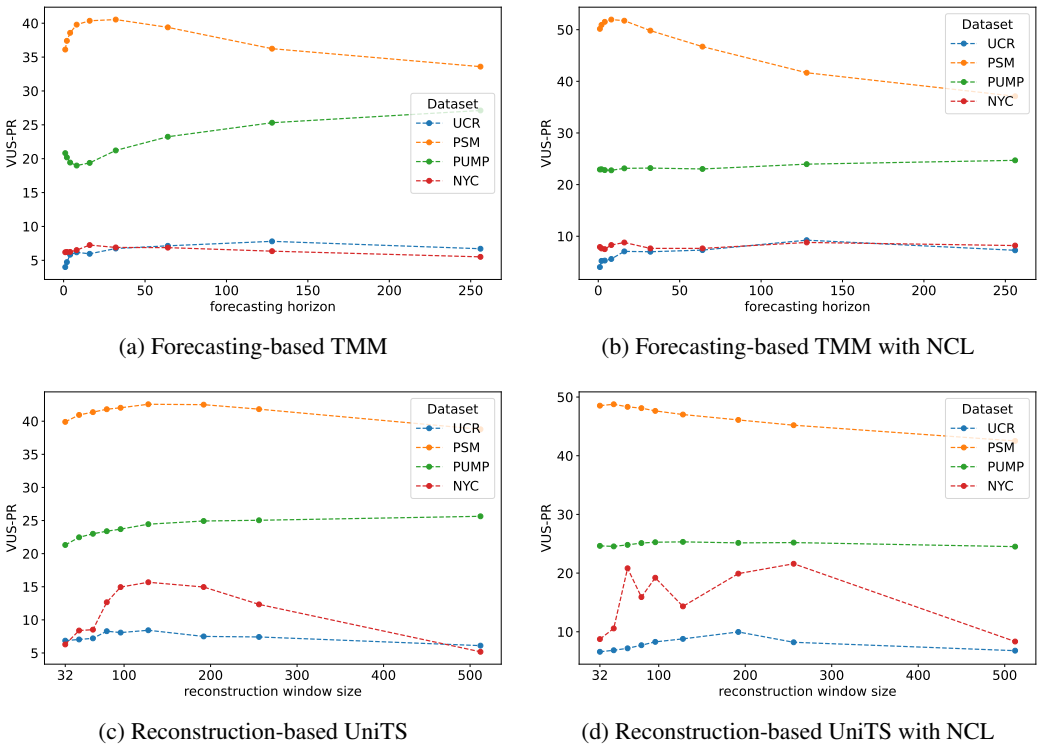
(d) NCL-augmented UniTS on the PSM dataset.

Figure 9: The t-SNE visualization of internal features from vanilla and NCL-augmented TSFMs.

M Horizon and Window Size Analysis for TSFMs

To investigate the impact of forecasting horizons and reconstruction window sizes on anomaly detection performance, we conduct comprehensive experiments on both forecasting-based and reconstruction-based TSFMs. Figure 10 presents the VUS-PR scores across different horizon/window configurations on four representative datasets.

Forecasting Horizon Analysis. For forecasting-based TMM, we vary the forecasting horizon $F \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ while fixing the input window length. As shown in Figure 10a and 10b, compared to vanilla TMM, NCL-enhanced TMM exhibits relatively stable performance across different horizons, and achieves excellent detection results even with small horizons. This robustness stems from NCL's explicit construction of a normal subspace. Even when the forecasting



(a) Forecasting-based TMM

(b) Forecasting-based TMM with NCL

(c) Reconstruction-based UniTS

(d) Reconstruction-based UniTS with NCL

Figure 10: VUS-PR scores with various forecasting horizons and reconstruction window sizes, evaluated on four datasets for different TSFMs without and with using NCL.

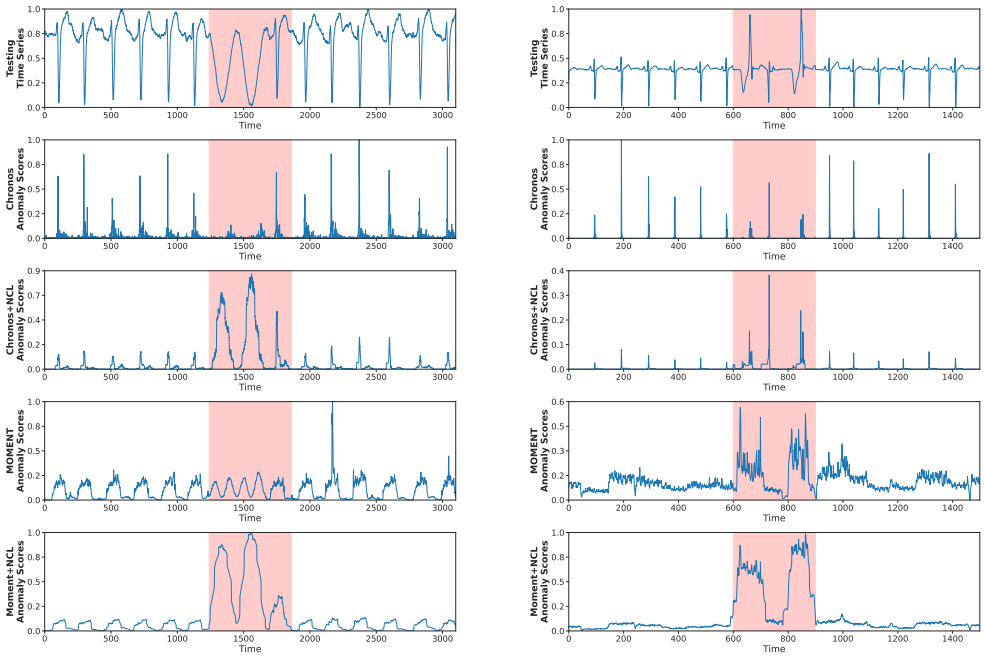
task becomes harder, the contrastive normality constraint ensures that features are consistently aligned with normality manifolds, preventing the model from fitting anomalous patterns equally well.

Reconstruction Window Size Analysis. For reconstruction-based UniTS, we evaluate different window sizes $L \in \{32, 48, 64, 80, 96, 128, 192, 256, 512\}$. Figure 10c and 10d reveal that NCL can stably enhance the detection performance of vanilla UniTS under different window sizes. For instance, on PSM, NCL continuously increases the VUS-PR from around 40 to nearly 50. This validates that NCL's normality transformation operates effectively across different temporal scales. By constructing the normal subspace from patch features rather than raw observations, NCL captures scale-invariant normal patterns that remain discriminative across various window configurations.

The robustness to hyperparameter choices is crucial for practical deployment. Performing extensive grid search to identify optimal horizons or window sizes for each dataset is computationally expensive and may not generalize to new domains. NCL alleviates this burden by providing stable performance across a wide range of configurations, making it easier to deploy TSFMs for anomaly detection in real-world scenarios.

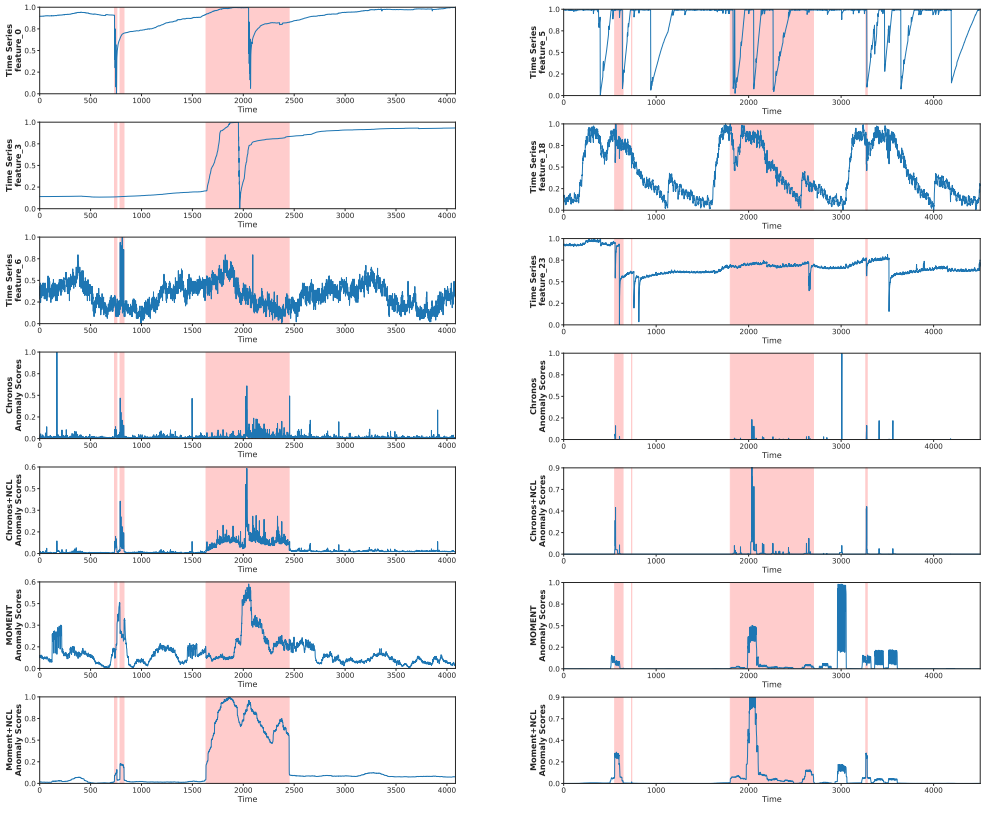
N Visualization of Anomaly Scores

We visualize the anomaly scores and ground truth labels on representative time series segments from both univariate and multivariate datasets. Figure 11 presents original time series, ground truth anomaly intervals, and anomaly scores for different TSFMs. Figure 11a and 11b show the detection results on the UCR time series. Notably, after applying NCL, the anomaly scores exhibit significantly improved alignment with the ground truth. The scores for normal regions are consistently suppressed close to zero, while true anomalies produce sharp, prominent peaks that clearly stand out. For instance, in Figure 11a, the anomalous segment around timesteps 1250-1750 that was previously missed now generates distinctly elevated scores. This improvement validates that NCL's normality constraint effectively amplifies the normal-abnormal discrimination gap by steering features toward the normal subspace and penalizing deviations through contrastive misalignment. NCL-enhanced



(a) DISTORTED1sddb40 from UCR dataset

(b) mit14046longtermecg from UCR dataset



(c) Multivariate segment from PSM dataset

(d) Multivariate segment from SMD dataset

Figure 11: Visualization of anomaly scores in comparison with vanilla and NCL-augmented TSFMs on the univariate and multivariate datasets. Red regions highlight the ground truth anomaly intervals.

TSFMs also produce distinct, well-separated peaks responding to true anomalies in multivariate time series (as shown in Figure 11c and 11d). The reduced false positive rate and improved peak sharpness on actual anomalies demonstrate NCL's effectiveness in multivariate scenarios where cross-channel dependencies add complexity. These visual improvements further confirm that NCL successfully amplifies the normal-abnormal discrimination signal that vanilla TSFMs lack.

O Utility of Few-Shot Normal Samples

A key practical question for NCL is how many normal samples are required to construct an informative normality subspace and learn accurate normality transformations. To investigate this, we evaluate NCL's performance with varying numbers of few-shot normal samples on three representative TSFMs: ROSE (forecasting-based), TMM (forecasting-based), and UniTS (reconstruction-based). Figure 12 presents the VUS-PR performance across ten datasets with sample sizes ranging from 1 to 20.

Remarkably, NCL achieves competitive performance even with extremely limited normal samples. As shown in Figure 12a, ROSE+NCL with only one normal sample already achieves substantial improvements across all datasets. This demonstrates that even one representative normality sample can provide sufficient information to construct a meaningful normal subspace via SVD, which also confirms Theorem 2 in Appendix B.1. This extreme sample efficiency is crucial for scenarios where normal data is scarce or expensive to obtain, such as in industrial anomaly detection where collecting extensive normal operation data may require prolonged monitoring periods.

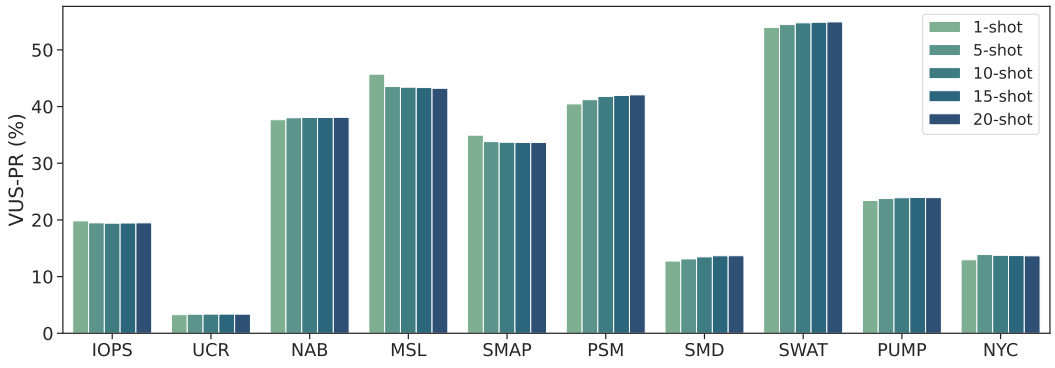
While performance generally improves with more normal samples, the gains exhibit diminishing returns beyond 5-10 samples. This pattern appears consistently across all three TSFMs, suggesting that the top singular vectors capturing the principal normal directions stabilize relatively quickly. Once the dominant axes of normal variation are identified, additional samples provide only marginal refinement. It is worth noting that different datasets exhibit varying sensitivity to sample size. On relatively simple univariate datasets like IOPS and NAB where normal patterns are more homogeneous, 1-3 samples often suffice to capture the essential normal structure. On more complex multivariate datasets like SMD and SWAT with diverse normal modes, 5-10 samples provide better coverage of the normality manifolds. However, even on complex datasets, NCL never requires extensive data, and five samples consistently approach asymptotic performance across all tested scenarios.

Based on these empirical observations, we recommend using five normal samples as the default configuration for NCL, which provides an excellent balance between sample efficiency and detection performance. This configuration achieves near-optimal performance while minimizing data collection requirements. For deployment scenarios with extremely limited access to normal data, even 1-3 samples can yield substantial improvements over vanilla TSFMs. For critical applications where maximal performance is required and data collection is feasible, using 10-20 samples provides marginal additional gains with minimal additional cost.

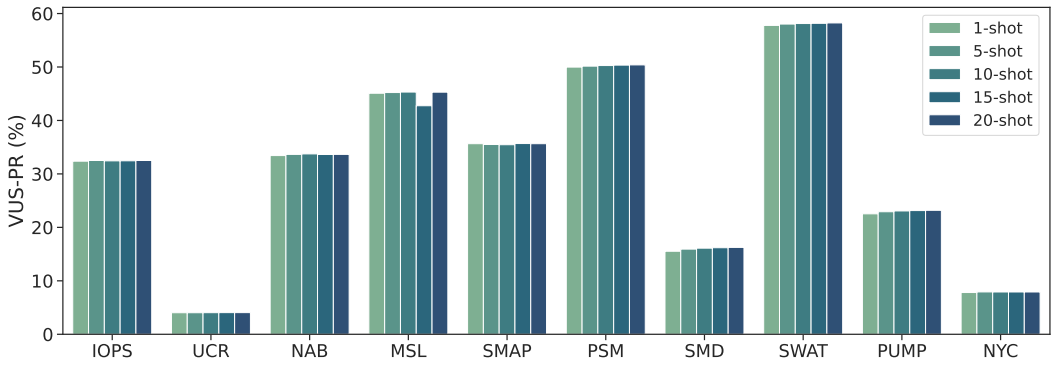
P Sensitivity Analysis for Singular Vectors

The number of top singular vectors k retained in NormSpace directly determines the dimensionality of the normality subspace and the number of normality transformation views produced by PatchNT. To understand how this critical hyperparameter affects performance, we conduct comprehensive sensitivity analysis by varying $k \in \{8, 12, 16, 20, 24\}$ across three representative TSFMs.

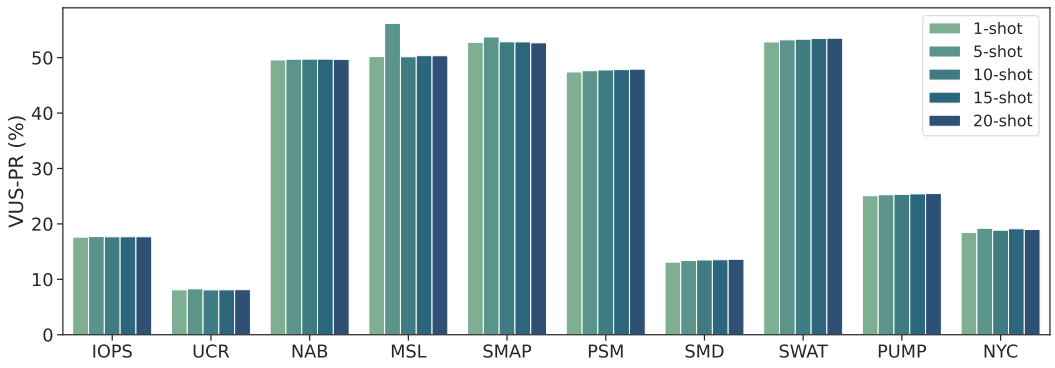
Figure 13 presents the VUS-PR performance changes. This shows that too few singular vectors result in suboptimal performance because the normality subspace becomes overly simplistic and fails to capture the full spectrum of normal variation. For instance, on the NYC dataset, UniTS only achieves VUS-PR of 16.86 using $k = 8$, which is substantially lower than the peak performance. Conversely, too many singular vectors also degrade performance because minor singular vectors corresponding to noise or dataset-specific idiosyncrasies are included, diluting the discriminative power of the principal normal directions. Importantly, different TSFMs on various datasets exhibit slightly different sensitivity patterns, which depend on the complexity of the architecture and the data itself. Simpler univariate datasets like NAB with more homogeneous normal patterns show relatively flat performance curves with less pronounced peaks, suggesting that even moderate values of k capture the essential normal structure. More complex multivariate datasets like SWAT exhibit sharper



(a) ROSE with NCL on ten TSAD datasets.



(b) TMM with NCL on ten TSAD datasets.

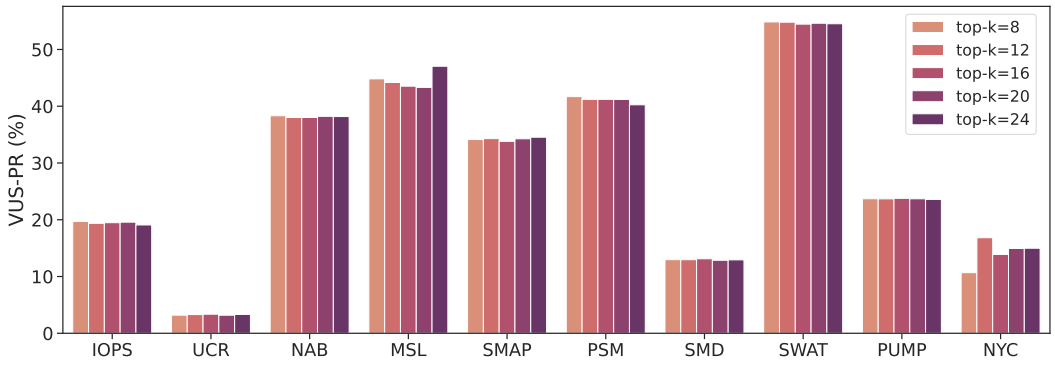


(c) UniTS with NCL on ten TSAD datasets.

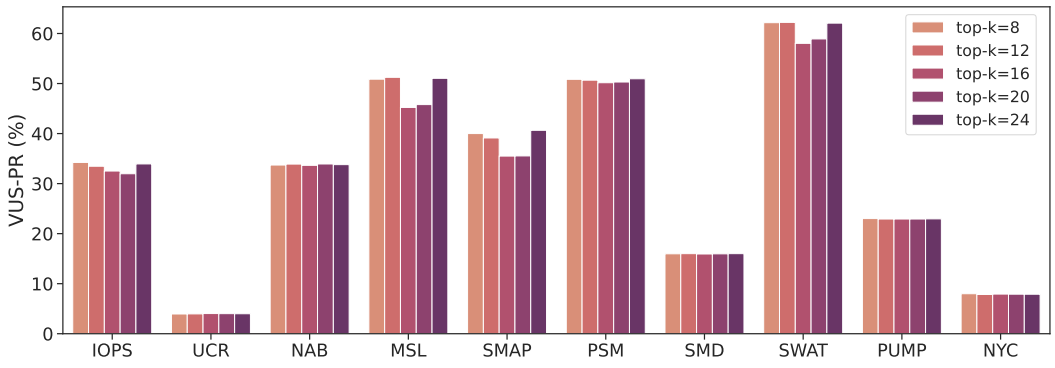
Figure 12: VUS-PR performance of three NCL-augmented TSFMs w.r.t. different number of normal samples.

peaks, indicating that these datasets require a more carefully balanced subspace dimensionality to effectively separate normal and abnormal patterns.

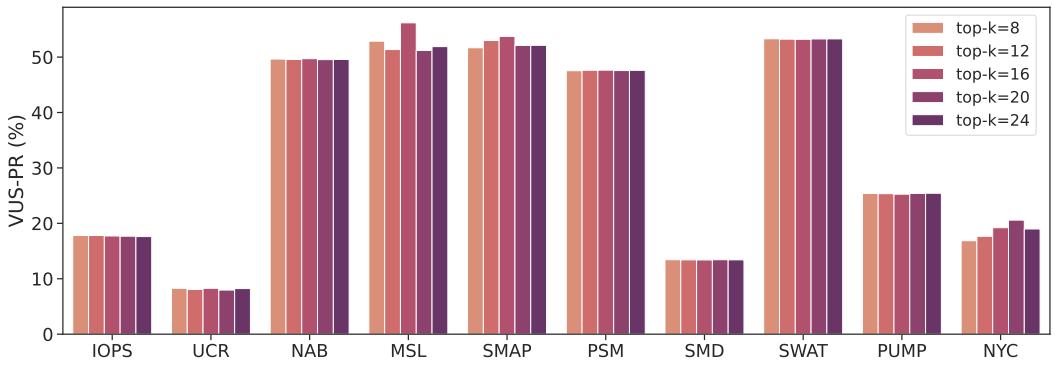
Beyond detection performance, the choice of k also affects computational efficiency. Smaller k values reduce both the number of transformation views computed in PatchNT and the number of cluster centers maintained in ConNC, leading to faster training and inference. The fact that $k = 16$ delivers good performance while remaining computationally modest (16 parallel transformations per patch) represents an ideal trade-off. Higher k would increase the transformation cost while may not improve performance.



(a) ROSE with NCL on ten TSAD datasets.



(b) TMM with NCL on ten TSAD datasets.



(c) UniTS with NCL on ten TSAD datasets.

Figure 13: VUS-PR performance of three NCL-augmented TSFMs w.r.t. different number of singular vectors.

Q Full Experimental Results of Vanilla and NCL-Augmented TSFMs

We report the full experimental results of ten vanilla TSFMs and their NCL-augmented variants on ten TSAD benchmarks in Table 7 and 8, evaluated by both PR-based and ROC-based metrics. Overall, NCL demonstrates strong and consistent effectiveness across a wide range of TSFMs and datasets. In the PR evaluation, NCL improves the performance of nearly all backbones in most cases. Similar trends are also observed in ROC metrics, where NCL substantially boosts the anomaly discrimination capability of the underlying TSFMs. Moreover, the paired t-tests show statistically

significant improvements for almost all model groups, with very small p-values in both PR and ROC results, confirming that the observed improvements are not due to random variation.

To further understand the relationships among TSFMs before and after NCL augmentation, we visualize the pairwise correlation matrices in Figure 14 and 15. In PR metrics, the vanilla TSFMs already exhibit several strong correlations, especially among models with similar forecasting/reconstruction behaviors. However, after applying NCL, the correlations among the augmented TSFMs become even more uniformly high, with many pairwise coefficients approaching 1.0. This suggests that NCL drives different TSFMs toward a more aligned notion of normality, despite their heterogeneous architectures and pretrained representations. In other words, while vanilla TSFMs may capture broad temporal patterns, NCL encourages them to converge to a shared normality-centered representation space. Meanwhile, the correlations between vanilla TSFMs and NCL-augmented TSFMs are noticeably lower than the correlations within the vanilla group, indicating that NCL-augmented TSFMs are recalibrated through explicit normality learning to enhance the anomaly detection performance. A similar but more revealing trend is observed in ROC correlations.

Together, the full experimental results and the pairwise correlation analyses provide strong evidence for the effectiveness and universality of NCL. Across diverse TSFMs, datasets, and evaluation metrics, NCL consistently improves anomaly detection performance. This demonstrates that NCL is not merely a model-specific enhancement, but a general normality-learning framework that can reliably constrain pre-trained TSFM features toward compact and discriminative normality manifolds.

Table 7: PR results of ten vanilla and NCL-augmented TSFMs that use forecasting/reconstruction errors to detect anomalies on ten TSAD datasets. Meanwhile, we perform the paired t-tests, and p-**values** are reported.

Dataset	Metric	Aurora	+NCL	ROSE	+NCL	Chronos	+NCL	TMM	+NCL	TimeMoE	+NCL
IOPS	A-P	0.1433	$0.2646_{\pm 0.0148}$	0.1712	$0.2396_{\pm 0.0118}$	0.2649	$0.3654_{\pm 0.0019}$	0.2093	$0.3393_{\pm 0.0144}$	0.2535	$0.3367_{\pm 0.0094}$
	R-A-P	0.1331	$0.2540_{\pm 0.0247}$	0.1300	$0.1935_{\pm 0.0137}$	0.2390	$0.3619_{\pm 0.0017}$	0.1886	$0.3349_{\pm 0.0130}$	0.2171	$0.3360_{\pm 0.0074}$
	V-PR	0.1333	$0.2528_{\pm 0.0244}$	0.1307	$0.1936_{\pm 0.0138}$	0.2387	$0.3614_{\pm 0.0017}$	0.1877	$0.3338_{\pm 0.0130}$	0.2163	$0.3358_{\pm 0.0077}$
UCR	A-P	0.0162	$0.0214_{\pm 0.0012}$	0.0200	$0.0332_{\pm 0.0013}$	0.0636	$0.0754_{\pm 0.0041}$	0.0446	$0.0443_{\pm 0.0008}$	0.0661	$0.0830_{\pm 0.0082}$
	R-A-P	0.0215	$0.0246_{\pm 0.0025}$	0.0238	$0.0323_{\pm 0.0008}$	0.0531	$0.0665_{\pm 0.0032}$	0.0348	$0.0356_{\pm 0.0007}$	0.0512	$0.0740_{\pm 0.0074}$
	V-PR	0.0216	$0.0247_{\pm 0.0025}$	0.0242	$0.0329_{\pm 0.0007}$	0.0566	$0.0691_{\pm 0.0032}$	0.0400	$0.0410_{\pm 0.0007}$	0.0568	$0.0780_{\pm 0.0069}$
NAB	A-P	0.2070	$0.2631_{\pm 0.0018}$	0.2356	$0.2845_{\pm 0.0054}$	0.1896	$0.2992_{\pm 0.0014}$	0.2054	$0.2684_{\pm 0.0006}$	0.1998	$0.2980_{\pm 0.0061}$
	R-A-P	0.3066	$0.3428_{\pm 0.0051}$	0.3451	$0.3840_{\pm 0.0086}$	0.2775	$0.3764_{\pm 0.0013}$	0.2956	$0.3402_{\pm 0.0015}$	0.2875	$0.3714_{\pm 0.0067}$
	V-PR	0.3004	$0.3401_{\pm 0.0040}$	0.3365	$0.3754_{\pm 0.0062}$	0.2766	$0.3756_{\pm 0.0012}$	0.2894	$0.3368_{\pm 0.0012}$	0.2875	$0.3712_{\pm 0.0068}$
MSL	A-P	0.2551	$0.3275_{\pm 0.0267}$	0.3056	$0.3968_{\pm 0.0248}$	0.3161	$0.3684_{\pm 0.0046}$	0.3243	$0.4374_{\pm 0.0254}$	0.2881	$0.3577_{\pm 0.0042}$
	R-A-P	0.2973	$0.3693_{\pm 0.0192}$	0.3616	$0.4506_{\pm 0.0242}$	0.4021	$0.4401_{\pm 0.0038}$	0.3665	$0.4884_{\pm 0.0311}$	0.3690	$0.4420_{\pm 0.0025}$
	V-PR	0.2982	$0.3681_{\pm 0.0192}$	0.3604	$0.4488_{\pm 0.0214}$	0.3943	$0.4361_{\pm 0.0031}$	0.3653	$0.4849_{\pm 0.0304}$	0.3618	$0.4342_{\pm 0.0012}$
SMAP	A-P	0.1537	$0.2644_{\pm 0.0113}$	0.1896	$0.2980_{\pm 0.0150}$	0.2363	$0.3480_{\pm 0.0075}$	0.2411	$0.3411_{\pm 0.0306}$	0.1907	$0.3248_{\pm 0.0147}$
	R-A-P	0.1898	$0.2833_{\pm 0.0119}$	0.2430	$0.3512_{\pm 0.0165}$	0.3189	$0.3779_{\pm 0.0185}$	0.2733	$0.3722_{\pm 0.0169}$	0.2182	$0.3318_{\pm 0.0126}$
	V-PR	0.1889	$0.2823_{\pm 0.0123}$	0.2417	$0.3484_{\pm 0.0162}$	0.3106	$0.3741_{\pm 0.0184}$	0.2736	$0.3686_{\pm 0.0176}$	0.2182	$0.3324_{\pm 0.0118}$
PSM	A-P	0.3590	$0.4772_{\pm 0.0039}$	0.3703	$0.4181_{\pm 0.0136}$	0.3483	$0.4998_{\pm 0.0022}$	0.3618	$0.5116_{\pm 0.0045}$	0.4097	$0.4810_{\pm 0.0051}$
	R-A-P	0.3594	$0.4737_{\pm 0.0051}$	0.3668	$0.4096_{\pm 0.0143}$	0.3473	$0.4900_{\pm 0.0020}$	0.3611	$0.5014_{\pm 0.0050}$	0.4014	$0.4565_{\pm 0.0031}$
	V-PR	0.3592	$0.4743_{\pm 0.0050}$	0.3672	$0.4107_{\pm 0.0145}$	0.3474	$0.4926_{\pm 0.0021}$	0.3611	$0.5039_{\pm 0.0051}$	0.4017	$0.4575_{\pm 0.0030}$
SMD	A-P	0.1074	$0.1506_{\pm 0.0000}$	0.1012	$0.1395_{\pm 0.0073}$	0.1003	$0.1442_{\pm 0.0001}$	0.1134	$0.1601_{\pm 0.0011}$	0.0959	$0.1496_{\pm 0.0004}$
	R-A-P	0.1071	$0.1483_{\pm 0.0004}$	0.1024	$0.1330_{\pm 0.0077}$	0.0980	$0.1463_{\pm 0.0001}$	0.1111	$0.1593_{\pm 0.0012}$	0.0930	$0.1452_{\pm 0.0004}$
	V-PR	0.1070	$0.1483_{\pm 0.0004}$	0.1025	$0.1331_{\pm 0.0077}$	0.0980	$0.1462_{\pm 0.0001}$	0.1109	$0.1591_{\pm 0.0012}$	0.0930	$0.1452_{\pm 0.0004}$
SWAT	A-P	0.0879	$0.6872_{\pm 0.0002}$	0.1079	$0.5041_{\pm 0.0039}$	0.7117	$0.7117_{\pm 0.0000}$	0.0837	$0.6787_{\pm 0.0316}$	0.6445	$0.7185_{\pm 0.0003}$
	R-A-P	0.1290	$0.6641_{\pm 0.0079}$	0.1708	$0.5673_{\pm 0.0015}$	0.6209	$0.6210_{\pm 0.0001}$	0.1211	$0.6363_{\pm 0.0291}$	0.6425	$0.6314_{\pm 0.0015}$
	V-PR	0.1265	$0.6484_{\pm 0.0102}$	0.1638	$0.5437_{\pm 0.0012}$	0.5979	$0.6281_{\pm 0.0002}$	0.1185	$0.6099_{\pm 0.0276}$	0.6238	$0.6384_{\pm 0.0015}$
PUMP	A-P	0.1203	$0.1881_{\pm 0.0006}$	0.1796	$0.1795_{\pm 0.0005}$	0.1870	$0.2094_{\pm 0.0005}$	0.1228	$0.1971_{\pm 0.0024}$	0.1700	$0.2077_{\pm 0.0008}$
	R-A-P	0.2065	$0.2276_{\pm 0.0025}$	0.2412	$0.2447_{\pm 0.0037}$	0.2092	$0.2360_{\pm 0.0005}$	0.2107	$0.2322_{\pm 0.0006}$	0.1996	$0.2351_{\pm 0.0006}$
	V-PR	0.2027	$0.2247_{\pm 0.0022}$	0.2360	$0.2392_{\pm 0.0022}$	0.2097	$0.2347_{\pm 0.0005}$	0.2082	$0.2297_{\pm 0.0007}$	0.1986	$0.2329_{\pm 0.0006}$
NYC	A-P	0.0200	$0.0298_{\pm 0.0008}$	0.0710	$0.0901_{\pm 0.0217}$	0.0285	$0.0422_{\pm 0.0010}$	0.0359	$0.0547_{\pm 0.0009}$	0.0277	$0.0478_{\pm 0.0064}$
	R-A-P	0.0315	$0.0407_{\pm 0.0023}$	0.1034	$0.1479_{\pm 0.0383}$	0.0498	$0.0635_{\pm 0.0002}$	0.0652	$0.0860_{\pm 0.0015}$	0.0399	$0.0685_{\pm 0.0066}$
	V-PR	0.0341	$0.0408_{\pm 0.0021}$	0.0969	$0.1431_{\pm 0.0148}$	0.0482	$0.0604_{\pm 0.0003}$	0.0619	$0.0806_{\pm 0.0012}$	0.0392	$0.0677_{\pm 0.0071}$
p-**values**		—	**0.0006**	—	**0.0002**	—	**1.92e-07**	—	**0.0001**	—	**3.74e-09**
Dataset	Metric	TimesFM	+NCL	MOMENT	+NCL	UniTS	+NCL	TSPulse	+NCL	SEMPO	+NCL
IOPS	A-P	0.2180	$0.3242_{\pm 0.0065}$	0.1036	$0.1732_{\pm 0.0009}$	0.1810	$0.3133_{\pm 0.0041}$	0.2038	$0.3494_{\pm 0.0037}$	0.1772	$0.2360_{\pm 0.0077}$
	R-A-P	0.1904	$0.3249_{\pm 0.0066}$	0.0652	$0.1051_{\pm 0.0005}$	0.1524	$0.1874_{\pm 0.0093}$	0.1918	$0.3360_{\pm 0.0025}$	0.1326	$0.1820_{\pm 0.0072}$
	V-PR	0.1902	$0.3244_{\pm 0.0066}$	0.0657	$0.1053_{\pm 0.0005}$	0.1530	$0.1861_{\pm 0.0094}$	0.1919	$0.3354_{\pm 0.0026}$	0.1335	$0.1822_{\pm 0.0070}$
UCR	A-P	0.0636	$0.0612_{\pm 0.0027}$	0.0626	$0.0833_{\pm 0.0003}$	0.0735	$0.0766_{\pm 0.0029}$	0.0228	$0.0300_{\pm 0.0012}$	0.0247	$0.0364_{\pm 0.0017}$
	R-A-P	0.0492	$0.0580_{\pm 0.0025}$	0.0792	$0.0939_{\pm 0.0000}$	0.0827	$0.0843_{\pm 0.0029}$	0.0272	$0.0318_{\pm 0.0020}$	0.0278	$0.0357_{\pm 0.0014}$
	V-PR	0.0543	$0.0600_{\pm 0.0025}$	0.0796	$0.0929_{\pm 0.0000}$	0.0808	$0.0832_{\pm 0.0012}$	0.0274	$0.0320_{\pm 0.0018}$	0.0282	$0.0363_{\pm 0.0014}$
NAB	A-P	0.1941	$0.2947_{\pm 0.0059}$	0.4245	$0.4654_{\pm 0.0001}$	0.3890	$0.4398_{\pm 0.0070}$	0.2258	$0.2902_{\pm 0.0031}$	0.2614	$0.3067_{\pm 0.0040}$
	R-A-P	0.2808	$0.3731_{\pm 0.0055}$	0.5296	$0.5635_{\pm 0.0004}$	0.4528	$0.4933_{\pm 0.0033}$	0.3270	$0.3818_{\pm 0.0048}$	0.3876	$0.4187_{\pm 0.0039}$
	V-PR	0.2804	$0.3728_{\pm 0.0051}$	0.5210	$0.5565_{\pm 0.0003}$	0.4533	$0.4962_{\pm 0.0036}$	0.3201	$0.3750_{\pm 0.0043}$	0.3753	$0.4084_{\pm 0.0037}$
MSL	A-P	0.3122	$0.3715_{\pm 0.0141}$	0.3307	$0.4593_{\pm 0.0037}$	0.4317	$0.4511_{\pm 0.0257}$	0.2582	$0.4197_{\pm 0.0070}$	0.3882	$0.4458_{\pm 0.0149}$
	R-A-P	0.3975	$0.4386_{\pm 0.0209}$	0.4807	$0.5881_{\pm 0.0022}$	0.5169	$0.5410_{\pm 0.0321}$	0.3064	$0.4720_{\pm 0.0089}$	0.4199	$0.5123_{\pm 0.0132}$
	V-PR	0.3928	$0.4360_{\pm 0.0202}$	0.4726	$0.5703_{\pm 0.0028}$	0.5033	$0.5657_{\pm 0.0287}$	0.3060	$0.4694_{\pm 0.0072}$	0.4223	$0.5081_{\pm 0.0135}$
SMAP	A-P	0.2824	$0.3348_{\pm 0.0087}$	0.3485	$0.3827_{\pm 0.0021}$	0.4669	$0.4617_{\pm 0.0171}$	0.1933	$0.3646_{\pm 0.0042}$	0.2225	$0.3885_{\pm 0.0066}$
	R-A-P	0.3441	$0.3935_{\pm 0.0073}$	0.4326	$0.4756_{\pm 0.0012}$	0.5385	$0.5376_{\pm 0.0022}$	0.2253	$0.3699_{\pm 0.0041}$	0.2867	$0.4276_{\pm 0.0095}$
	V-PR	0.3366	$0.3862_{\pm 0.0064}$	0.4288	$0.4716_{\pm 0.0013}$	0.5263	$0.5345_{\pm 0.0046}$	0.2251	$0.3687_{\pm 0.0048}$	0.2830	$0.4227_{\pm 0.0089}$
PSM	A-P	0.3500	$0.4840_{\pm 0.0113}$	0.3735	$0.4751_{\pm 0.0006}$	0.4183	$0.5244_{\pm 0.0054}$	0.3608	$0.4875_{\pm 0.0015}$	0.3670	$0.4162_{\pm 0.0128}$
	R-A-P	0.3496	$0.4779_{\pm 0.0113}$	0.3671	$0.4622_{\pm 0.0006}$	0.4191	$0.4747_{\pm 0.0050}$	0.3612	$0.4829_{\pm 0.0019}$	0.3618	$0.4076_{\pm 0.0124}$
	V-PR	0.3497	$0.4794_{\pm 0.0110}$	0.3682	$0.5279_{\pm 0.0006}$	0.4203	$0.4735_{\pm 0.0050}$	0.3618	$0.4840_{\pm 0.0024}$	0.3624	$0.4084_{\pm 0.0123}$
SMD	A-P	0.1122	$0.1391_{\pm 0.0016}$	0.1402	$0.1878_{\pm 0.0007}$	0.1804	$0.1898_{\pm 0.0006}$	0.1016	$0.1259_{\pm 0.0003}$	0.1060	$0.1295_{\pm 0.0016}$
	R-A-P	0.1124	$0.1496_{\pm 0.0017}$	0.0938	$0.1302_{\pm 0.0005}$	0.1291	$0.1338_{\pm 0.0005}$	0.1040	$0.1244_{\pm 0.0001}$	0.1077	$0.1247_{\pm 0.0017}$
	V-PR	0.1123	$0.1498_{\pm 0.0017}$	0.0942	$0.1312_{\pm 0.0004}$	0.1296	$0.1343_{\pm 0.0004}$	0.1041	$0.1246_{\pm 0.0001}$	0.1080	$0.1249_{\pm 0.0017}$
SWAT	A-P	0.0822	$0.5154_{\pm 0.0660}$	0.1403	$0.5594_{\pm 0.0006}$	0.0862	$0.7237_{\pm 0.0020}$	0.0855	$0.7087_{\pm 0.0006}$	0.1173	$0.5375_{\pm 0.0042}$
	R-A-P	0.1156	$0.5200_{\pm 0.0610}$	0.2032	$0.5571_{\pm 0.0006}$	0.1258	$0.5750_{\pm 0.0092}$	0.1217	$0.6473_{\pm 0.0017}$	0.1783	$0.5559_{\pm 0.0065}$
	V-PR	0.1132	$0.5117_{\pm 0.0592}$	0.1895	$0.5281_{\pm 0.0006}$	0.1215	$0.5346_{\pm 0.0078}$	0.1190	$0.6168_{\pm 0.0009}$	0.1702	$0.5287_{\pm 0.0054}$
PUMP	A-P	0.1201	$0.2111_{\pm 0.0024}$	0.1803	$0.1399_{\pm 0.0002}$	0.1389	$0.2046_{\pm 0.0002}$	0.1140	$0.1898_{\pm 0.0003}$	0.1753	$0.1768_{\pm 0.0013}$
	R-A-P	0.1800	$0.2400_{\pm 0.0046}$	0.2402	$0.2872_{\pm 0.0009}$	0.2415	$0.2525_{\pm 0.0009}$	0.1995	$0.2352_{\pm 0.0004}$	0.2446	$0.2457_{\pm 0.0012}$
	V-PR	0.1792	$0.2380_{\pm 0.0042}$	0.2346	$0.2764_{\pm 0.0011}$	0.2370	$0.2534_{\pm 0.0011}$	0.1963	$0.2324_{\pm 0.0002}$	0.2389	$0.2512_{\pm 0.0012}$
NYC	A-P	0.0277	$0.0286_{\pm 0.0023}$	0.0158	$0.0158_{\pm 0.0001}$	0.1447	$0.2093_{\pm 0.0052}$	0.0334	$0.0422_{\pm 0.0008}$	0.0803	$0.0943_{\pm 0.0296}$
	R-A-P	0.0454	$0.0494_{\pm 0.0027}$	0.0285	$0.0285_{\pm 0.0001}$	0.1496	$0.1846_{\pm 0.0060}$	0.0645	$0.0694_{\pm 0.0041}$	0.0887	$0.1431_{\pm 0.0333}$
	V-PR	0.0444	$0.0483_{\pm 0.0022}$	0.0318	$0.0327_{\pm 0.0001}$	0.1495	$0.1850_{\pm 0.0063}$	0.0624	$0.0670_{\pm 0.0043}$	0.0862	$0.1332_{\pm 0.0232}$
p-**values**		—	**7.43e-05**	—	**0.0003**	—	**0.0062**	—	**0.0001**	—	**0.0004**

Table 8: ROC results of ten vanilla and NCL-augmented TSFMs that use forecasting/reconstruction errors to detect anomalies on ten TSAD datasets. Meanwhile, we perform the paired t-tests, and p-**values** are reported.

Dataset	Metric	Aurora	NCL	ROSE	NCL	Chronos	NCL	TMM	NCL	TimeMoE	NCL
IOPS	A-R	0.6676	$0.7848_{\pm 0.0237}$	0.6770	$0.7845_{\pm 0.0185}$	0.7006	$0.8398_{\pm 0.0005}$	0.6868	$0.8292_{\pm 0.0085}$	0.6908	$0.8301_{\pm 0.0237}$
	R-A-R	0.6943	$0.8016_{\pm 0.0219}$	0.6866	$0.7770_{\pm 0.0125}$	0.7359	$0.8636_{\pm 0.0004}$	0.7158	$0.8437_{\pm 0.0080}$	0.7262	$0.8628_{\pm 0.0202}$
	V-ROC	0.6922	$0.8006_{\pm 0.0221}$	0.6850	$0.7740_{\pm 0.0132}$	0.7318	$0.8621_{\pm 0.0003}$	0.7117	$0.8426_{\pm 0.0080}$	0.7224	$0.8603_{\pm 0.0209}$
UCR	A-R	0.5738	$0.5849_{\pm 0.0056}$	0.5394	$0.5485_{\pm 0.0079}$	0.5567	$0.5916_{\pm 0.0042}$	0.5484	$0.5851_{\pm 0.0018}$	0.5453	$0.5799_{\pm 0.0143}$
	R-A-R	0.6610	$0.6486_{\pm 0.0067}$	0.6317	$0.6244_{\pm 0.0081}$	0.6471	$0.6569_{\pm 0.0037}$	0.6379	$0.6442_{\pm 0.0013}$	0.6405	$0.6568_{\pm 0.0096}$
	V-ROC	0.6530	$0.6445_{\pm 0.0070}$	0.6236	$0.6186_{\pm 0.0080}$	0.6389	$0.6530_{\pm 0.0037}$	0.6295	$0.6403_{\pm 0.0012}$	0.6316	$0.6499_{\pm 0.0105}$
NAB	A-R	0.5689	$0.6139_{\pm 0.0075}$	0.5741	$0.6090_{\pm 0.0084}$	0.5558	$0.6328_{\pm 0.0030}$	0.5530	$0.5828_{\pm 0.0017}$	0.5674	$0.6264_{\pm 0.0054}$
	R-A-R	0.6678	$0.6926_{\pm 0.0051}$	0.6821	$0.6929_{\pm 0.0116}$	0.6662	$0.7164_{\pm 0.0029}$	0.6496	$0.6598_{\pm 0.0028}$	0.6801	$0.7153_{\pm 0.0061}$
	V-ROC	0.6590	$0.6865_{\pm 0.0056}$	0.6729	$0.6859_{\pm 0.0102}$	0.6579	$0.7119_{\pm 0.0028}$	0.6417	$0.6550_{\pm 0.0023}$	0.6724	$0.7096_{\pm 0.0058}$
MSL	A-R	0.6722	$0.7324_{\pm 0.0130}$	0.6953	$0.7823_{\pm 0.0060}$	0.7214	$0.7745_{\pm 0.0012}$	0.6386	$0.7326_{\pm 0.0290}$	0.6511	$0.7578_{\pm 0.0053}$
	R-A-R	0.7565	$0.7821_{\pm 0.0067}$	0.7641	$0.8244_{\pm 0.0032}$	0.8108	$0.8256_{\pm 0.0016}$	0.7195	$0.7737_{\pm 0.0297}$	0.7469	$0.8150_{\pm 0.0117}$
	V-ROC	0.7486	$0.7789_{\pm 0.0086}$	0.7588	$0.8212_{\pm 0.0029}$	0.7990	$0.8209_{\pm 0.0013}$	0.7091	$0.7677_{\pm 0.0298}$	0.7369	$0.8078_{\pm 0.0083}$
SMAP	A-R	0.6556	$0.7304_{\pm 0.0027}$	0.7040	$0.7626_{\pm 0.0042}$	0.6211	$0.6878_{\pm 0.0141}$	0.6070	$0.6333_{\pm 0.0327}$	0.5424	$0.6124_{\pm 0.0138}$
	R-A-R	0.7699	$0.7866_{\pm 0.0020}$	0.7777	$0.8132_{\pm 0.0018}$	0.7269	$0.7430_{\pm 0.0138}$	0.6938	$0.6962_{\pm 0.0212}$	0.6368	$0.6756_{\pm 0.0120}$
	V-ROC	0.7594	$0.7822_{\pm 0.0014}$	0.7715	$0.8099_{\pm 0.0028}$	0.7166	$0.7399_{\pm 0.0141}$	0.6853	$0.6911_{\pm 0.0216}$	0.6280	$0.6709_{\pm 0.0128}$
PSM	A-R	0.5807	$0.6778_{\pm 0.0042}$	0.5818	$0.6341_{\pm 0.0086}$	0.5602	$0.6838_{\pm 0.0021}$	0.5697	$0.6904_{\pm 0.0069}$	0.6047	$0.6816_{\pm 0.0030}$
	R-A-R	0.5748	$0.6616_{\pm 0.0055}$	0.5772	$0.6157_{\pm 0.0103}$	0.5576	$0.6597_{\pm 0.0016}$	0.5691	$0.6648_{\pm 0.0081}$	0.5943	$0.6591_{\pm 0.0024}$
	V-ROC	0.5730	$0.6605_{\pm 0.0050}$	0.5762	$0.6174_{\pm 0.0107}$	0.5566	$0.6611_{\pm 0.0019}$	0.5665	$0.6677_{\pm 0.0085}$	0.5934	$0.6614_{\pm 0.0024}$
SMD	A-R	0.7001	$0.7420_{\pm 0.0004}$	0.6986	$0.7326_{\pm 0.0076}$	0.6660	$0.7278_{\pm 0.0005}$	0.6831	$0.7293_{\pm 0.0073}$	0.6556	$0.7302_{\pm 0.0008}$
	R-A-R	0.7149	$0.7595_{\pm 0.0006}$	0.7068	$0.7373_{\pm 0.0084}$	0.6833	$0.7496_{\pm 0.0002}$	0.6987	$0.7508_{\pm 0.0059}$	0.6731	$0.7494_{\pm 0.0004}$
	V-ROC	0.7136	$0.7575_{\pm 0.0005}$	0.7060	$0.7361_{\pm 0.0083}$	0.6825	$0.7474_{\pm 0.0003}$	0.6972	$0.7489_{\pm 0.0058}$	0.6721	$0.7474_{\pm 0.0004}$
SWAT	A-R	0.2493	$0.8315_{\pm 0.0003}$	0.2709	$0.8286_{\pm 0.0001}$	0.8117	$0.8117_{\pm 0.0001}$	0.2506	$0.8245_{\pm 0.0026}$	0.8194	$0.8192_{\pm 0.0006}$
	R-A-R	0.3638	$0.8374_{\pm 0.0060}$	0.3866	$0.8513_{\pm 0.0023}$	0.7873	$0.7874_{\pm 0.0002}$	0.3625	$0.8127_{\pm 0.0093}$	0.8252	$0.7860_{\pm 0.0011}$
	V-ROC	0.3547	$0.8265_{\pm 0.0064}$	0.3752	$0.8342_{\pm 0.0018}$	0.7670	$0.7670_{\pm 0.0001}$	0.3537	$0.7933_{\pm 0.0082}$	0.8124	$0.7612_{\pm 0.0013}$
PUMP	A-R	0.5130	$0.7778_{\pm 0.0001}$	0.7476	$0.7476_{\pm 0.0010}$	0.7738	$0.8084_{\pm 0.0007}$	0.5245	$0.7904_{\pm 0.0036}$	0.7390	$0.7981_{\pm 0.0011}$
	R-A-R	0.6664	$0.7860_{\pm 0.0009}$	0.7781	$0.7793_{\pm 0.0016}$	0.7651	$0.8010_{\pm 0.0005}$	0.6660	$0.7929_{\pm 0.0020}$	0.7392	$0.7935_{\pm 0.0008}$
	V-ROC	0.6522	$0.7841_{\pm 0.0002}$	0.7749	$0.7757_{\pm 0.0010}$	0.7665	$0.7996_{\pm 0.0005}$	0.6546	$0.7912_{\pm 0.0020}$	0.7396	$0.7920_{\pm 0.0009}$
NYC	A-R	0.4508	$0.6291_{\pm 0.0146}$	0.6599	$0.7849_{\pm 0.0459}$	0.5654	$0.6870_{\pm 0.0169}$	0.6249	$0.6824_{\pm 0.0110}$	0.5669	$0.6386_{\pm 0.0167}$
	R-A-R	0.5394	$0.6422_{\pm 0.0175}$	0.7481	$0.8302_{\pm 0.0428}$	0.6764	$0.7484_{\pm 0.0039}$	0.7546	$0.8142_{\pm 0.0025}$	0.6161	$0.7322_{\pm 0.0072}$
	V-ROC	0.5355	$0.6428_{\pm 0.0152}$	0.7241	$0.8136_{\pm 0.0395}$	0.6608	$0.7263_{\pm 0.0069}$	0.7332	$0.7846_{\pm 0.0038}$	0.6043	$0.7177_{\pm 0.0048}$
p-**values**		—	**0.0002**	—	**0.0017**	—	**4.12e-08**	—	**0.0001**	—	**4.58e-08**
Dataset	Metric	TimesFM	NCL	MOMENT	NCL	UniTS	NCL	TSPulse	NCL	SEMPO	NCL
IOPS	A-R	0.6625	$0.8219_{\pm 0.0037}$	0.6757	$0.7454_{\pm 0.0005}$	0.7446	$0.8155_{\pm 0.0026}$	0.7234	$0.8583_{\pm 0.0015}$	0.7046	$0.7699_{\pm 0.0132}$
	R-A-R	0.6966	$0.8461_{\pm 0.0035}$	0.5501	$0.6539_{\pm 0.0010}$	0.7388	$0.7453_{\pm 0.0071}$	0.7549	$0.8689_{\pm 0.0018}$	0.6968	$0.7563_{\pm 0.0094}$
	V-ROC	0.6929	$0.8447_{\pm 0.0032}$	0.5478	$0.6542_{\pm 0.0010}$	0.7375	$0.7439_{\pm 0.0072}$	0.7521	$0.8679_{\pm 0.0018}$	0.6958	$0.7540_{\pm 0.0090}$
UCR	A-R	0.5617	$0.5970_{\pm 0.0098}$	0.5411	$0.5203_{\pm 0.0002}$	0.5506	$0.5566_{\pm 0.0051}$	0.5803	$0.6005_{\pm 0.0084}$	0.5513	$0.5584_{\pm 0.0107}$
	R-A-R	0.6490	$0.6639_{\pm 0.0066}$	0.5790	$0.5833_{\pm 0.0006}$	0.6055	$0.6116_{\pm 0.0053}$	0.6680	$0.6655_{\pm 0.0090}$	0.6262	$0.6241_{\pm 0.0038}$
	V-ROC	0.6415	$0.6591_{\pm 0.0069}$	0.5781	$0.5776_{\pm 0.0004}$	0.6021	$0.6083_{\pm 0.0053}$	0.6591	$0.6601_{\pm 0.0076}$	0.6210	$0.6198_{\pm 0.0044}$
NAB	A-R	0.5648	$0.6328_{\pm 0.0023}$	0.7029	$0.7476_{\pm 0.0005}$	0.6841	$0.7097_{\pm 0.0035}$	0.5758	$0.6117_{\pm 0.0045}$	0.5862	$0.6148_{\pm 0.0029}$
	R-A-R	0.6725	$0.7192_{\pm 0.0015}$	0.7461	$0.7973_{\pm 0.0004}$	0.7458	$0.7625_{\pm 0.0025}$	0.6665	$0.6890_{\pm 0.0067}$	0.6932	$0.7026_{\pm 0.0051}$
	V-ROC	0.6653	$0.7141_{\pm 0.0009}$	0.7435	$0.7944_{\pm 0.0002}$	0.7434	$0.7613_{\pm 0.0028}$	0.6586	$0.6812_{\pm 0.0060}$	0.6807	$0.6927_{\pm 0.0040}$
MSL	A-R	0.7101	$0.7798_{\pm 0.0163}$	0.8043	$0.8204_{\pm 0.0005}$	0.8067	$0.8216_{\pm 0.0146}$	0.6664	$0.7575_{\pm 0.0029}$	0.6869	$0.7762_{\pm 0.0174}$
	R-A-R	0.7980	$0.8395_{\pm 0.0154}$	0.8326	$0.8740_{\pm 0.0003}$	0.8394	$0.8560_{\pm 0.0149}$	0.7431	$0.8064_{\pm 0.0023}$	0.7525	$0.8134_{\pm 0.0177}$
	V-ROC	0.7873	$0.8321_{\pm 0.0148}$	0.8322	$0.8653_{\pm 0.0004}$	0.8333	$0.8487_{\pm 0.0141}$	0.7356	$0.8019_{\pm 0.0019}$	0.7473	$0.8114_{\pm 0.0153}$
SMAP	A-R	0.6195	$0.6883_{\pm 0.0132}$	0.8342	$0.7859_{\pm 0.0011}$	0.7654	$0.7603_{\pm 0.0047}$	0.6752	$0.7069_{\pm 0.0068}$	0.7308	$0.7646_{\pm 0.0041}$
	R-A-R	0.7175	$0.7559_{\pm 0.0070}$	0.8605	$0.8302_{\pm 0.0010}$	0.8042	$0.8036_{\pm 0.0004}$	0.7832	$0.7773_{\pm 0.0059}$	0.8191	$0.8246_{\pm 0.0035}$
	V-ROC	0.7062	$0.7487_{\pm 0.0068}$	0.8584	$0.8265_{\pm 0.0010}$	0.8000	$0.7985_{\pm 0.0012}$	0.7732	$0.7722_{\pm 0.0063}$	0.8098	$0.8202_{\pm 0.0030}$
PSM	A-R	0.5610	$0.6826_{\pm 0.0131}$	0.6192	$0.7000_{\pm 0.0014}$	0.6387	$0.7100_{\pm 0.0062}$	0.5947	$0.6991_{\pm 0.0014}$	0.5959	$0.6449_{\pm 0.0095}$
	R-A-R	0.5580	$0.6648_{\pm 0.0126}$	0.5731	$0.6570_{\pm 0.0014}$	0.6174	$0.6402_{\pm 0.0060}$	0.5902	$0.6793_{\pm 0.0031}$	0.5780	$0.6202_{\pm 0.0125}$
	V-ROC	0.5572	$0.6653_{\pm 0.0125}$	0.5742	$0.6565_{\pm 0.0014}$	0.6160	$0.6381_{\pm 0.0060}$	0.5896	$0.6789_{\pm 0.0036}$	0.5776	$0.6206_{\pm 0.0124}$
SMD	A-R	0.6719	$0.7511_{\pm 0.0016}$	0.7568	$0.7875_{\pm 0.0009}$	0.7590	$0.7697_{\pm 0.0009}$	0.7170	$0.7438_{\pm 0.0004}$	0.7194	$0.7378_{\pm 0.0024}$
	R-A-R	0.6885	$0.7755_{\pm 0.0011}$	0.5617	$0.6780_{\pm 0.0037}$	0.6875	$0.6696_{\pm 0.0039}$	0.7263	$0.7475_{\pm 0.0003}$	0.7211	$0.7310_{\pm 0.0020}$
	V-ROC	0.6875	$0.7735_{\pm 0.0012}$	0.5616	$0.6774_{\pm 0.0035}$	0.6854	$0.6681_{\pm 0.0036}$	0.7249	$0.7465_{\pm 0.0003}$	0.7207	$0.7302_{\pm 0.0018}$
SWAT	A-R	0.2567	$0.8202_{\pm 0.0048}$	0.2730	$0.8258_{\pm 0.0013}$	0.2339	$0.8274_{\pm 0.0005}$	0.2444	$0.8283_{\pm 0.0002}$	0.2721	$0.8298_{\pm 0.0017}$
	R-A-R	0.3604	$0.8228_{\pm 0.0084}$	0.3918	$0.8079_{\pm 0.0015}$	0.3523	$0.7533_{\pm 0.0043}$	0.3585	$0.8044_{\pm 0.0031}$	0.3861	$0.8296_{\pm 0.0033}$
	V-ROC	0.3519	$0.8116_{\pm 0.0100}$	0.3784	$0.7885_{\pm 0.0015}$	0.3417	$0.7148_{\pm 0.0049}$	0.3489	$0.7845_{\pm 0.0030}$	0.3729	$0.8086_{\pm 0.0027}$
PUMP	A-R	0.5532	$0.7942_{\pm 0.0037}$	0.7587	$0.5406_{\pm 0.0009}$	0.5598	$0.7914_{\pm 0.0009}$	0.5066	$0.7812_{\pm 0.0002}$	0.7436	$0.7475_{\pm 0.0034}$
	R-A-R	0.6501	$0.7962_{\pm 0.0018}$	0.7814	$0.7370_{\pm 0.0006}$	0.6904	$0.7904_{\pm 0.0006}$	0.6459	$0.7895_{\pm 0.0004}$	0.7800	$0.7820_{\pm 0.0018}$
	V-ROC	0.6440	$0.7941_{\pm 0.0020}$	0.7780	$0.7201_{\pm 0.0006}$	0.6828	$0.7873_{\pm 0.0006}$	0.6334	$0.7875_{\pm 0.0003}$	0.7757	$0.7779_{\pm 0.0020}$
NYC	A-R	0.5439	$0.5591_{\pm 0.0129}$	0.3474	$0.3531_{\pm 0.0051}$	0.8186	$0.8620_{\pm 0.0036}$	0.5723	$0.7032_{\pm 0.0220}$	0.7382	$0.7915_{\pm 0.0131}$
	R-A-R	0.6510	$0.6660_{\pm 0.0105}$	0.4945	$0.4983_{\pm 0.0027}$	0.8354	$0.8513_{\pm 0.0016}$	0.7318	$0.7605_{\pm 0.0211}$	0.7813	$0.8205_{\pm 0.0074}$
	V-ROC	0.6361	$0.6513_{\pm 0.0087}$	0.4989	$0.5040_{\pm 0.0027}$	0.8264	$0.8434_{\pm 0.0017}$	0.7106	$0.7490_{\pm 0.0229}$	0.7718	$0.8093_{\pm 0.0095}$
p-**values**		—	**4.28e-05**	—	**0.0237**	—	**0.0092**	—	**0.0002**	—	**0.0066**

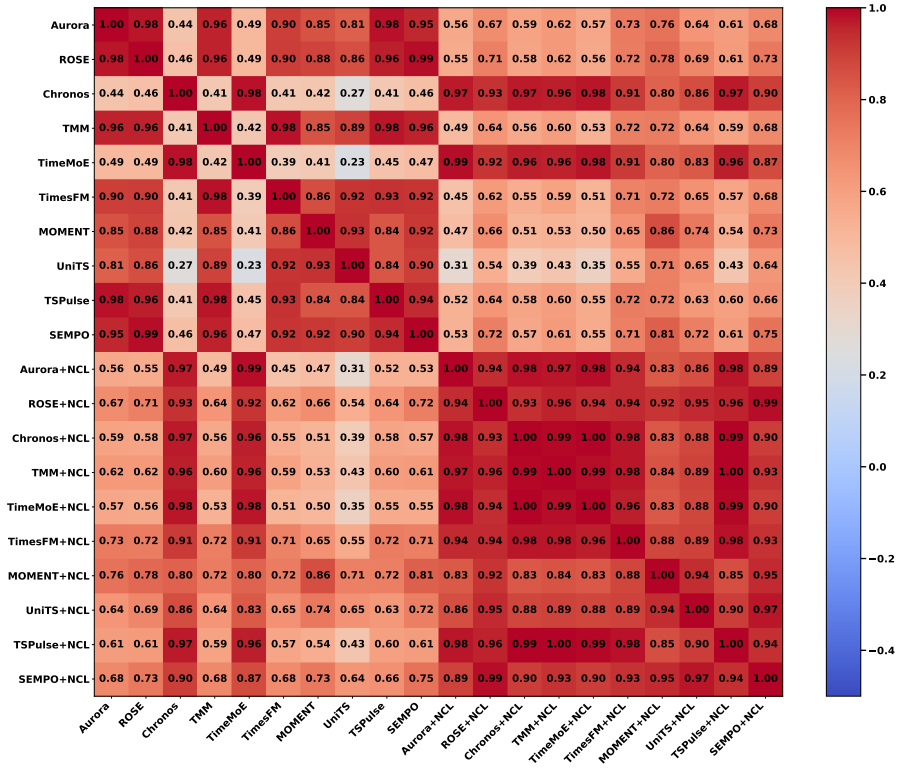


Figure 14: Pairwise correlation in PR metrics.

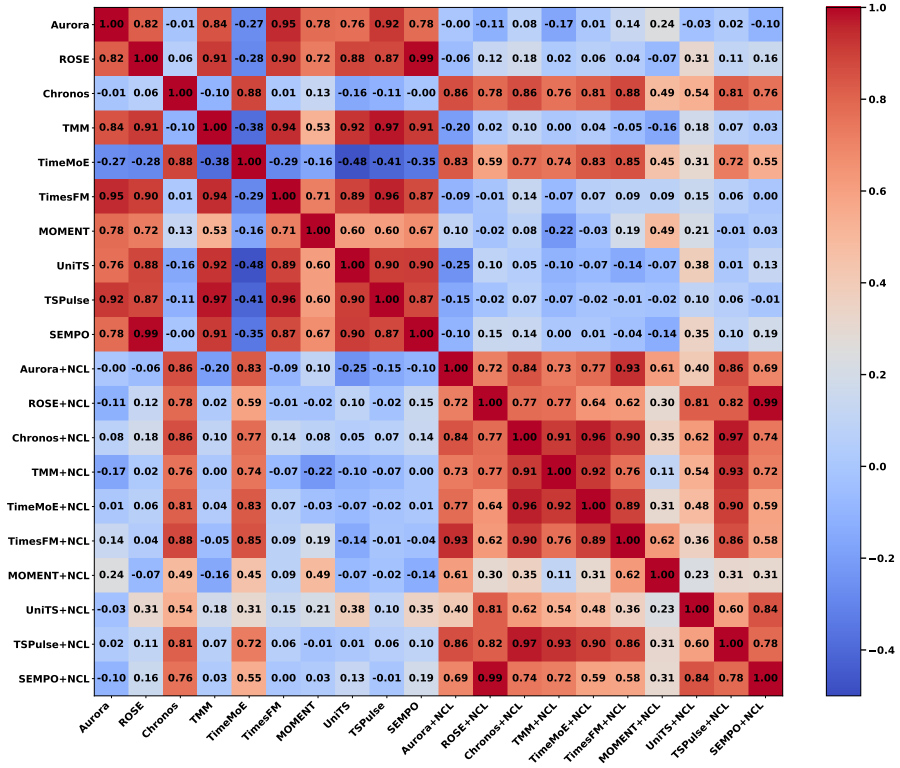


Figure 15: Pairwise correlation in ROC metrics.