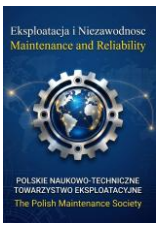

Eksploatacja i Niezawodnosc - Maintenance and Reliability

Volume 29 (2027), Issue 1

https://ein.org.pl

The Polish Maintenance Society

Test-time adaptation for cross-domain remaining useful life prediction: when does it work and what should practitioners use

Jinhe Yu[1], Jianyin Zhao[1,*], Yufeng Qin[1]

[1] Naval Aviation University, China

* Corresponding Author: zhaojianyin7603@163.com

Received: 10 May 2026
Revised: 9 July 2026
Accepted: 17 July 2026
Online: 25 July 2026

Abstract

Deep learning models for remaining useful life (RUL) prediction degrade under varying operating conditions due to distribution shifts. Test-time adaptation (TTA) addresses this issue by updating a pretrained model using unlabeled target data without source data access; however, it has not been systematically studied for RUL prediction. Four TTA strategies are reformulated and evaluated across six cross-domain scenarios on C-MAPSS. TTA improves performance only when the source has more diverse operating conditions than the target. Restricting adaptation to batch-normalization (BN) affine parameters makes all gradient-based methods produce functionally equivalent outputs (BN affine bottleneck). Gradient-free adaptive BN gives the best risk–benefit trade-off, matching or outperforming them in five of six scenarios. A four-step deployment workflow for cross-domain predictive maintenance is proposed. The direction dependence of TTA effectiveness and the BN affine bottleneck are further confirmed on an LSTM backbone and the N-CMAPSS dataset.

Keywords: *remaining useful life, predictive maintenance, test-time adaptation, cross-domain deployment, condition monitoring, distribution shift*

Article citation:
Yu J, Zhao J, Qin Y, Test-time adaptation for cross-domain remaining useful life prediction: when does it work and what should practitioners use, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2027: 29(1) http://doi.org/10.17531/ein/226057

Highlights

- Direction of distribution shift determines test-time adaptation (TTA) effectiveness.
- BN affine bottleneck makes gradient TTA methods functionally equivalent.
- Gradient-free adaptive BN matches gradient-based methods in 5 of 6 scenarios.
- Four-step deployment workflow guides strategy choice for maintenance engineers.

1. Introduction

Unplanned downtime in industrial equipment can result in significant economic losses, often ranging from tens to hundreds of thousands of dollars per hour. Consequently, predicting the remaining useful life (RUL) of equipment through condition monitoring has become a critical component of predictive maintenance [1]. Traditional methods predominantly utilized physics-based and statistical degradation models (e.g., Wiener and Gamma processes), which required prior knowledge of the degradation mechanism [2]. In contrast, data-driven methods learn degradation patterns directly from sensor signals, bypassing these limitations. Long short-term memory (LSTM) networks [3] and bidirectional LSTMs [4] are prevalent sequence models in RUL prediction owing to their effective gating mechanisms. Additionally, one-dimensional convolutional neural networks employ localized convolutional kernels to capture short-range temporal patterns [5]. Temporal convolutional networks (TCNs) [6] leverage dilated causal convolutions to create an exponentially growing receptive field and have demonstrated competitive performance in RUL predictions [7]. In addition, although Transformers have been introduced for RUL prediction, harnessing their self-attention mechanism [8], they face challenges related to quadratic computational complexity and high data demands in industrial degradation datasets. In this study, we adopted TCNs as the backbone network to balance the modeling capacity and computational efficiency. Recently, Li et al. [9] conducted a systematic comparison of various generative models and

prediction networks under different degradation scenarios, underscoring the ongoing debate regarding method selection.

The success of the aforementioned data-driven methods hinges on the assumption that the training data and deployment environments have similar distributions. However, in practice, systematic differences in operating conditions and fault modes between the training and deployment phases, termed distribution shifts, can lead to a significant degradation in the prediction accuracy of source-domain-trained models when applied in the target domain.

Domain adaptation (DA) serves as the principal strategy for mitigating issues related to distribution shifts. In the RUL prediction domain, key methods include distribution-distance minimization (e.g., maximum mean discrepancy (MMD) [10] and correlation alignment (CORAL) [11]), adversarial training (e.g., domain-adversarial neural networks (DANNs) [12] and their RUL-specific variants [13]), and transfer learning via source-domain pretraining followed by target-domain fine-tuning [14]. Lv et al. [15] further explored multisource unsupervised alignment methods leveraging TCNs. Recent studies continue to expand the methodological frontier of cross-domain RUL prediction. For example, Nejjar et al. [16] achieved DA amid varying degradation conditions through operational profile alignment; Hou et al. [17] introduced a target-oriented feature reconstruction and degradation-stage consistency alignment strategy; whereas Dong et al. [18] proposed a multi-constrained domain adaptation network that jointly aligns the marginal and conditional distributions between the source and target domains. However, these strategies generally require simultaneous access to source- and target-domain data during training, a prerequisite that is often not met in practice because of intellectual-property restrictions, the need for rapid online adaptation to previously unseen operating conditions, and data-sharing regulations in sectors such as aerospace and nuclear power.

Test-time adaptation (TTA) provides an alternative pathway under the above constraints, requiring only a pretrained model and unlabeled target-domain data available during testing [19]. Adaptive batch normalization (ABN) [20] enables gradient-free adaptation by replacing the batch normalization (BN) layer statistics. TENT [21] minimizes the prediction entropy and updates only the BN affine parameters, which is a configuration widely adopted in subsequent studies. EATA [22] and SAR [23] introduce sample selection and sharpness-aware minimization, respectively, to enhance adaptation stability. T3A [24] achieves pseudo-label adaptation by maintaining pseudo-prototypes. Feature alignment (FAL) methods [11] synchronize the feature statistics of the source and target domains during testing. CoTTA [25] mitigates error accumulation during continual adaptation through weight averaging and stochastic restoration. TTT [26] embeds self-supervised auxiliary tasks during training to bolster TTA. Wu et al. [27] enhanced the TTA mechanism for BN layers from a domain-aware perspective, while Kim et al. [28] recently adopted TTA for time-series forecasting tasks, demonstrating its applicability for continuous-valued prediction.

Despite these advances, the aforementioned TTA methods are primarily designed for classification. The discrete probability distribution of classification outputs provides direct optimization signals for entropy minimization and confidence-based sample selection. Conversely, RUL prediction is a continuous-valued regression task that lacks such a class probability distribution, necessitating the redesign of adaptation objectives. Moreover, existing studies have focused mainly on performance ranking among strategies, leaving a fundamental question unexplored: when adaptable parameters are restricted to the affine parameters of BN layers, can gradient updates driven by different loss functions converge in similar effective directions, resulting in effectively similar adaptation behavior? In addition, label truncation and other pre-processing operations specific to regression tasks may induce failure modes absent in classification tasks. However, these issues remain systematically unexamined.

This context raises a pivotal question for maintenance engineers deploying pretrained predictive models: When a pretrained RUL model must adapt to a new operating environment without access to the original training data or failure labels from the target environment, can TTA yield reliable improvements, and which strategy should be implemented? Accordingly, this study is organized around three research questions.

RQ1. Is the effectiveness of TTA in cross-domain RUL prediction influenced by the direction of the distribution shift?

RQ2. When adaptation is restricted to the BN affine

parameters, do TTA loss functions with different objectives converge to functionally equivalent behavior?

RQ3. Do gradient-based TTA methods provide a practical advantage over gradient-free statistics replacement?

The scope of this study is limited to architectures that contain BN layers, such as the TCN and LSTM backbones used here, because the BN affine bottleneck is defined with respect to the affine parameters of BN layers. Whether a comparable bottleneck arises in normalization-free or layer-normalized architectures, such as Transformers, is outside the scope of this study and is left for future work.

The main contributions of this study are as follows. (1) Using a TCN as the primary backbone, four TTA strategies, namely ABN, variance minimization (VMN), pseudo-label self-training (PSL), and FAL, are reformulated for regression and systematically evaluated across six cross-domain scenarios using the C-MAPSS dataset, with the consistency of these findings confirmed on an LSTM backbone and the N-CMAPSS dataset. Results demonstrate that TTA effectiveness is highly dependent on the shift direction, with consistent improvements noted only when the source domain is more complex than the target domain. (2) The BN affine bottleneck (BAB) phenomenon was identified and verified: when adaptation is limited to approximately 3,072 BN affine parameters, the four gradient-based methods yield functionally equivalent outputs. (3) The gradient-free ABN matches or outperforms all gradient-based methods in five of the six evaluated scenarios. (4) A sequential deployment decision workflow for maintenance engineers is proposed based on these findings.

2. Material and methods

The workflow of this study is divided into two phases: a source-domain training phase, in which a TCN with uncertainty estimation capability was trained on labeled data, serving as the backbone model, and a TTA phase, which involves adjusting partial model parameters on unlabeled target-domain data through four distinct strategies, statistics replacement, VMN, PSL, and FAL, to mitigate the distribution shift. Fig. 1 illustrates the overall architecture of these two phases.

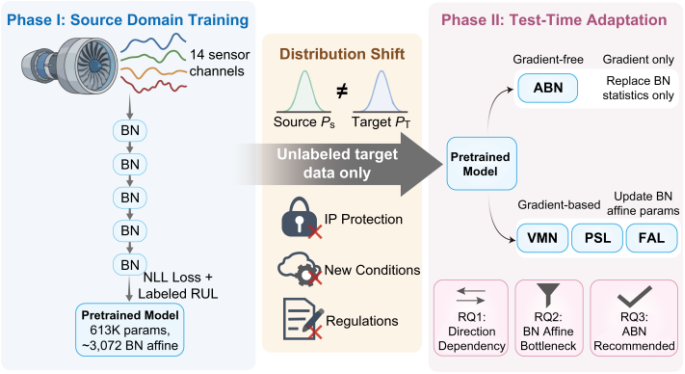

Figure 1. Overall research framework. Left: Source domain training phase, where 14-channel sensor data are fed into a temporal convolutional network backbone with BN layers, trained with negative log-likelihood (NLL) loss to obtain a pretrained model (613K parameters and ~3,072 BN affine parameters). Right: Test-time adaptation (TTA) phase using only unlabeled target data via gradient-free (ABN) and gradient-based (VMN, PSL, FAL) strategies. The three research questions (RQ1–RQ3), defined in Section 1, are listed at the bottom. ABN, adaptive batch normalization; VMN, variance minimization; PSL, pseudo-label self-training; FAL, feature alignment.

2.1. Problem definition

Let the source-domain training dataset be $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, where $x_i^s \in R^{T\times d}$ is a multivariate time series comprising T time steps and d sensor channels, and $y_i^s \in R^+$ is the corresponding RUL label. The target-domain test dataset, denoted $D_t = \{x_j^t\}_{j=1}^{N_t}$, lacks label information at test time. The marginal distributions of the source and target domains differ, that is, $P_s(x) \neq P_t(x)$, although both domains share the same prediction task. The BN layer affine parameters are denoted as $\theta_{BN} = \{\gamma_l, \beta_l\}_{l=1}^{L}$, where γ_l and β_l are the scale and shift parameters of the l-th layer, respectively, and L is the total number of BN layers. These parameters serve as primary targets for subsequent TTA strategies.

The objective of TTA is to minimize prediction error on the target domain by adjusting a subset of model parameters using only a pretrained source model and unlabeled target-domain data.

2.2. Backbone network and source-domain training

2.2.1. Temporal convolutional network

The backbone network employs a TCN consisting of an input

projection layer and six stacked residual convolutional blocks. The input projection layer maps the raw d-dimensional sensor features to a 128-dimensional hidden space via a 1D convolution with a kernel size of 1. Each residual block comprises two cascaded units of 1D dilated convolution, BN, ReLU activation, and dropout, supplemented by a residual shortcut connection. The dilation factors of the six blocks are 1, 2, 4, 8, 16, and 32, with a uniform kernel size of 3, yielding a receptive field of 253 time steps. This field adequately encompasses the input window of length $T = 30$. The network contains 12 BN layers whose affine parameters—scale factor γ and shift parameter β—constitute the primary adaptation targets for all subsequent TTA methods. In the extended adaptation experiments, we additionally adapt the convolutional weights of the last two residual blocks (blocks 5 and 6) since higher-level features are closer to the regression output and more sensitive to domain shifts, while keeping the lower-level feature extractors frozen.

The convolutional features are globally average-pooled to produce a 128-dimensional feature vector, which is then processed by a two-layer fully connected regression head (128 → 64 → 2) that outputs a predicted mean $\hat{\mu}$ and log-variance $log\ \hat{\sigma}^2$. This dual-output design facilitates point prediction and uncertainty estimation, providing critical inputs for subsequent VMN and PSL strategies. The model contains approximately 613K parameters, with the affine parameters from the 12 BN layers accounting for approximately 3,072.

2.2.2. Training loss function

The source model was trained using the Gaussian negative log-likelihood (NLL) as the loss function.

$$\mathcal{L}_{\mathrm{NLL}} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\log\hat{\sigma}_i^2 + \frac{(\hat{\mu}_i - y_i)^2}{2\hat{\sigma}_i^2}\right] \quad (1)$$

where $\hat{\mu}_i$ is the predicted RUL mean for the i-th sample, $\hat{\sigma}_i^2 = exp(log\ \hat{\sigma}_i^2)$ indicates the predicted variance, y_i is the ground truth RUL label, and N is the batch size. Compared with the standard mean-squared error loss, NLL jointly optimizes the mean and variance, encouraging the model to output larger uncertainty estimates in areas where predictions are more difficult.

2.3. Test-time adaptation strategies

This study evaluates four TTA strategies within a unified framework, comprising one gradient-free method (ABN) and three gradient-based methods (VMN, PSL, and FAL). Each of the four strategies adapts an established TTA principle from the classification setting to continuous-valued regression: ABN reuses the gradient-free replacement of BN statistics introduced by Li et al. 20; VMN reformulates the entropy minimization of TENT 21 by replacing discrete-class entropy with predicted log-variance; PSL adapts Monte Carlo dropout pseudo-labeling; and FAL applies the CORAL distance 11 at test time. In addition, a stabilized variant of VMN, denoted VMN-S, is introduced in Section 2.3.3; the functional-equivalence analysis in Section 3.3 therefore compares four gradient-based methods (VMN, VMN-S, PSL, and FAL). The novelty of this work therefore lies not in the individual loss designs, but in (i) their unified reformulation for regression, and (ii) the empirical phenomena revealed through systematic evaluation—most notably the BN affine bottleneck and the direction dependence of TTA effectiveness.

Unless otherwise stated, the default configuration for the gradient-based methods was as follows: only the BN affine parameters were adapted, a single gradient update was performed per mini-batch, the learning rate was 10^{-3}, and the momentum coefficient for the stochastic gradient descent with momentum optimizer was 0.9.

2.3.1. Adaptive batch normalization

ABN is a gradient-free statistical adaptation method. During inference, all BN layers are switched to training mode so that they compute and use the mean and variance of the current target-domain mini-batch instead of the source-domain running statistics. Forward inference is then performed directly using these updated statistics. ABN does not involve parameter updates or gradient-based optimization and does not introduce additional hyperparameters.

2.3.2. Variance minimization

VMN adapts the entropy minimization technique from classification to a form suitable for regression. In regression, the outputs are continuous values rather than class probabilities. Therefore, VMN uses the predicted variance as the optimization signal, minimizing the log variance of the target-domain mini-batch:

$$\mathcal{L}_{\mathrm{VMN}} = \frac{1}{|B|}\sum_{i\in B} log\ \hat{\sigma}_i^2 \quad (2)$$

where B is the current target-domain mini-batch and $\hat{\sigma}_i^2$ is the predicted variance of the model for the i-th sample. Minimizing $log\,\hat{\sigma}^2$ encourages the model to make more confident predictions for target-domain samples. Additionally, VMN incorporates a consistency regularization term that penalizes prediction discrepancies between the original input and its augmented view.

2.3.3. Stabilized variance minimization

VMN-S extends VMN with two mechanisms to improve stability. First, parameter anchoring introduces a quadratic regularization term that penalizes deviations of the adapted BN affine parameters from their source-domain values (with a default coefficient $\lambda = 0.1$). Second, an adaptive gating mechanism computes a per-sample confidence score based on the change in predicted variance before and after adaptation. If adaptation increases uncertainty, the model reverts to the source-domain prediction using a convex combination of the pre- and post-adaptation outputs. The complete formulations of the regularization term, total loss, and gating mechanism are provided in Appendix A.1.

2.3.4. Pseudo-label self-training

PSL employs Monte Carlo (MC) dropout to estimate prediction uncertainty and identify reliable samples for adaptation. During inference, M stochastic forward passes (with $M = 5$ by default) are performed for each sample with dropout enabled. Samples are ranked according to their epistemic variance, and the top K% with the lowest variance (default $K = 30$) are selected as high-confidence samples. The mean of their MC predictions is used as a pseudo-label. The BN affine parameters are then fine-tuned by minimizing the mean squared error between the current predictions and these pseudo-labels. Detailed formulations are provided in Appendix A.2.

2.3.5. Feature alignment

FAL minimizes the distributional discrepancy between source- and target-domain feature representations at the inputs of BN layers. After source-domain training, the mean and covariance of the input features at each BN layer are computed offline and stored as reference statistics. During testing, corresponding statistics are computed for each target-domain mini-batch, and the covariance discrepancy is minimized using the CORAL distance [11] across all 12 BN layers. The complete loss formulation is provided in Appendix A.3.

3. Results

To systematically evaluate TTA performance in cross-domain RUL prediction, we developed a unified experimental benchmark framework encompassing multiple shift directions, TTA strategies, and parameter adaptation settings. This section outlines the experimental setup and subsequently analyzes the results with respect to the three research questions (RQ1–RQ3) defined in Section 1.

3.1. Experimental setup

All experiments were conducted using the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) turbofan engine degradation benchmark dataset [29]. This dataset comprises four subsets: FD001–FD004, which systematically differ in the number of operating conditions and fault modes. Specifically, FD001 and FD003 feature a single operating condition, whereas FD002 and FD004 encompass six. Moreover, FD001 and FD002 contain only high-pressure compressor (HPC) degradation, whereas FD003 and FD004 include HPC and fan degradation. Each engine record consists of 21 sensor channels. Following established conventions, we eliminated seven uninformative sensors with near-zero variance (sensors 1, 5, 6, 10, 16, 18, and 19), retaining 14 effective channels as input features. We also excluded the three operating condition setting columns from the input to preserve distribution disparities across the subsets.

Sensor readings were normalized to [0, 1] using min–max scaling fitted on the source-domain training set. The same normalization parameters were applied to the target domain during testing, consistent with the TTA deployment assumption. RUL labels were derived using a piecewise linear degradation model with an upper clipping threshold of 125 operating cycles. Input samples were generated using a sliding window of length 30 and a stride of 1. During evaluation, only the last window of each test engine was used for prediction.

To systematically evaluate TTA performance under various distribution shift conditions, we defined six cross-domain transfer scenarios S1–S6, addressing operating condition and fault mode discrepancies, as well as their combinations. Table 1 lists the source–target dataset pairs and the corresponding shift

characteristics for each scenario.

Table 1. Definitions of the six cross-domain transfer scenarios.

Scenario	Source → Target	Shift characteristics
S1	FD001→FD002	Single → multiple operating conditions; same fault mode
S2	FD001→FD003	Same operating condition; single → dual fault modes
S3	FD002→FD001	Multiple → single operating condition; same fault mode
S4	FD003→FD001	Same operating condition; dual → single fault mode
S5	FD001→FD004	Single → multiple operating conditions + dual fault modes
S6	FD003→FD004	Single → multiple operating conditions; comparable fault complexity

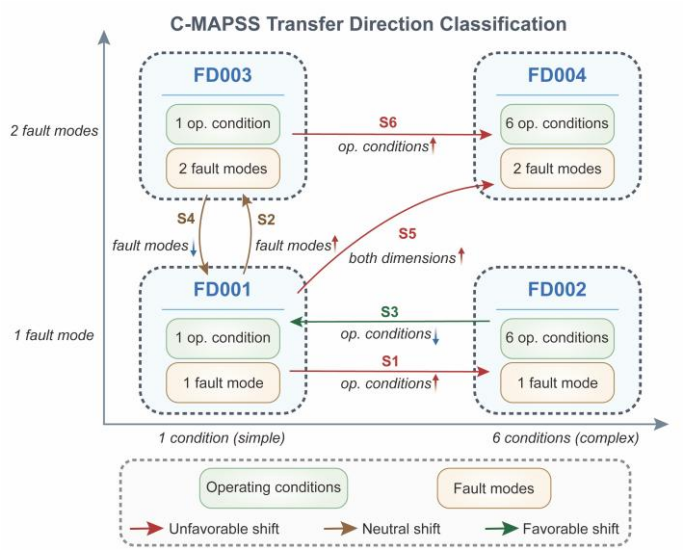

Figure 2. Transfer direction classification of six C-MAPSS cross-domain scenarios. The horizontal axis represents the number of operating conditions (1–6), and the vertical axis represents the number of fault modes (1–2). Green arrows denote favorable shifts (source more complex), brown arrows denote neutral shifts, and red arrows denote unfavorable shifts (target more complex). The shifted dimensions are annotated alongside each arrow.

Fig. 2 visualizes the transfer direction classification of the six scenarios on a two-dimensional plane defined by the number of operating conditions and fault modes. Scenarios S1, S5, and S6 (red arrows) correspond to unfavorable shifts from simpler source domains to more complex target domains. Scenario S3 (green arrow) represents the only favorable shift, in which a model trained under multiple operating conditions is deployed in a single-condition environment. Scenarios S2 and S4 (brown arrows) correspond to neutral shifts in which the operating conditions remain unchanged while only the fault mode varies—S2 transitions from a single fault mode to dual fault modes, whereas S4 reverses this direction.

This study evaluated seven methods categorized into three groups:

(1) No-adaptation baselines

Source-only (SRC): It involves direct prediction using the source model without any adaptation, serving as a performance lower bound.

ORA (Oracle): It involves supervised fine-tuning of BN affine parameters using ground truth labels from the target domain, providing an idealized upper-bound reference that is unavailable in practice.

(2) Gradient-free TTA

ABN: During testing, BN layers operate in training mode, allowing the running statistics from the source domain to be supplanted by batch statistics from the target-domain mini-batch, without requiring gradient computation or hyperparameter tuning.

(3) Gradient-based TTA

The following four methods adapt only the BN affine parameters and perform a single gradient step per mini-batch:

VMN: VMN minimizes the log-variance of model predictions.

VMN-S: VMN-S enhances VMN by integrating parameter anchoring regularization and an adaptive gating mechanism.

PSL: PSL utilizes self-training on high-confidence samples selected via MC dropout.

FAL: FAL minimizes the CORAL covariance distance between the source- and target-domain features at each BN layer input.

Two parameter adaptation scopes are defined to assess the impact of parameter-space dimensionality on adaptation behavior: **Scope-BN** adapts only the affine parameters of all 12 BN layers, yielding approximately 3,072 trainable parameters; **Scope-Ext** additionally adapts the convolutional weights of four convolutional layers in the last two TCN residual blocks, whose high-level features are more sensitive to domain shifts, yielding approximately 131K trainable parameters.

The source model was trained using the Adam optimizer with an initial learning rate of 10^{-3} and cosine annealing over 50 epochs. The training batch size was 256, gradient norm clipping was set to 1.0, and weight decay was 10^{-4}. An 80/20 train/validation split was performed using the engine ID,

ensuring that windows from the same engine do not appear in both splits. Early stopping was applied based on validation root mean square error (RMSE) with a patience of 10 epochs. Each source-domain subset was trained independently under five random seeds.

RMSE was used as the primary evaluation metric:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \quad (3)$$

where $\hat{y}_i$ represents the predicted RUL, y_i denotes the ground truth RUL, and N is the number of test samples.

In total, the main experiments comprise 420 runs (7 methods × 6 scenarios × 2 scopes × 5 seeds), supplemented by 144 ablation runs (4 gradient-based methods × 3 adaptation-step settings {1, 5, 10} × 2 scopes × 3 scenarios × 2 seeds). All results are reported as mean ± standard deviation across repeated trials. Statistical significance is evaluated using the paired Wilcoxon signed-rank test (see Section 3.6).

3.2. Direction dependence of TTA effectiveness (RQ1)

Table 2 summarizes the RMSE results of all seven methods across six cross-domain scenarios under Scope-BN. The findings demonstrate a systematic relationship between TTA effectiveness and the direction of the distribution shift. The six scenarios can be categorized into three groups based on the shift direction:

Table 2. RMSE results (↓) of all methods under Scope-BN (mean ± std over 5 seeds). Bold indicates the best TTA method. Δ denotes the RMSE change of the best TTA method relative to SRC (negative values indicate improvement). All values are rounded to one decimal; Δ is computed from unrounded means.

Scenario	SRC	ORA	ABN	VMN	VMN-S	PSL	FAL	Best TTA Δ
S1	69.4±0.0	70.8±14.9	89.1±11.0	**88.8±11.9**	**88.8±11.9**	**88.8±11.9**	**88.8±11.9**	+19.4
S2	39.3±9.8	21.1±1.5	**40.7±6.9**	42.5±5.5	42.5±5.5	42.5±5.5	42.5±5.5	+1.4
S3	24.1±1.1	22.6±5.0	26.4±3.5	**23.0±1.5**	**23.0±1.5**	**23.0±1.5**	**23.0±1.5**	-1.2
S4	15.5±0.7	17.2±0.7	**19.2±1.2**	25.1±1.2	25.1±1.2	25.1±1.2	25.1±1.2	+3.8
S5	66.7±0.0	77.2±17.7	94.0±11.0	**93.8±11.8**	**93.8±11.8**	**93.8±11.8**	**93.8±11.8**	+27.0
S6	66.7±0.0	69.2±16.0	**80.2±17.8**	85.6±16.6	85.6±16.6	85.6±16.6	85.6±16.6	+13.5

Abbreviations: SRC, source-only (no adaptation); ORA, oracle (supervised fine-tuning with ground-truth labels); ABN, adaptive batch normalization; VMN, variance minimization; VMN-S, stabilized variance minimization; PSL, pseudo-label self-training; FAL, feature alignment; TTA, test-time adaptation; RMSE, root mean square error; BN, batch normalization.

Favorable direction (source domain is more complex). Scenario S3 (FD002 → FD001, six conditions → one condition) is the only case in which TTA yields an improvement: the gradient-based methods decrease RMSE from 24.1 to approximately 23.0, whereas ABN increases RMSE to 26.4 in this scenario. The increased diversity of operating conditions in the source domain facilitates effective recalibration of the target-domain feature distribution through BN statistics replacement and affine parameter adjustment.

Neutral direction (comparable complexity). In S2, the source and target domains share a single operating condition but differ in fault mode. The magnitude shift in BN statistics is small, and TTA adjustment is limited; the best-performing method, ABN, achieves an RMSE of 40.7, compared to 39.3 for SRC, indicating a marginal increase of 1.4. In S4, despite the source domain exhibiting more complex fault modes, it shares the same single operating condition, resulting in TTA degradation (RMSE increases from 15.5 to 19.2). This suggests that the advantage of the source domain for TTA primarily arises from the diversity of operating conditions rather than fault-mode complexity. The key insight is that different operating conditions directly influence the mean and variance of sensor signals, specifically, the low-order statistics governed by BN layers, whereas variations in fault modes primarily affect the temporal evolution patterns of degradation trajectories, which are encoded in the convolutional weights and lie outside the scope of BN calibration.

Unfavorable direction (target domain is more complex). In S1, S5, and S6, the source model has overfitted to limited operating conditions; the distribution mismatch is captured in the frozen convolutional weights, leading to performance degradation when TTA is applied and resulting in RMSE increases of approximately 19, 27, and 13 points, respectively. In S1 and S5, this is accompanied by prediction saturation: under certain seed conditions, the source model's predictions exceed the clipping threshold of 125, causing the outputs of all methods to converge to a constant. Additionally, the ORA baseline fails to outperform SRC in S1, S5, and S6, confirming

that the performance bottleneck stems from the limited adaptability of the BN affine parameters rather than from the optimization signal quality.

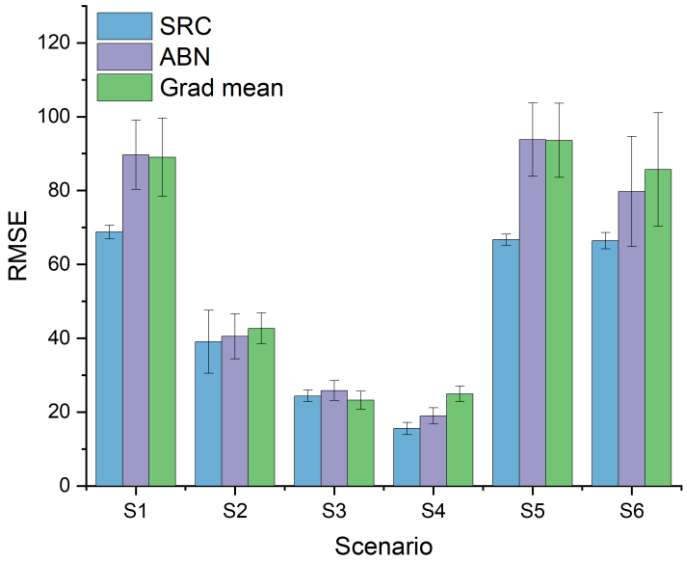

Figure 3. Grouped bar chart of RMSE across six transfer scenarios under Scope-BN. Each group compares the SRC baseline, ABN, and the mean of the four gradient-based methods (VMN, VMN-S, PSL, FAL); error bars denote the standard deviation across five seeds. S3 is the only scenario in which the gradient-based methods reduce RMSE below the SRC baseline.

Fig. 3 presents a grouped visualization of these results. S3 is the only scenario where the gradient-based methods achieve lower RMSE than the SRC baseline. In contrast, in S1 and S5 the TTA bars substantially exceed those of SRC, highlighting clear direction dependence. Furthermore, ABN aligns closely with the mean of the gradient-based methods in most scenarios; however, in S2, S4, and S6 the ABN bar is noticeably lower than this mean. The relative advantage of ABN is discussed in detail in Section 3.5.

3.3. Functional equivalence among gradient-based methods (RQ2)

A notable finding from Table 2 is that the four gradient-based TTA methods (VMN, VMN-S, PSL, and FAL) exhibit nearly identical RMSE values across all scenarios. Despite employing different loss functions, the output predictions are indistinguishable. This trend is consistent across all six scenarios and across five random seeds, suggesting a robust phenomenon.

To quantify this convergence, we compute the standard deviation σ of RMSE across the four gradient-based methods for each scenario–seed combination. Fig. 4 illustrates a heatmap of the mean σ across scenarios under both parameter scopes.

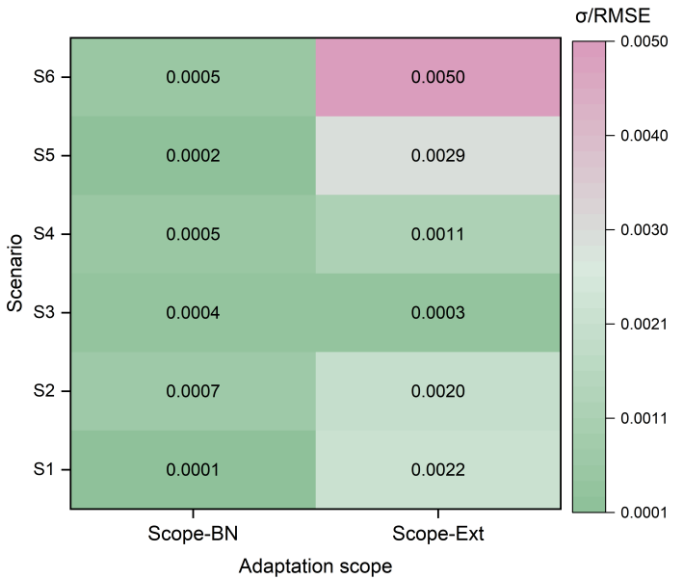

Figure 4. Heatmap of inter-method divergence (σ of RMSE among four gradient-based methods) across scenarios under Scope-BN and Scope-Ext. Under Scope-BN, σ does not exceed 0.0007 in any scenario, indicating that all four methods produce virtually identical adaptation outcomes.

Under Scope-BN, the overall mean σ, excluding prediction-saturated samples, is only 0.0004. In S1 and S5, σ drops below 0.0002; in S2, S3, S4, and S6, σ ranges from 0.0004 to 0.0007. Notably, when applied to the same set of adaptable parameters, the four loss functions yield absolute RMSE differences that do not exceed 0.002 in any scenario, well below the single-seed noise level and negligible compared with the inter-seed variance of 1–18 RMSE points.

This phenomenon is termed BAB. When adaptation is limited to BN affine parameters, only approximately 3,072 of the 613K model parameters can be adjusted. This constraint compresses the gradient directions of the different loss functions into a similar effective update direction, rendering the choice of loss function functionally irrelevant. Fig. 5 conceptually illustrates this bottleneck effect.

The behavior of the stabilization mechanisms in VMN-S further corroborates the BAB mechanism. Parameter anchoring regularization and adaptive gating are designed to constrain and regulate the adaptation process. However, since VMN and VMN-S yielded identical outcomes, this suggests that these

mechanisms lack operational space under the bottleneck condition, as unconstrained updates are negligibly small, leaving no room for stabilization mechanisms to have an effect.

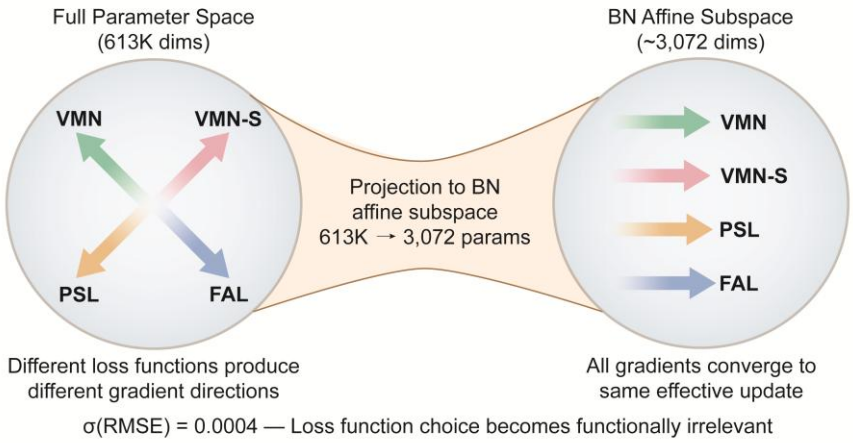

Figure 5. Conceptual illustration of the BN affine bottleneck (BAB). Left: in the full 613K-dimensional parameter space, four loss functions (VMN, VMN-S, PSL, FAL) produce diverse gradient directions. Right: upon projection to the ~3,072-dimensional BN affine subspace, all gradients converge to essentially the same effective update direction, rendering loss function choice functionally irrelevant (σ(RMSE) = 0.0004).

Under Scope-Ext, the single-step σ is 0.0022, comparable to Scope-BN's 0.0004. This seemingly contradictory observation reflects a gradient dilution effect. With gradients distributed over ~131K parameters instead of 3,072, the per-parameter update magnitude decreases, thereby reducing inter-method divergence. The boundary conditions of this effect are explored in the following multi-step ablation experiments.

3.4. Multi-step adaptation ablation study (RQ2)

The BAB hypothesis posits that the functional equivalence of the four gradient-based methods stems from the constraints of parameter-space dimensionality rather than from similar optimization landscapes of the loss functions. If this hypothesis holds, simultaneously expanding the parameter space and increasing the number of adaptation steps should disrupt this equivalence, thereby increasing inter-method divergence. To test this prediction, multi-step ablation experiments were conducted using step counts of {1, 5, 10} across two parameter scopes, three representative scenarios (S2, S3, and S6), and two random seeds (42, 789). Fig. 6 shows the relationship between σ, adaptation steps, and parameter scope.

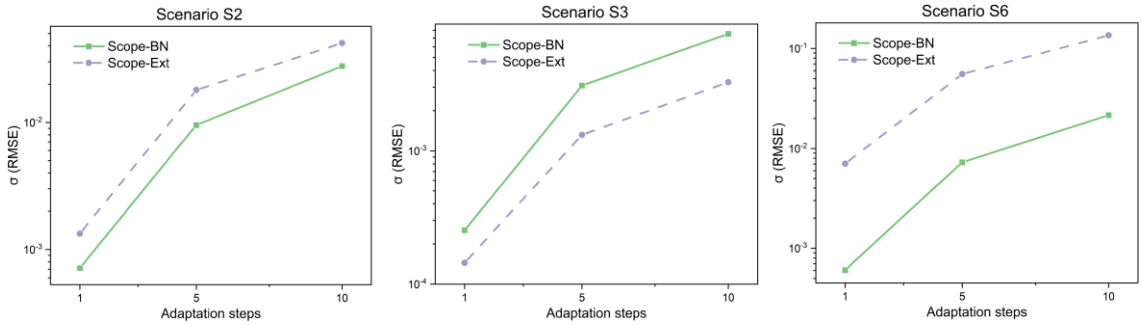

Figure 6. Method divergence (σ) as a function of adaptation steps under Scope-BN and Scope-Ext. Under Scope-BN, σ remains below 0.03, confirming a robust bottleneck. Under Scope-Ext, σ in S6 rises from 0.009 at a single step to approximately 0.16 at 10 steps, an 18-fold increase.

Under Scope-BN, even with 10 adaptation steps, the mean σ across all three scenarios remains below 0.03, confirming the persistence of the bottleneck despite extended adaptation. In contrast, under Scope-Ext, a significant increase is observed: in S6, σ rises from 0.009 at a single step to approximately 0.16 at 10 steps, an 18-fold increase. Similar growth is evident in scenarios S2 and S3 under Scope-Ext, although to a lesser extent.

These findings confirm that the BAB phenomenon arises

from constraints in parameter-space dimensionality rather than from intrinsic similarities among the loss functions. Under Scope-BN, the limited set of approximately 3,072 parameters provides insufficient degrees of freedom for different loss functions to explore distinct update directions, thereby preserving the bottleneck regardless of the number of adaptation steps. In contrast, under Scope-Ext, the expanded parameter space (~131K parameters) allows gradient trajectories to diverge; however, this divergence becomes apparent only after multiple adaptation steps accumulate directional differences across the four loss functions.

3.5. Performance comparison between ABN and gradient-based methods (RQ3)

Having established the functional equivalence of the four gradient-based methods, this study further examines their performance relative to the gradient-free ABN baseline. Fig. 7 presents a direct comparison of the mean RMSE differences between ABN and the four gradient-based methods under Scope-BN.

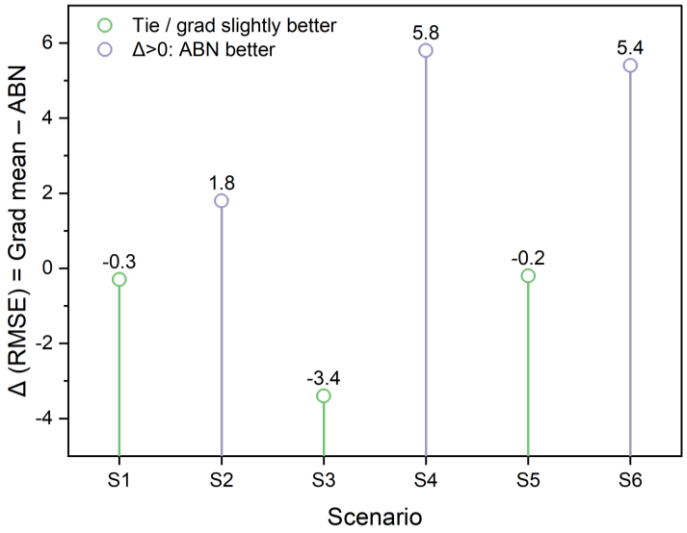

Figure 7. RMSE difference between gradient-based TTA methods and ABN under Scope-BN. Positive values indicate ABN outperforms gradient-based methods. ABN leads by 1.8, 5.8, and 5.4 RMSE points in S2, S4, and S6 respectively; only in S3 do gradient-based methods outperform ABN by approximately 3.4 RMSE points.

In three of the six scenarios—S2, S4, and S6—ABN significantly outperforms the gradient-based methods, with differences of 1.8 in S2, 5.8 in S4, and 5.4 in S6. In S1 and S5, the gradient-based methods marginally surpass ABN by approximately 0.2–0.3 RMSE points, a difference that falls within the inter-seed standard deviation and lacks practical significance. Notably, in S3, the gradient-based methods achieve an RMSE of 23.0, outperforming ABN's 26.4 by approximately 3.4 RMSE points. This difference is attributable to S3, which features a more complex source domain and presents a favorable TTA direction; under these conditions, gradient-based methods achieve improvements that exceed ABN's coarse-grained statistics replacement via targeted optimization. Overall, ABN matches or outperforms gradient-based methods in five of the six scenarios, demonstrating less degradation under unfavorable conditions and offering the best overall risk–benefit trade-off.

The performance disadvantage of the gradient-based methods in S2, S4, and S6 can be explained from two perspectives: First, a systematic misalignment exists between the gradient directions of the proxy loss functions (VMN, pseudo-label MSE, and CORAL distance) and the direction that minimizes the true RUL prediction error. The low-dimensional space of approximately 3,072 parameters does not afford sufficient degrees of freedom for the gradient search to circumvent this misalignment. Second, gradient updates directly modify the BN affine parameters (scale factors and shift parameters) optimized during source-domain training. Under unfavorable or neutral shift conditions, these modifications can push the parameters away from their optimal source-domain configurations. In contrast, ABN avoids the systematic bias introduced by proxy objectives under unfavorable shift conditions by updating only the running statistics while retaining the source-domain affine configuration.

3.6. Statistical tests

To avoid over-interpreting small numerical differences, paired Wilcoxon signed-rank tests were conducted across all 30 scenario–seed combinations under Scope-BN, covering all six pairwise comparisons among the four gradient-based methods (VMN, VMN-S, PSL, and FAL). Five of the six comparisons are not significant ($p > 0.05$). The only significant comparison, PSL vs. FAL ($p = 0.046$), exhibits a mean absolute RMSE difference of 0.0004, well below the inter-seed variance (1–18 RMSE points). These outcomes confirm the functional equivalence among gradient-based methods: under the defined

parameter constraints, gradient updates driven by varying loss functions yield outputs that are virtually indistinguishable in practice.

3.7. Backbone generalization

To examine whether the core findings depend on the specific backbone architecture, experiments for three representative scenarios were repeated using an LSTM backbone. The LSTM architecture contains two BN layers with approximately 512 affine parameters, substantially fewer than the 3,072 parameters in the TCN. The training configuration matches that of the TCN, consisting of 50 epochs and early stopping with a patience of 10, and was independently trained under four random seeds. The results are summarized in Table 3.

Table 3. RMSE results (↓) for all methods with the LSTM backbone (mean ± std over 4 seeds).

Scenario	SRC	ABN	VMN	PSL	σ(VMN, PSL)
S2	31.90±3.91	31.27±3.96	29.36±3.97	29.37±3.97	0.002
S3	22.56±3.47	19.66±1.46	19.61±1.42	19.62±1.42	0.002
S6	68.19±3.74	67.63±4.82	98.09±5.52	98.07±5.55	0.032

Abbreviations: SRC, source-only; ABN, adaptive batch normalization; VMN, variance minimization; PSL, pseudo-label self-training; σ(VMN, PSL), mean absolute RMSE difference between VMN and PSL; RMSE, root mean square error; LSTM, long short-term memory.

Table 4. Cross-domain RMSE results (↓) on the N-CMAPSS dataset (mean ± std).

Scenario	Source	Target	*n*	SRC	ABN	VMN	PSL	σ(VMN, PSL)
N1	DS01	DS03	30	**5.47±0.44**	6.87±0.39	52.35±1.89	52.35±1.89	0.006
N2	DS03	DS04	20	65.75±2.40	65.33±1.57	**38.78±1.65**	**38.78±1.65**	0.003

Abbreviations: *n*, number of test samples per seed; σ(VMN, PSL), mean absolute RMSE difference between the two gradient-based methods. Bold indicates the best method for each scenario. SRC, source-only; ABN, adaptive batch normalization; VMN, variance minimization; PSL, pseudo-label self-training; RMSE, root mean square error.

Both principal findings generalize to the LSTM backbone. Functional equivalence remains evident, with RMSE differences between VMN and PSL not exceeding 0.032 across all scenarios, indicating that the BAB is not architecture-specific and persists even with only ~512 BN affine parameters. Direction dependence is also consistent: TTA improves performance in S3 (RMSE: 22.56 → 19.61) but degrades performance in S6 (68.19 → 98.09).

ABN demonstrates a pronounced robustness advantage for the LSTM backbone. In S6, ABN (67.63) outperforms the gradient-based methods (98.09) by more than 30 RMSE points. In contrast, gradient-based methods outperform ABN by only 1.9 RMSE points in S2 and yield nearly identical results in S3. These findings support the recommendation of ABN as a default strategy across different architectures.

3.8. Cross-dataset generalization

To evaluate generalizability beyond C-MAPSS, additional experiments were conducted using the NASA N-CMAPSS dataset 30, a next-generation turbofan simulation dataset with a substantially larger data volume, 14 sensor channels, and multiple flight conditions. Two cross-domain scenarios were defined: N1 (DS01 → DS03), where both subsets share the same flight-class distribution ($F_c \in \{1,2,3\}$), with domain differences arising from variations among individual engines; and N2 (DS03 → DS04), where DS04 lacks flight class $F_c = 1$, resulting in a more pronounced distribution shift. The backbone architecture and training configuration follow those used for the TCN experiments, with four random seeds. The results are presented in Table 4.

Both core findings are consistently validated using the N-CMAPSS dataset. First, the BAB is evident: the RMSE difference between VMN and PSL is only 0.006 and 0.003 in N1 and N2, respectively, aligning with the functional equivalence observed in C-MAPSS. Second, the direction dependence of TTA effectiveness is also observed: in N1, the source model demonstrates robust predictive performance, achieving an RMSE of only 5.47, whereas all TTA methods result in performance degradation, with gradient-based methods increasing RMSE to 52.35, indicating catastrophic failure.

In N2, gradient-based methods reduce RMSE from 65.75 to 38.78, an approximately 41% reduction. This improvement reflects genuine feature recalibration rather than truncation artifacts, as the MAE/RMSE ratio simultaneously drops from 0.99 to 0.85–0.94, indicating that predictions spread toward the true RUL values rather than collapsing onto the clipping threshold. ABN achieves an RMSE of 65.33, comparable to that of SRC and consistent with its conservative behavior observed

on C-MAPSS.

Overall, the BN affine bottleneck and the direction dependence of TTA effectiveness are consistently observed across two turbofan degradation datasets, C-MAPSS and N-CMAPSS, and two backbone architectures, TCN and LSTM. This suggests that neither finding is specific to a single dataset or architecture. Whether these findings generalize to non-turbofan degradation processes remains to be verified, as discussed in Section 3.9.

3.9. Limitations

This study has several limitations. First, the experimental evaluation is limited to turbofan engine degradation datasets; generalization to other types of degradation (e.g., bearings or batteries) requires further investigation. Second, the relatively small sample sizes in C-MAPSS (100–260 engines per subset) and N-CMAPSS (20–30 test samples) may constrain the statistical power of inter-method comparisons. Third, as noted in Section 1, the analysis is confined to architectures with BN layers, with adaptation restricted to BN-related parameters. Extending this analysis to normalization-free or layer-normalized architectures, such as Transformers, is left for future work. Fourth, the impact of continual adaptation over extended deployment periods, particularly under progressive distribution drift, was not examined.

4. Discussion

This section examines the implications of our findings for developing maintenance strategies and guiding model deployment decisions in predictive maintenance.

4.1. Deployment decision guidelines

Based on our experimental findings on direction dependence and BAB, the following decision workflow is recommended for cross-domain pretrained RUL model deployment:

Step 1: Assess the relative complexity of the training and deployment environments—specifically, whether the training data encompass a broader range of operating conditions than the deployment environment.

Step 2: If the training environment is more complex (i.e., deploying a multi-condition model to a single-condition scenario), apply the ABN directly, as it requires only a single forward pass, with no gradient computation or hyperparameter selection.

Step 3: If the deployment environment is of comparable or greater complexity than the training environment, or if the relationship is unclear, deploy the source model directly and prioritize collecting labeled data from the target environment for supervised retraining.

Step 4: Consider gradient-based TTA methods only when the shift direction is favorable (i.e., when the training environment is more complex than the deployment environment), and the model architecture supports unlocking additional adaptable parameters beyond normalization layers.

The four-step decision workflow is summarized in Fig. 8.

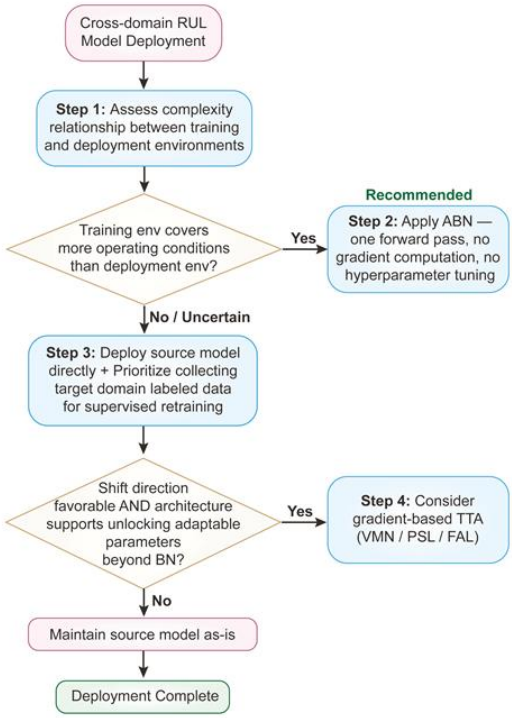

Figure 8. Deployment decision flowchart for cross-domain RUL models. Starting from assessing the complexity relationship between training and deployment environments, three paths are recommended based on shift direction and adaptable parameter scope: apply ABN directly when the source domain is more complex (recommended), deploy the source model and prioritize collecting labeled data for supervised retraining when the shift is unfavorable or uncertain, and consider gradient-based TTA only when the shift direction is favorable and the architecture supports extending adaptable parameters beyond normalization layers.

4.2. Theoretical implications of the BN affine bottleneck

BAB indicates that, within the BN-only adaptation

configuration, performance comparisons among different TTA methods may lack practical relevance. Our ablation experiments confirm that meaningful inter-method divergence requires satisfying two conditions: a sufficiently large adaptable parameter space and a sufficient number of adaptation steps. This finding suggests that designing more sophisticated loss functions within a constrained parameter space may not yield the expected performance gains; resources would be better allocated toward expanding the adaptation scope or establishing effective parameter selection strategies.

4.3. Complementary relationship between test-time adaptation and domain adaptation

TTA is not a substitute for DA, as they operate under different conditions. DA leverages source- and target-domain data during training and achieves cross-domain generalization by explicitly aligning their feature distributions; its adaptation is not limited to normalization-layer parameters. The concept of BAB exemplifies this: limiting adaptation to a small number of parameters physically constrains its capacity, whereas DA can adjust the full set of network parameters, thereby circumventing this limitation. Consequently, DA is preferable when source- and target-domain data are simultaneously accessible.

DA is not included in the experimental comparison because the underlying assumptions differ fundamentally. DA methods access target-domain data during training, whereas TTA methods operate after deployment without such access. Direct numerical comparisons under these unequal information settings would therefore be neither valid nor informative. The practical advantage of TTA lies in its minimal requirements: it does not require access to source data or model retraining. This makes it particularly suitable for scenarios involving intellectual property constraints, regulatory restrictions, or time-critical deployments, where DA is infeasible.

4.4. Maintenance safety implications of the prediction saturation effect

The prediction saturation effect observed in S1 and S5 presents significant maintenance safety implications: when source model predictions collapse to the RUL upper clipping threshold, the maintenance system continuously receives erroneous signals indicating that the equipment is in good health, potentially delaying critical maintenance interventions. Under these conditions, no TTA method, including the Oracle baseline, can correct this failure. Consequently, maintenance systems deployed across domains should incorporate prediction anomaly detection mechanisms that trigger alerts when model outputs persistently approach the clipping threshold, prompting operations personnel to retrain the model using target environment data or revert to a physics-based backup prediction model.

5. Conclusion

This study constructs a unified evaluation framework for TTA strategies in cross-domain RUL prediction and systematically examines the performance of four TTA strategies on the C-MAPSS and N-CMAPSS datasets with TCN and LSTM architectures. The main findings are as follows: (1) The effectiveness of TTA depends on the direction of domain shift, yielding consistent improvement only when the source domain exhibits greater operating-condition diversity than the target domain. Operating-condition differences directly influence the feature statistics calibrated by BN layers, whereas fault-mode differences are encoded in convolutional weights and cannot be leveraged by BN adaptation. (2) When adaptation is limited to BN affine parameters, all four gradient-based methods produce functionally equivalent outputs (BAB). Parameter-space expansion and multi-step ablation experiments have confirmed this causal relationship. (3) Gradient-free ABN matches or outperforms gradient-based methods in five of the six evaluated scenarios, and its performance degradation under unfavorable shift directions is either comparable to or less than that of gradient-based methods, yielding the best overall risk–benefit trade-off.

The findings of this study have been validated on turbofan engine degradation datasets; however, additional research is required to assess their generalizability to other degradation types, such as bearings and batteries. Furthermore, future investigations should consider whether BAB has an analogous counterpart in architectures employing layer normalization.

References

1. Kang Z, Catal C, Tekinerdogan B. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. *Sensors* 2021; 21(3): 932. https://doi.org/10.3390/s21030932.
2. Si XS, Wang W, Hu CH, Zhou DH. Remaining useful life estimation—A review on the statistical data driven approaches. *European Journal of Operational Research* 2011; 213(1): 1-14. https://doi.org/10.1016/j.ejor.2010.11.018.
3. Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. *Proceedings of the IEEE International Conference on Prognostics and Health Management (ICPHM);* 2017. p. 88-95. https://doi.org/10.1109/ICPHM.2017.7998311.
4. Zhang J, Wang P, Yan R, Gao RX. Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems* 2018; 48: 78-86. https://doi.org/10.1016/j.jmsy.2018.05.011.
5. Li X, Ding Q, Sun JQ. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety* 2018; 172: 1-11. https://doi.org/10.1016/j.ress.2017.11.021.
6. Lea C, Flynn MD, Vidal R, Reiter A, Hager GD. Temporal convolutional networks for action segmentation and detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR);* 2017. p. 1003-1012. https://doi.org/10.1109/CVPR.2017.113.
7. Wang S, Zhang C, Lv D, Zhao W. Temporal convolutional network with attention mechanism for bearing remaining useful life prediction. In: Zhang H, Ji Y, Liu T, Sun X, Ball AD, editors. *Proceedings of TEPEN 2022. Mechanisms and Machine Science*, vol 129. Cham: Springer; 2023. p. 391-400. https://doi.org/10.1007/978-3-031-26193-0_33.
8. Mo Y, Wu Q, Li X, Huang B. Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit. *Journal of Intelligent Manufacturing* 2021; 32(7): 1997-2006. https://doi.org/10.1007/s10845-021-01750-x.
9. Li X, Song K, Shi J. Degradation generation and prediction based on machine learning methods: A comparative study. *Eksploatacja i Niezawodnosc – Maintenance and Reliability* 2025; 27(1): 192168. https://doi.org/10.17531/ein/192168.
10. Yu S, Wu Z, Zhu X, Pecht M. A domain adaptive convolutional LSTM model for prognostic remaining useful life estimation under variant conditions. *Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Paris)*; 2019. p. 130-137. https://doi.org/10.1109/PHM-Paris.2019.00030.
11. Sun B, Feng J, Saenko K. Return of frustratingly easy domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence* 2016; 30(1): 2058-2065. https://doi.org/10.1609/aaai.v30i1.10306.
12. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 2016; 17(59): 1-35. https://jmlr.org/papers/v17/15-239.html.
13. da Costa PRDO, Akçay A, Zhang Y, Kaymak U. Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety* 2020; 195: 106682. https://doi.org/10.1016/j.ress.2019.106682.
14. Zhang A, Wang H, Li S, Cui Y, Liu Z, Yang G, Hu J. Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Applied Sciences* 2018; 8(12): 2416. https://doi.org/10.3390/app8122416.
15. Lv Y, Zhou N, Wen Z, Shen Z, Chen A. Remaining useful life prediction based on multisource domain transfer and unsupervised alignment. *Eksploatacja i Niezawodnosc – Maintenance and Reliability* 2025; 27(2): 194116. https://doi.org/10.17531/ein/194116.
16. Nejjar I, Geissmann F, Zhao M, Taal C, Fink O. Domain adaptation via alignment of operation profile for remaining useful lifetime prediction. *Reliability Engineering & System Safety* 2024; 242: 109718. https://doi.org/10.1016/j.ress.2023.109718.
17. Hou Y, Ragab M, Wu M, Kwoh CK, Li X, Chen Z. Target-specific adaptation and consistent degradation alignment for cross-domain remaining useful life prediction. *IEEE Transactions on Automation Science and Engineering* 2026; 23: 3596-3606. https://doi.org/10.1109/TASE.2025.3590839.
18. Dong X, Zhang C, Liu H, Wang D, Wang T. A multi-constrained domain adaptation network for remaining useful life prediction of bearings. *Mechanical Systems and Signal Processing* 2024; 206: 110900. https://doi.org/10.1016/j.ymssp.2023.110900.
19. Liang J, He R, Tan T. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision* 2025; 133(1): 31-64. https://doi.org/10.1007/s11263-024-02181-w.
20. Li Y, Wang N, Shi J, Hou X, Liu J. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition* 2018; 80: 109-117. https://doi.org/10.1016/j.patcog.2018.03.005.

21. Wang D, Shelhamer E, Liu S, Olshausen B, Darrell T. Tent: Fully test-time adaptation by entropy minimization. *International Conference on Learning Representations (ICLR)*. 2021. https://openreview.net/forum?id=uXl3bZLkr3c.
22. Niu S, Wu J, Zhang Y, Chen Y, Zheng S, Zhao P, Tan M. Efficient test-time model adaptation without forgetting. *Proceedings of the 39th International Conference on Machine Learning (ICML);* 2022. p. 16888-16905. https://proceedings.mlr.press/v162/niu22a.html.
23. Niu S, Wu J, Zhang Y, Wen Z, Chen Y, Zhao P, Tan M. Towards stable test-time adaptation in dynamic wild world. *International Conference on Learning Representations (ICLR)*. 2023. https://openreview.net/forum?id=g2YraF75Tj.
24. Iwasawa Y, Matsuo Y. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems (NeurIPS)* 2021; 34: 2427-2440. https://proceedings.neurips.cc/paper/2021/hash/1415fe9fea0fa1e45dddcff5682239a0-Abstract.html.
25. Wang Q, Fink O, Van Gool L, Dai D. Continual test-time domain adaptation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2022. p. 7201-7211. https://doi.org/10.1109/CVPR52688.2022.00706.
26. Sun Y, Wang X, Liu Z, Miller J, Efros AA, Hardt M. Test-time training with self-supervision for generalization under distribution shifts. *Proceedings of the 37th International Conference on Machine Learning (ICML)*; 2020. p. 9229-9248. https://proceedings.mlr.press/v119/sun20b.html.
27. Wu Y, Chi Z, Wang Y, Plataniotis KN, Feng S. Test-time domain adaptation by learning domain-aware batch normalization. *Proceedings of the AAAI Conference on Artificial Intelligence* 2024; 38(14): 15961-15969. https://doi.org/10.1609/aaai.v38i14.29527.
28. Kim HG, Kim S, Mok J, Yoon S. Battling the non-stationarity in time series forecasting via test-time adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence* 2025; 39(17): 17868-17876. https://doi.org/10.1609/aaai.v39i17.33965.
29. Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. *International Conference on Prognostics and Health Management (PHM)*; 2008. p. 1-9. https://doi.org/10.1109/PHM.2008.4711414.
30. Arias Chao M, Kulkarni C, Goebel K, Fink O. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data* 2021; 6(1): 5. https://doi.org/10.3390/data6010005.

Appendix A. Detailed Formulations of TTA Strategies

This appendix provides the complete mathematical formulations for the three TTA strategies whose derivations are summarized in the main text (Sections 2.3.3–2.3.5).

A.1. Stabilized variance minimization (VMN-S)

VMN-S extends VMN by incorporating two stabilization mechanisms: parameter anchoring regularization and adaptive gating.

Parameter anchoring regularization. A quadratic penalty term is added to the TTA objective to prevent excessive deviation of the adapted BN affine parameters from their initial source-domain values obtained after training:

$$\mathcal{L}_{\text{anchor}} = \lambda \left\| \theta_{BN} - \theta_{BN}^{(0)} \right\|^2 \tag{A.1}$$

where θ_{BN} denotes the current BN affine parameter vector, $\theta_{BN}{}^{(0)}$ represents the initial parameter vector, and λ is the regularization coefficient (λ = 0.1 by default). The total VMN-S loss is defined as:

$$\mathcal{L}_{\text{VMN-S}} = \mathcal{L}_{\text{VMN}} + \mathcal{L}_{\text{anchor}} \tag{A.2}$$

Adaptive gating. A per-sample gating coefficient is computed based on the change in predictive uncertainty before and after adaptation to determine whether the adapted prediction should be accepted:

$$g_i = \sigma\left(\hat{\sigma}^2_{i,\text{before}} - \hat{\sigma}^2_{i,\text{after}}\right) \tag{A.3}$$

where $\sigma(\cdot)$ denotes the sigmoid function, and $\hat{\sigma}^2_{i,\text{before}}$ and $\hat{\sigma}^2_{i,\text{after}}$ are the predicted variances for the i-th sample before and after adaptation, respectively. The final prediction is obtained as a convex combination of the pre- and post-adaptation outputs:

$$\hat{\mu}_{i,\text{final}} = g_i \cdot \hat{\mu}_{i,\text{after}} + (1 - g_i) \cdot \hat{\mu}_{i,\text{before}} \tag{A.4}$$

If adaptation increases uncertainty (i.e., $\hat{\sigma}^2_{\text{after}} > \hat{\sigma}^2_{\text{before}}$), then g_i approaches zero, and the model defaults to the source-domain prediction.

A.2. Pseudo-label self-training (PSL)

PSL estimates prediction confidence using Monte Carlo (MC) dropout and selects high-confidence samples for pseudo-label–based self-training. During inference, BN layers remain in evaluation mode, while dropout layers are activated. For each target-domain sample, M stochastic forward passes are performed (M = 5 by default), yielding M predictions $\{\hat{\mu}_i^{(1)}, \ldots, \hat{\mu}_i^{(M)}\}$. The predictive mean and epistemic variance are computed as

$$\overline{\mu}_i = \frac{1}{M}\sum_{m=1}^{M} \hat{\mu}_i^{(m)} \tag{A.5}$$

$$\nu_i = \frac{1}{M}\sum_{m=1}^{M} \left(\hat{\mu}_i^{(m)} - \overline{\mu}_i\right)^2 \tag{A.6}$$

where $\bar{\mu}_i$ and ν_i denote the MC prediction mean and epistemic uncertainty estimate for the i-th sample, respectively. Samples are ranked in ascending order of ν_i, and the top K% with the lowest variance (K = *30* by default) are selected as reliable samples. Their corresponding means $\bar{\mu}_i$ serve as pseudo-labels. The BN affine parameters are then updated by minimizing the mean squared error:

$$\mathcal{L}_{\text{PSL}} = \frac{1}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} \left(f_\theta(x_i) - \overline{\mu}_i\right)^2 \tag{A.7}$$

where $\mathcal{R}$ denotes the set of selected reliable samples and $f_\theta(x_i)$ is the current model prediction for sample x_i.

A.3. Feature alignment (FAL)

FAL reduces distributional discrepancies between source and target domains by aligning feature statistics at the inputs of BN layers. After completing source-domain training, the mean and covariance matrices of the input features at each BN layer are computed offline and stored as reference statistics. During testing, the corresponding statistics are computed for each target-domain mini-batch, and the CORAL distance is minimized:

$$\mathcal{L}_{\text{FAL}} = \sum_{l=1}^{L} \frac{1}{4d_l^2} \left\| \boldsymbol{C}_s^{(l)} - \boldsymbol{C}_t^{(l)} \right\|_F^2 \tag{A.8}$$

where L is the total number of BN layers (L = 12 in this study), d_l is the feature dimensionality of the l-th layer, $\boldsymbol{C}_s^{(l)}$ and $\boldsymbol{C}_t^{(l)}$ denote the covariance matrices of source- and target-domain features at the l-th layer, respectively, and $\|\cdot\|_F$ represents the Frobenius norm.