machines

MDPI

Article

Adaptive Transfer Consistency Network for Multilevel Cross-Domain Fault Diagnosis of Wind Turbine Bearings

Zhezheng Zhong and Fanghua Ye *

Jiangxi Province Key Laboratory of Light Alloy, School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China; zhongzz@email.ncu.edu.cn
* Correspondence: yefh@ncu.edu.cn

Abstract

Wind turbine bearings are critical drivetrain components, and their failures can compromise operational safety, cause turbine shutdowns, and increase maintenance costs. In practical applications, wind turbine bearing fault diagnosis often involves two coexisting domain shifts: variations in rotational speed and differences between machines. Unsupervised domain adaptation (UDA) provides an effective means of transferring diagnostic knowledge from labeled source domains to unlabeled target domains. However, existing UDA studies typically address one transfer scenario at a time. Applying these methods to cross-speed and cross-machine transfer tasks remains challenging. In particular, global distribution alignment, fault-class correspondence, target class information, and prediction consistency must be coordinated. To address these challenges, we propose an Adaptive Transfer Consistency Network (ATCN). First, ATCN uses Progressive Multi-Level Adaptive Alignment (PMLA) to progressively strengthen global distribution alignment and learn a weighted combination of Euclidean and cosine distances for same-class centroid alignment, thereby reducing interdomain discrepancies while preserving fault-class correspondence. It then uses Progressive Balanced Knowledge Mining (PBKM) to delay target pseudo-label supervision during a warm-up stage and combines dynamic confidence thresholding with class-wise Top-K selection to obtain balanced target pseudo-labels for classification supervision and centroid computation. Finally, it employs Output-Level Consistency Regularization (OCR) to minimize target entropy and constrain diagnostic outputs under input perturbations. These components are jointly optimized within a unified objective. Across twelve transfer tasks from the public HIT-SM bearing dataset, using data collected on the MFS and Self test rigs at three rotational speeds, ATCN achieves mean cross-speed and cross-machine accuracies of 99.36% and 84.33%, respectively, outperforming the corresponding strongest baseline means by 8.65 and 23.38 percentage points, demonstrating its effectiveness on the investigated transfer tasks under laboratory conditions.

Keywords: wind turbine bearings; unsupervised domain adaptation; multilevel cross-domain fault diagnosis; pseudo-labels; consistency regularization

Check for updates

Academic Editor: Maciej Sułowicz

Received: 22 October 2026
Revised: 14 November 2026
Accepted: 16 November 2026
Published: 18 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Bearings are critical components of wind turbine drivetrains, and their faults can cause turbine shutdowns and increase maintenance costs [1]. Over the past five years, a field-data study estimated that approximately 10% of main bearings in the studied wind turbine population would have been replaced by 10.5 years of operation [2]. Main bearings experience complex contact loads that affect their reliability [3]. Accurate bearing fault diagnosis is therefore important for the safe and reliable operation of wind turbines [4–6].

Traditional diagnosis methods extract fault information through signal processing and manually designed features. Chen et al. [7] used empirical wavelet transforms to identify weak and compound faults in wind turbine generator bearings. Zhang et al. [8] combined component selection with singular value decomposition to improve bearing signal denoising. Wang et al. [9] investigated bearing damage identification using signal decomposition. For low-speed bearings, Ma et al. [10] used acoustic emission signals to detect and locate main bearing damage. Wu et al. [11] identified pitch bearing cracks from changes in outer-ring stress. Jiang et al. [12] extracted wavelet packet statistical features and fused multisource signals through canonical correlation analysis to diagnose wind turbine gearbox bearing faults. Although these methods have achieved some success, their feature extraction and analysis generally rely on expert experience and require substantial time and effort.

Deep learning automatically learns discriminative fault features, reducing the effort required for manual feature design and analysis. Bach-Andersen et al. [13] used convolutional networks to analyze multi-sensor signals from operating wind turbines. Chang et al. [14] extracted wind turbine bearing features using parallel multiscale convolutions. Zhang and Zhang [15] combined temporal convolutions with Bayesian parameter selection for wind turbine pitch bearing fault detection. However, for wind turbines, models trained on labeled data from a fixed domain may lose diagnostic accuracy when deployed at different speeds or on different machines [16–18]. Wind turbine speeds and operating conditions vary, while drivetrain structure and loading also differ between machines. The resulting distribution shifts can reduce diagnostic performance [19].

Unsupervised domain adaptation (UDA) uses labeled source data and unlabeled target data to reduce the effects of distribution shifts on diagnosis. Liang et al. [20] combined deformable residual convolutions with subdomain alignment for unsupervised wind turbine bearing diagnosis under time-varying speeds. Jin et al. [21] combined residual networks with domain-adversarial learning to investigate cross-condition diagnosis in actual wind power systems. Wang et al. [22] transferred synthetic faults generated from expert knowledge to unlabeled real wind turbine bearing data, addressing class imbalance through conditional adaptation. Su et al. [23] combined deep contrastive convolutions with the Wasserstein distance to investigate wind turbine bearing fault localization in an unlabeled target domain. The comparison methods used in this study also represent several approaches to learning transferable or discriminative fault features. CNN-C combines convolutional feature learning with center loss and has been investigated for diagnosis under unseen operating conditions [24]. Deep CORAL (D-CORAL) aligns second-order feature statistics across domains [25]. DDTLN combines distribution alignment with discriminative feature learning for cross-machine fault diagnosis [26]. DXDA uses a two-stage unilateral alignment strategy to address missing classes in the target domain [27]. PKMTCL integrates prior diagnostic knowledge, prototype-based alignment, and multi-task learning for few-shot aero-engine bearing diagnosis [28]. DANN uses adversarial domain classification and gradient reversal to learn domain-invariant features [29]. These methods provide complementary reference points for evaluating ATCN under the transfer tasks considered here.

In practical wind farms, bearing diagnosis must address both changes in rotational speed on the same machine and differences between machines when diagnostic knowledge is transferred to another turbine. To meet this need, the same Adaptive Transfer Consistency Network (ATCN) architecture and training objective are used, with a separate model trained for each source–target task.

At new operating speeds or on other turbines, bearing monitoring requires transferring fault diagnosis knowledge learned from labeled source-domain vibration data to an

unlabeled target domain. This requirement poses the following challenges for conventional UDA methods.

First, speed changes alter signal distributions on the same machine, while machine changes also introduce structural and loading differences. In the presence of these different sources of domain discrepancy, a unified framework must balance interdomain distribution matching with class discrimination. Related methods have explored global or class-wise alignment and confidence-aware class matching [30–33], as summarized in Table 1. However, a fixed alignment strength selected for one scenario may not adapt well to different domain discrepancies: weak constraints can leave domain differences unresolved, whereas strong constraints introduced before discriminative features have formed can distort class structure.

Table 1. Comparison of adaptation mechanisms in recent bearing fault diagnosis studies.

Method (Year)	Global Alignment (G)	Class-Wise Alignment (C)	Target Reliability Control (R)	Perturbation Consistency (S)
NG-UDAN [30] (2025)	✓	✓	×	×
CASSA [31] (2025)	×	✓	✓	×
TF-MFA-DA [32] (2026)	✓	✓	×	×
CWAN-BPDR [33] (2026)	×	✓	✓	×
CRG-DA Net [34] (2026)	✓	×	✓	×
DSR-OSDA [35] (2026)	×	✓	✓	✓
ATCN (this work)	✓	✓	✓	✓

G: an explicit class-independent global or marginal source–target alignment term; C: explicit class-conditional or centroid/subdomain source–target alignment; R: explicit filtering, correction, or reliability weighting of target pseudo-labels or class information; S: consistency of diagnostic outputs across perturbed views of the same target sample. ✓ indicates a reported mechanism; × indicates no matching independent mechanism identified in the reported method.

Second, target domains in these different transfer scenarios are unlabeled during adaptation, so target class information must be inferred from model diagnostic outputs and represented by pseudo-labels. Reliability-oriented target-sample selection has also been investigated [31,33,34]. When domain discrepancies make target diagnostic outputs unreliable, early classification errors can propagate through target supervision and class alignment; meanwhile, confidence filtering can favor classes that are easier to classify, resulting in an imbalanced distribution of selected samples.

Finally, the stability of target diagnostic results must also be maintained under different domain discrepancies. Small input perturbations can change diagnostic results near decision boundaries. Diagnostic output consistency across perturbed target views has also been explored [35]. Explicitly constraining target probability differences complements feature alignment and pseudo-label supervision.

We propose the Adaptive Transfer Consistency Network (ATCN) to integrate PMLA, PBKM, and OCR in a unified training objective. The same framework is applied to both cross-speed and cross-machine adaptation, with a separate model trained for each source–target task.

(1) We formulate a multilevel cross-domain UDA setting for bearing fault diagnosis to address domain shifts associated with speed changes and differences in machine structure and loading. This setting considers cross-speed transfer within the same machine and cross-machine transfer at the same speed, representing two distinct adaptation requirements relevant to wind turbine bearing diagnosis.

(2) We propose PMLA to address global distribution differences and class misalignment. PMLA gradually strengthens global distribution constraints and adjusts same-class features using positional and directional differences between class centroids. It reduces interdomain differences while maintaining correspondence between the same fault classes.

(3) We propose PBKM to address unreliable target-domain diagnostic outputs and uneven pseudo-label selection across classes. PBKM delays pseudo-label supervision, adjusts confidence thresholds during training, and limits the number of selected samples per class. The selected pseudo-labeled samples support both target classification supervision and class-centroid alignment.

(4) We propose OCR to address the sensitivity of target diagnostic outputs to local input changes. OCR reduces the entropy of target class probabilities and limits output differences under perturbations, complementing feature alignment and pseudo-label supervision.

The distinctive feature of ATCN is the coordinated use of progressive global and class-centroid alignment, balanced pseudo-label selection, and target output regularization within a unified adaptation framework. In particular, selected pseudo-labeled samples support both target classification and class-centroid alignment, while output regularization constrains diagnostic outputs across the target batch.

The remainder of this paper is organized as follows. Section 2 presents the problem definition, overall ATCN framework, module designs, and optimization objective. Section 3 describes the dataset and experimental settings and presents cross-speed and cross-machine diagnostic results, hyperparameter sensitivity analysis, module ablation studies, and an inference cost comparison. Section 4 summarizes the findings and discusses future work.

2. Materials and Methods

2.1. Problem Definition and Classification Objective

To formally define our proposed multilevel cross-domain setting for wind turbines, let the labeled source dataset be $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where $x_i^s \in \mathcal{X} \subseteq \mathbb{R}^{1\times L}$ denotes the ith source signal segment of length L. Here, $y_i^s \in \mathcal{Y}_s$ is its fault class label, and n_s is the number of source samples. The unlabeled target dataset is $\mathcal{D}_t = \left\{x_j^t\right\}_{j=1}^{n_t}$, where $x_j^t \in \mathcal{X} \subseteq \mathbb{R}^{1\times L}$ denotes the jth target signal segment and n_t is the number of target samples. Both domains share the label space $\mathcal{Y}_s = \mathcal{Y}_t = \mathcal{Y} = \{1, 2, \ldots, C\}$, where C is the number of classes. Mathematical labels start at 1, whereas the implementation codes in Table 2 start at 0. They correspond in class order.

Table 2. Data domains and class settings.

Item	Setting
Test rigs	MFS, Self
Speeds	600, 900, 1200 r/min
Sampling frequency	51.2 kHz
Selected states	Normal, IR5, IR8, OR5
Class codes	0, 1, 2, 3
Sample length	1024 points, approximately 20 ms
Source samples per domain	4 classes × 1280 = 5120
Source training and validation	Stratified split into 4096 and 1024 samples
Target domain	4096 unlabeled adaptation samples; 1024 held-out test samples

For any transfer task, the source and target domains satisfy $\mathcal{P}_s(X) \neq \mathcal{P}_t(X)$. Here, $\mathcal{P}_s(X)$ and $\mathcal{P}_t(X)$ denote the marginal distributions of source and target input signals, respectively. We define two types of transfer tasks. A cross-speed task is denoted by $T_{speed}: D_s^{(r_s)} \rightarrow D_t^{(r_t)}$, with $r_s \neq r_t$, where both domains originate from the same test platform but have different speeds. A cross-machine task is denoted by $T_{device}: D_s^{(d_s,r)} \rightarrow D_t^{(d_t,r)}$, with $d_s \neq d_t$, where the domains originate from different test platforms at the same speed. The distribution shift of a task is expressed as $\delta(T) = D(P_s(X), P_t(X))$, where D is a distribution

discrepancy measure. The two task types correspond to $\delta(T_{\text{speed}})$ and $\delta(T_{\text{device}})$, whose magnitudes vary with the source and target combination. Both use the same training objective but learn separate diagnostic functions for their respective target domains. Here, "multilevel cross-domain" describes the two transfer task families. Within ATCN, PMLA further performs adaptation at the global-distribution and class-centroid levels.

Let $G_\theta(\cdot)$ be the shared feature extractor and $C_\phi(\cdot)$ the classification head, which outputs class logits. The diagnostic function is $f(x) = C_\phi(G_\theta(x))$, where θ and ϕ are their respective learnable parameters. Class probabilities are obtained by applying Softmax to these logits. The objective is to minimize the target classification risk:

$$\min_{\theta,\phi} \mathcal{R}_t(f) = \mathbb{E}_{(x,y)\sim \mathcal{D}_t(X,Y)}[\ell(f(x), y)] \tag{1}$$

where $\ell(\cdot,\cdot)$ denotes the classification loss. The training objective combines source supervision with target adaptation constraints from PMLA, PBKM, and OCR, as detailed in Section 2.7.

2.2. Overall ATCN Framework

ATCN comprises a shared feature extractor G_θ, a fault classifier C_ϕ, and the PMLA, PBKM, and OCR modules, as shown in Figure 1. Source samples x_s and target samples x_t are first normalized. The shared network then produces features z_s and z_t and classification probabilities p_s and p_t. PBKM selects pseudo-labels from target classification probabilities, providing the samples and class information used for target supervision. PMLA aligns the global distributions of both feature sets and computes same-class centroids using the PBKM selections, linking pseudo-label mining with class alignment. Meanwhile, OCR computes output entropy and probability differences under perturbations for all target samples. Source classification supervision and these adaptation losses jointly update the feature extractor and classifier. During inference, the model sequentially performs normalization, feature extraction, and classification.

The same architecture and training objective are used for cross-speed and cross-machine tasks, with a separate model trained for each source–target pair.

2.3. Shared Feature Extractor

Before network input, each signal segment undergoes Z-score normalization to reduce amplitude differences and maintain numerical stability. For a signal segment x of length L, the normalized signal is $\overline{x} = (x - \mu_x)/(\sigma_x + \epsilon)$. Here, μ_x and σ_x are the mean and standard deviation of that individual sample, and ϵ is a numerical stability constant set to 10^{-8}.

ATCN maps samples from both domains into the same feature space using a shared feature extractor. For normalized source and target samples x^s and x^t, the corresponding features are $z^s = G_\theta(x^s)$ and $z^t = G_\theta(x^t)$. This shared mapping provides a common representation for subsequent distribution alignment, pseudo-label mining, and output constraints.

Local waveforms and their temporal combinations in one-dimensional signals contain discriminative fault information. One-dimensional convolutional blocks progressively extract these patterns, followed by global average pooling and FC1 to obtain a 512-dimensional feature representation. Both domains share the same parameters, ensuring identical computational rules for feature generation and subsequent distance measurements within a common representation space. Source supervision and target adaptation jointly update these parameters, balancing class discrimination and interdomain distribution adjustment. Table 3 details the architecture.

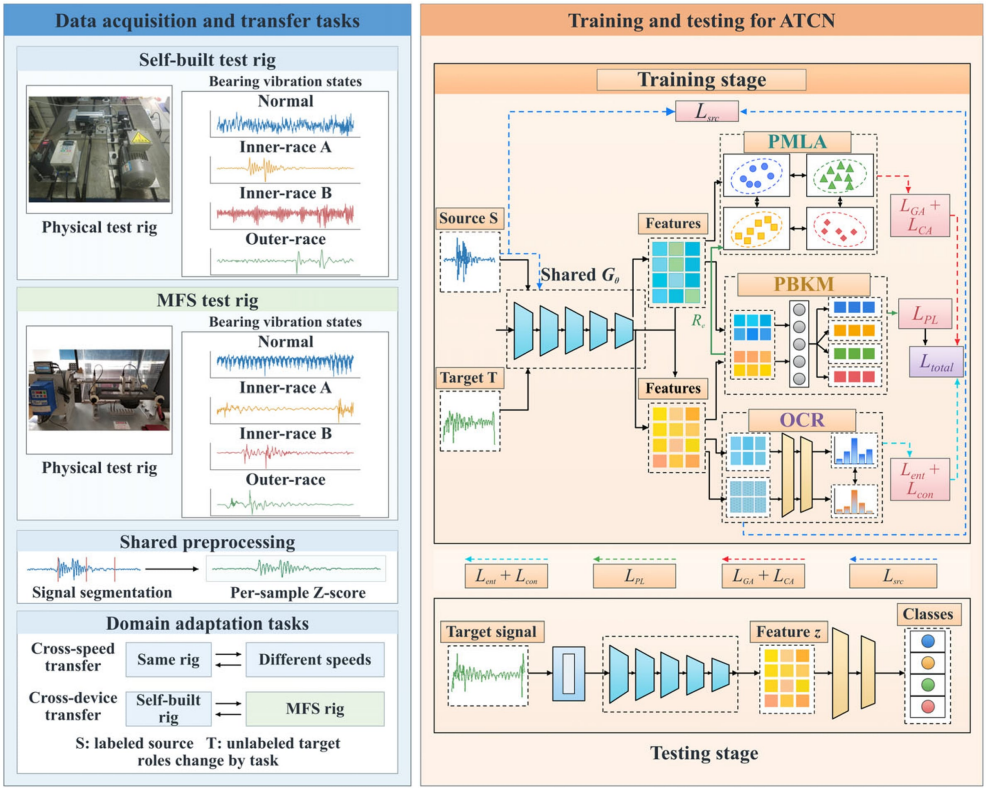


Figure 1. ATCN training and inference framework. Both domains share the feature extractor and classifier. Target pseudo-labels selected by PBKM support classification supervision and PMLA class-centroid computation. OCR constrains all target outputs. The modules are jointly optimized during training. Inference sequentially performs normalization, feature extraction, and classification.

Table 3. Parameters of the feature extractor and classifier used in the experiments.

Layer	Operation	Kernel/Stride	Output Shape
Conv1	Conv1d + BN + ReLU	64/16	(32, 64)
MaxPool1	MaxPool1d	2/2	(32, 32)
Conv2	Conv1d + BN + ReLU	3/1	(64, 32)
MaxPool2	MaxPool1d	2/2	(64, 16)
Conv3	Conv1d + BN + ReLU	3/1	(128, 16)
MaxPool3	MaxPool1d	2/2	(128, 8)
Conv4	Conv1d + BN + ReLU	3/1	(256, 8)
MaxPool4	MaxPool1d	2/2	(256, 4)
Conv5	Conv1d + BN + ReLU	3/1	(512, 4)
MaxPool5	MaxPool1d	2/2	(512, 2)
GAP	AdaptiveAvgPool1d	—	(512, 1)
FC1	Linear + ReLU + Dropout (0.3)	—	(512,)
FC2	Linear	—	(C,)

FC2 produces class logits, which are converted into probabilities by Softmax. Label-smoothed cross-entropy uses the logits, whereas PBKM pseudo-label selection and OCR use the probabilities. Both source and target inputs are perturbed during training, as described in Sections 2.6 and 2.7.

2.4. Progressive Multi-Level Adaptive Alignment

Speed changes alter temporal and statistical signal features, while device differences further affect feature distributions. Matching global distributions alone can leave same-class samples misaligned across domains. Subdomain adaptation addresses this issue by matching same-class distributions [36]. PMLA therefore combines global distribution alignment with class-centroid alignment. The former reduces overall interdomain differences, while the latter uses source labels and target pseudo-labels to adjust same-class feature correspondence.

Let the deep source and target features in the current mini-batch be $Z_s = \{z_i^s\}_{i=1}^m$ and $Z_t = \left\{z_j^t\right\}_{j=1}^n$, respectively, with feature dimension d. PMLA first combines kernel mean discrepancy and second-order covariance discrepancy to construct the global distribution alignment loss:

$$\mathcal{L}_{GA} = \kappa_{MMD}\mathcal{D}_{MMD}(Z_s, Z_t) + \frac{\kappa_{COV}}{4d^2}\|C_s - C_t\|_F \quad (2)$$

where $\mathcal{D}_{MMD}(Z_s, Z_t)$ is the biased estimate of squared maximum mean discrepancy computed using a single radial basis function kernel $k(u, v) = \exp\left(\frac{-\|u-v\|^2}{2{\sigma_k}^2}\right)$. The kernel scale is $\sigma_k = 10$. Here, C_s and C_t are the source and target sample covariance matrices, respectively, computed with numerical stabilization. The notation $\|\cdot\|_F$ denotes the Frobenius norm. Fixed scale coefficients κ_{MMD} and κ_{COV} balance the numerical magnitudes of the two terms; their values are listed in Table 4.

Table 4. Shared training and module parameters of ATCN.

Parameter	Value
Optimizer; initial learning rate; weight decay	AdamW; 0.0003; 0.01
Batch size; maximum epochs	64; 100
Learning rate scheduler	ReduceLROnPlateau, factor = 0.5, patience = 10
Gradient clipping maximum norm; Dropout	1.0; 0.3
Label smoothing coefficient; source noise supervision weight	0.1; 0.5
Source and target Gaussian noise standard deviations	0.01; 0.05 (normalized input scale)
Global alignment warm-up; pseudo-label warm-up	20 epochs; 10 epochs
Dynamic threshold upper and lower limits; decay coefficient	0.95; 0.30; 2
Target entropy weight λ_{ent}	0.1
Pseudo-label classification weight; class-centroid alignment weight	0.5; 1
Initial hybrid distance weight α_0; RBF kernel scale σ_k	0.5; 10
Global alignment scale coefficients	$\kappa_{MMD} = 100$; $\kappa_{COV} = 10$

Maximum mean discrepancy (MMD) measures differences between distribution mean embeddings in a reproducing kernel Hilbert space [37]. Covariance alignment describes differences in second-order statistical structure [25]. MMD compares kernel mean embeddings, while the covariance term explicitly constrains second-order feature statistics. Joint constraints adjust both types of global discrepancy. The subsequent centroid alignment term addresses class correspondence.

The global alignment term combines MMD with covariance alignment. The covariance component follows the second-order alignment principle in [25], while Equation (2) uses the Frobenius norm with the scaling coefficient specified in Table 4. For class-level adaptation, the subdomain alignment concept in [36] motivates the use of class information, whereas PMLA matches class centroids using the learnable hybrid distance in Equation (5).

Discriminative feature structures remain unstable early in training, so excessive alignment may prematurely bring different fault classes together. PMLA therefore gradually increases the global alignment weight. Let the training epoch be $e = 1,2,\ldots,E$ and the global alignment warm-up duration be E_g. Then:

$$\lambda_g(e) = \lambda_{mda}\min\left[1, (e-1)/E_g\right] \tag{3}$$

The global alignment weight increases linearly from zero to λ_{mda} and then remains constant, with λ_{mda} controlling its strength. Linear growth increases alignment uniformly across warm-up epochs, reducing abrupt optimization changes caused by strong constraints. The model gradually strengthens cross-domain distribution constraints while establishing source discrimination.

PMLA aligns same-class feature centroids using pseudo-labels selected by PBKM. Here, a class centroid is the mean of the selected feature vectors. For class c, the source and target centroids are defined as:

$$\mu_s^c = \frac{1}{|\mathcal{B}_s^c|}\sum_{i\in\mathcal{B}_s^c} z_i^s, \quad \mu_t^c = \frac{1}{|\mathcal{B}_t^c|}\sum_{j\in\mathcal{B}_t^c} z_j^t \tag{4}$$

where $\mathcal{B}_s^c = \{i : y_i^s = c\}$ is the source class-index set in the current mini-batch, and $\mathcal{B}_t^c = \left\{j \in R(e) : \hat{y}_j^t = c\right\}$ is the target class-index set restricted to the PBKM-selected indices $R(e)$ defined in Section 2.5. If either set is empty, that class is excluded from centroid alignment in the current iteration.

To capture both positional and directional differences between class centroids, we define a hybrid distance:

$$d_h(\mu_s^c, \mu_t^c) = \alpha\|\mu_s^c - \mu_t^c\|_2 + (1-\alpha)\left(1 - \frac{(\mu_s^c)^\top \mu_t^c}{\|\mu_s^c\|_2\|\mu_t^c\|_2}\right) \tag{5}$$

where the first term is the Euclidean distance and the second is the cosine distance. The mixing weight is $\alpha = \sigma(a) \in (0,1)$, where $\sigma(\cdot)$ denotes the Sigmoid function and a is a shared learnable scalar jointly updated with the network parameters. Cosine similarity is evaluated with an epsilon-stabilized denominator.

The Euclidean distance reflects differences in class-centroid position and scale, whereas the cosine distance reflects directional differences. Speed and device changes can affect these properties differently. The hybrid distance combines both measures and adjusts their contributions through a learnable weight.

The mixing weight is shared across classes and optimized jointly with the network. It adjusts the relative contribution of centroid magnitude and directional agreement during training. Section 3.5.3 compares this adaptive combination with its Euclidean-only and cosine-only endpoints.

The class-centroid alignment loss and the overall PMLA optimization term are:

$$\mathcal{L}_{CA} = \frac{1}{|C_r|}\sum_{c\in C_r} d_h(\mu_s^c, \mu_t^c), \quad \mathcal{L}_{PMLA} = \lambda_g(e)\mathcal{L}_{GA} + \mathcal{L}_{CA} \tag{6}$$

where C_r contains the valid classes having both source samples and high-confidence target pseudo-labeled samples in the current mini-batch. If this set is empty, $\mathcal{L}_{CA} = 0$. Figure 2 shows the computational relationship between global distribution alignment and class-centroid alignment.

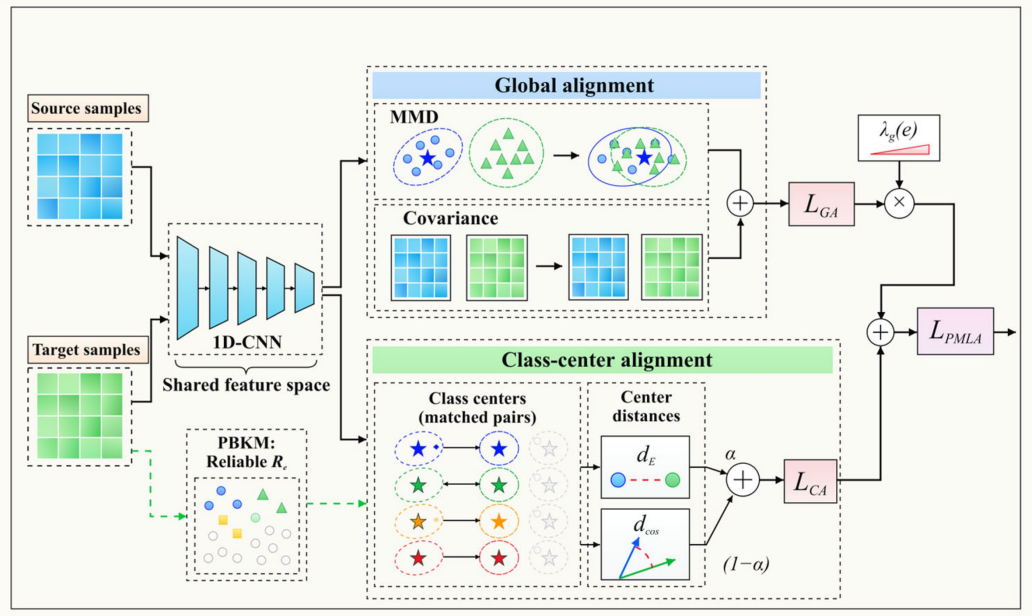


Figure 2. Global distribution and class-centroid alignment in PMLA. The class-centroid term uses high-confidence target pseudo-labels selected by PBKM.

2.5. Progressive Balanced Knowledge Mining

Pseudo-labeling provides supervision for unlabeled samples through model predictions, with confidence filtering controlling the quality of samples used in training [38]. Class-balanced self-training addresses uneven pseudo-label selection across classes [39]. Because target class information must be inferred from model predictions, early prediction errors and differences in class confidence can interfere with adaptation. PBKM controls pseudo-label introduction through warm-up, then uses dynamic thresholds and class-wise Top-K selection to regulate acceptance quality and class composition.

Confidence-based pseudo-labeling and class-balanced self-training have been studied in [38,39]. Building on these ideas, PBKM combines a warm-up stage, a decreasing confidence threshold, and class-wise Top-K selection to construct the target sample set used for classification and centroid alignment.

For a normalized target sample $\bar{x}_j^t$, its classification probability is $\mathbf{p}_j^t = \mathrm{softmax}\left(C_\phi\left(G_\theta\left(\bar{x}_j^t\right)\right)\right)$. The corresponding pseudo-label and confidence are defined as $\hat{y}_j^t = \underset{c}{\mathrm{argmax}}\, p_{jc}^t$ and $q_j = \underset{c}{\max}\, p_{jc}^t$, respectively.

During the first E_w epochs, PBKM disables pseudo-label classification and the class-centroid alignment that depends on pseudo-labels. The network is trained using source supervision, progressive global alignment, and OCR. After warm-up, when $e > E_w$, selected target pseudo-labels enter training to reduce the accumulation of early prediction errors.

To control progressive target sample acceptance, the dynamic confidence threshold at epoch e is defined as:

$$\tau(e) = \tau_{min} + (\tau_{max} - \tau_{min})\exp[-\gamma(e-1)/E] \tag{7}$$

where τ_{max} and τ_{min} are the upper limit and asymptotic lower limit, respectively. Here, γ is the decay coefficient and E the preset maximum number of training epochs. The threshold gradually decreases from a high level, initially accepting high-confidence samples before expanding target sample coverage. Maintaining a high threshold can limit later sample use, whereas adopting a low threshold prematurely increases the risk of erroneous supervision. The dynamic threshold therefore balances pseudo-label quality and sample coverage.

Global thresholding may allocate most pseudo-label slots to easily classified classes. PBKM therefore performs class-wise Top-K selection within each target mini-batch. For class c, the candidate and final selection sets are defined as:

$$\mathcal{S}_c(e) = \left\{ j \mid \hat{y}_j^t = c,\ q_j > \tau(e) \right\},\quad \mathcal{R}(e) = \bigcup_{c=1}^{C} \mathrm{TopK}(\mathcal{S}_c(e), K) \tag{8}$$

Candidates in each class are ranked by confidence, and at most K samples are retained. The class set is empty if no sample meets the threshold. The confidence threshold filters predictions, while the per-class cap controls the contribution of classes with many high-confidence samples to target supervision and centroid computation.

The target pseudo-label classification loss based on the high-confidence pseudo-label set is defined as:

$$\mathcal{L}_{PL} = \frac{1}{|\mathcal{R}(e)|} \sum_{j \in \mathcal{R}(e)} \ell_{LS}\left(C_\phi\left(G_\theta\left(\bar{x}_j^t \right) \right), \hat{y}_j^t \right) \tag{9}$$

where ℓ_{LS} denotes label-smoothed cross-entropy. If $\mathcal{R}(e) = \emptyset$, we set $\mathcal{L}_{PL} = 0$. Selected pseudo-labels provide both target classification supervision and target class-centroid information for PMLA, linking target knowledge mining with class-level feature alignment. Figure 3 illustrates the PBKM warm-up, dynamic thresholding, and class-wise Top-K selection procedure.

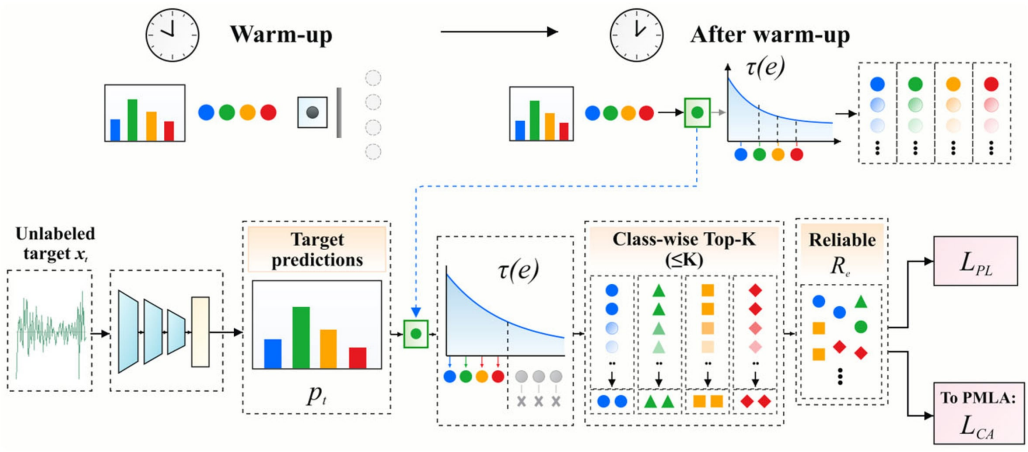


Figure 3. Progressive pseudo-label selection in PBKM. After warm-up, the dynamic threshold and class-wise Top-K selection jointly determine the high-confidence sample set.

2.6. Output-Level Consistency Regularization

Feature alignment and pseudo-label supervision mainly adjust representations and class information, but target predictions may still fluctuate under small input changes. OCR combines entropy minimization for unlabeled data [40] with perturbation consistency constraints [41]. It reduces predictive entropy and encourages similar classification probabilities for unperturbed and perturbed inputs.

Entropy minimization and prediction consistency provide the foundations for the target output constraints [40,41]. In OCR, these principles are combined through the weighted objective in Equation (12), with consistency evaluated between predictions for the original target input and its Gaussian-perturbed counterpart.

First, the target entropy minimization loss is defined as:

$$\mathcal{L}_{ent} = -\frac{1}{n}\sum_{j=1}^{n}\sum_{c=1}^{C} p_{jc}^{t}\log\left(p_{jc}^{t}+\epsilon\right) \tag{10}$$

Minimizing this term reduces target predictive entropy and encourages more concentrated class probabilities. To further constrain output behavior under perturbations, Gaussian noise is added to normalized target inputs, giving $\tilde{x}_j^t = \bar{x}_j^t + \eta_j^t$, where $\eta_j^t \sim \mathcal{N}(0, \sigma_t^2 I)$. Let the classification probability of the perturbed sample be $\tilde{\mathbf{p}}_j^t = \mathrm{softmax}\left(C_\phi\left(G_\theta\left(\tilde{x}_j^t\right)\right)\right)$. The noise perturbation consistency loss is defined as:

$$\mathcal{L}_{con} = \frac{1}{nC}\sum_{j=1}^{n}\sum_{c=1}^{C}\left(p_{jc}^{t}-\tilde{p}_{jc}^{t}\right)^2 \tag{11}$$

The consistency loss averages over all target samples and class probabilities in the current mini-batch, limiting output differences caused by local input changes. OCR computes losses from classification probabilities. Consequently, samples rejected by PBKM and samples encountered during warm-up can also contribute to target optimization. The entropy and consistency terms jointly constrain output decisiveness and local stability.

The weighted OCR optimization term is:

$$\mathcal{L}_{OCR} = \lambda_{ent}\mathcal{L}_{ent} + \lambda_{con}\mathcal{L}_{con} \tag{12}$$

where λ_{ent} and λ_{con} are the weights for entropy minimization and perturbation consistency, respectively. Source noise augmentation constructs supervision from true class labels, whereas OCR constrains unlabeled target classification outputs. Both are optimized through the total objective. Figure 4 shows the computation of the target output constraints.

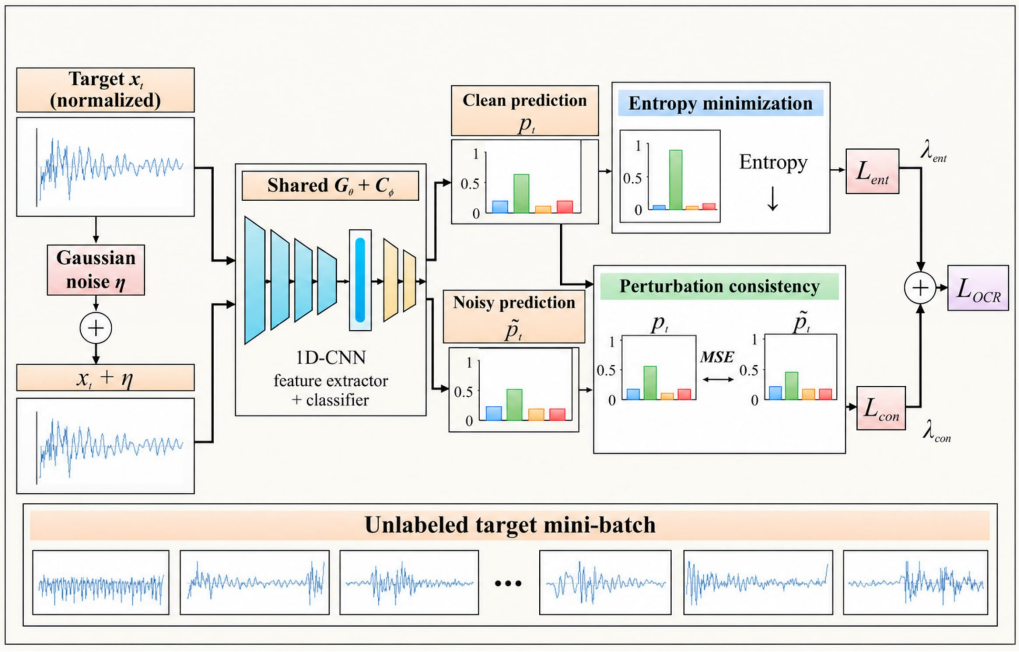


Figure 4. Target output constraints in OCR, comprising entropy minimization for unperturbed inputs and probability consistency between unperturbed and perturbed inputs.

2.7. Overall Objective and Optimization

The overall ATCN objective combines noise-augmented source supervision, PMLA, PBKM, and OCR. For a source mini-batch $\mathcal{B}_s = \{(\overline{x}_i^s, y_i^s)\}_{i=1}^m$, noisy samples $\tilde{x}_i^s = \overline{x}_i^s + \eta_i^s$ are first generated, where $\eta_i^s \sim \mathcal{N}(0, \sigma_s^2 I)$. Unperturbed and noisy source samples use the same true labels for joint supervision:

$$\mathcal{L}_{src} = \frac{1}{m}\sum_{i=1}^{m}\left[(1-\beta)\ell_{LS}\left(f\left(\overline{x}_i^s\right), y_i^s\right) + \beta\ell_{LS}\left(f\left(\tilde{x}_i^s\right), y_i^s\right)\right] \tag{13}$$

where $\beta \in [0, 1]$ controls the contribution of noisy source supervision and ℓ_{LS} denotes label-smoothed cross-entropy. Source augmentation trains unperturbed and perturbed inputs against the same ground-truth labels, while target OCR penalizes probability differences between the two target views. These losses are optimized jointly through the total objective.

Combining all losses, the total ATCN optimization objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{src} + \lambda_g(e)\mathcal{L}_{GA} + \mathcal{L}_{CA} + \lambda_{PL}\mathcal{L}_{PL} + \lambda_{ent}\mathcal{L}_{ent} + \lambda_{con}\mathcal{L}_{con} \tag{14}$$

The objective combines source class discrimination, interdomain distribution alignment, and target output constraints. Source supervision provides class discrimination, while PMLA adjusts global distributions and same-class feature centroids. PBKM supplies selected samples and pseudo-labels for target supervision and centroid computation, while OCR constrains outputs for all target samples. These losses jointly update the shared network. Pseudo-labels are reselected from model predictions, coupling the use of class information with feature adaptation. During pseudo-label warm-up, pseudo-label classification and class-centroid alignment losses are set to 0. The remaining losses contribute according to their respective weights. In the implementation, the class-centroid alignment loss has an explicit weight of 1.0. For clarity, its unit coefficient is omitted from Equation (14).

Label smoothing softens supervision labels to mitigate classifier overconfidence [42]. The label smoothing coefficient, noise standard deviations, and shared loss weights are listed in Table 4. Task-specific hyperparameters are selected within the ranges in Table 5.

Table 5. Values and fixed conditions for joint sensitivity analysis.

Parameter	Seven Values	Corresponding Fixed Planes
λ_{mda}	0.0625, 0.125, 0.25, 0.5, 1, 2, 4	$\lambda_{con} = 0.1$ or K = 16
K	2, 4, 8, 12, 16, 24, 32	$\lambda_{con} = 0.1$ or $\lambda_{mda} = 1$
λ_{con}	0.025, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4	K = 16 or $\lambda_{mda} = 1$

During training, source and target mini-batches enter the shared network simultaneously. Backpropagation jointly updates the feature extractor, classifier, and hybrid distance parameter. Gradient clipping limits the magnitude of individual updates. The learning rate is adjusted using the source supervision loss, and checkpoints are selected using the source validation set. Target labels are used to compute metrics after training.

During inference, each target sample is normalized and then passed through the shared feature extractor $G_\theta(\cdot)$ and classifier $C_\phi(\cdot)$ to obtain its diagnostic result. Algorithm 1 summarizes the training procedure.

Algorithm 1 ATCN Training Procedure

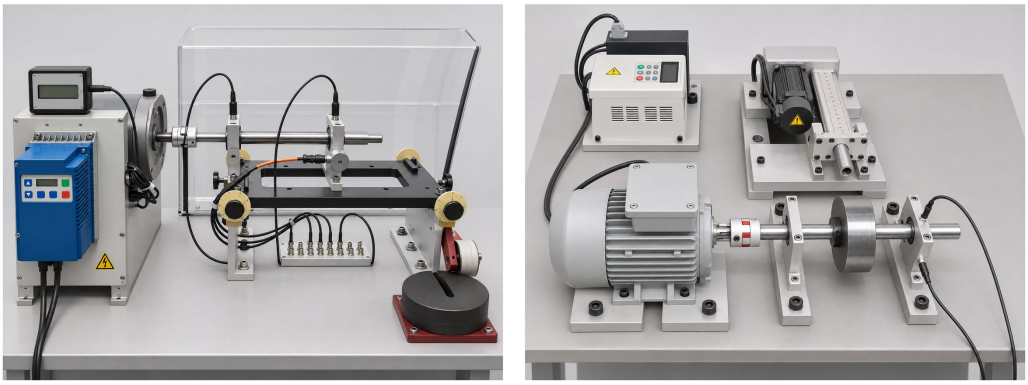
Input: labeled source domain $\mathcal{D}_s$; unlabeled target domain $\mathcal{D}_t$; maximum epochs E; pseudo-label and global alignment warm-up epochs E_w and E_g; thresholds τ_{max} and τ_{min}; decay coefficient γ; per-class selection limit K; noise standard deviations σ_s and σ_t; source noise weight β; and loss weights λ_{PL}, λ_{ent}, and λ_{con}. The task-specific global alignment strength λ_{mda} and the task-specific values of K and λ_{con} are selected within the ranges in Table 5; scale coefficients κ_{MMD} and κ_{COV} follow Table 4.
Output: trained feature extractor $G_\theta(\cdot)$ and classifier $C_\phi(\cdot)$.

Initialize $G_\theta(\cdot)$, $C_\phi(\cdot)$, and the learnable hybrid distance parameter a.
Apply sample-wise Z-score normalization to $\mathcal{D}_s$ and $\mathcal{D}_t$.
for epoch $e = 1$ to E do
  Compute $\tau(e) = \tau_{min} + (\tau_{max} - \tau_{min})\exp[-\gamma(e-1)/E]$.
  Compute $\lambda_g(e) = \lambda_{mda}\min[1,(e-1)/E_g]$.
  for each source and target mini-batch pair $(\mathcal{B}_s, \mathcal{B}_t)$ do
    Extract unperturbed features and compute source and target predictions.
    Generate perturbed inputs $\widetilde{X}_s = \overline{X}_s + \eta_s$ and $\widetilde{X}_t = \overline{X}_t + \eta_t$.
    Compute perturbed source and target predictions through the shared network.
    Compute the joint source supervision loss $\mathcal{L}_{src}$.
    Compute $\mathcal{L}_{GA}$, $\mathcal{L}_{ent}$, and $\mathcal{L}_{con}$.
    Set $\mathcal{L}_{PL} = 0$ and $\mathcal{L}_{CA} = 0$.
    if $e > E_w$ then
      Generate target pseudo-labels and confidence scores.
      For each class, retain candidates with confidence above $\tau(e)$ and select at most K highest-confidence samples.
      Combine selected samples into the high-confidence pseudo-label set $\mathcal{R}(e)$.
      if $\mathcal{R}(e) \neq \emptyset$ then
        Compute $\mathcal{L}_{PL}$ and $\mathcal{L}_{CA}$.
      end if
    end if
    Compute $\mathcal{L}_{total} = \mathcal{L}_{src} + \lambda_g(e)\mathcal{L}_{GA} + \mathcal{L}_{CA} + \lambda_{PL}\mathcal{L}_{PL} + \lambda_{ent}\mathcal{L}_{ent} + \lambda_{con}\mathcal{L}_{con}$.
    Update θ, ϕ, and a through backpropagation with gradient clipping.
  end for
  Update the learning rate using the source supervision loss.
end for
Restore the checkpoint with the best source-validation Macro-F1 and return $G_\theta(\cdot)$ and $C_\phi(\cdot)$.

3. Results and Discussion

3.1. Dataset and Test Rigs

The experiments used the public HIT-SM bearing dataset released by the HIT Sensing & Measurement Lab [43]. Data were collected using a SpectraQuest MFS test rig and the dataset providers' self-built test rig, denoted here as MFS and Self, respectively. Figure 5 shows both platforms. The MFS test rig uses a 1 hp rated motor and a flexible coupling. The faulty bearing is mounted at the non-drive end, with a 5 kg cantilever load applied to the shaft end. The bearing is a 6205 deep-groove ball bearing, and the accelerometer model is SQI608A11-3F/8. The Self test rig has a different mechanical structure and uses an electric cylinder to adjust radial loading. These differences in structure and loading provide a basis for evaluating cross-machine transfer.



(a) (b)

Figure 5. Test rigs in the public dataset: (**a**) MFS; (**b**) Self. Test-rig and acquisition details are provided in [43].

The data cover speeds of 600, 900, and 1200 r/min, with vibration sampled at 51.2 kHz. The HIT-SM dataset currently provides data at only these three rotational speeds. The original dataset contains seven states, including normal operation and inner- and outer-raceway faults. We selected four health states, Normal, IR5, IR8, and OR5, to ensure identical label spaces across domains. IR and OR denote inner- and outer-raceway faults, respectively. The numbers indicate the angular extents, in degrees, of the raceway fault regions. Thus, IR5, IR8, and OR5 represent 5° inner-raceway, 8° inner-raceway, and 5° outer-raceway faults, respectively [43]. Table 2 lists the data settings and sample counts. The two test rigs provide controlled changes in rotational speed and machine configuration for evaluating transfer diagnosis. The laboratory data cover selected raceway fault states at three nominal speeds. Field validation can extend this evaluation to operating-turbine vibration paths, changing aerodynamic loads, environmental conditions, and naturally evolving damage.

In Table 3, Conv1 uses 24-point padding on each side, while Conv2–Conv5 preserve sequence length. The globally averaged features are flattened, and FC1 produces a 512-dimensional shared representation. FC2 outputs classification scores for four classes.

3.2. Transfer Tasks and Data Processing

Table 6 defines twelve single-source-to-single-target transfer tasks. In each transfer task, one domain factor varies while the other is held constant: rotational speed varies in cross-speed tasks T1–T6 on the Self platform, covering all six directed combinations of the three speeds, whereas the test platform varies in cross-machine tasks D1–D6 at the same speed, in both the Self → MFS and MFS → Self directions. The same ATCN architecture and training objective are applied to both task families, with a separate model trained for each source–target pair and no combination of multiple labeled source domains.

Vibration recordings for each class were sequentially divided into nonoverlapping segments of length 1024. For source-domain training and validation, 1280 samples per class were retained, giving 5120 samples in total. The source data were stratified by class into 4096 training samples and 1024 validation samples, approximately an 80%:20% split. The sets contain no identical or overlapping samples. Each sample is independently normalized using its own mean and standard deviation, with 10^{-8} added to the denominator for numerical stability. All methods use identical splits for a given task and random seed.

For each target-domain recording, the first 80% of the signal is used for unlabeled domain adaptation, and the remaining 20% is held out for final evaluation. The two time segments are divided separately into nonoverlapping samples of 1024 points. Across four fault classes, the target adaptation set contains 4096 samples and the held-out test set contains 1024 samples. The held-out samples do not participate in training, validation, or

hyperparameter selection; their labels are used only to compute evaluation metrics. Source and target domains come from different acquisition conditions or machines, with no same-recording sample overlap between domains. The target adaptation and test samples come from different time segments of the same target-domain recording.

Table 6. Twelve transfer tasks.

Task	Type	Source Domain	Target Domain
T1	Cross-speed	Self600	Self900
T2	Cross-speed	Self900	Self600
T3	Cross-speed	Self600	Self1200
T4	Cross-speed	Self1200	Self600
T5	Cross-speed	Self900	Self1200
T6	Cross-speed	Self1200	Self900
D1	Cross-machine	Self600	MFS600
D2	Cross-machine	MFS600	Self600
D3	Cross-machine	Self900	MFS900
D4	Cross-machine	MFS900	Self900
D5	Cross-machine	Self1200	MFS1200
D6	Cross-machine	MFS1200	Self1200

Transfer directions are defined as Source Domain → Target Domain.

3.3. Comparison Methods and Evaluation Settings

The comparison methods are CNN-C [24], D-CORAL [25], DDTLN [26], DXDA [27], PKMTCL [28], and DANN [29]. Each model was configured for the four-class label space and data protocol while retaining its own network and optimization structure. CNN-C [24] uses ResNet-1D and is trained with source cross-entropy and Center Loss [44]. D-CORAL matches feature covariances across domains in addition to source classification [25]. DDTLN uses source I-Softmax and combines mean embedding and covariance discrepancies for global and class-conditional alignment [26]. Its target argmax pseudo-labels also support target classification loss. PKMTCL uses 12-dimensional statistical and time–frequency priors, a prior contrastive loss, and low-entropy Top-K prototype alignment [28]. DXDA first pretrains on the source domain. It then combines domain-adversarial learning with an MSE preservation loss relative to frozen pretrained features [27]. DANN learns domain-invariant features through gradient reversal [29].

For each comparison method and transfer task, we searched the learning rate, a method-specific loss weight, and weight decay. Bayesian optimization and local grid search were combined, and configurations were selected using source-domain validation performance. The search ranges are summarized in Table 7.

Learning rates and method-specific loss weights were searched on logarithmic scales. The DXDA settings shown apply to the adaptation stage. Nineteen distinct configurations were evaluated for each method–task pair.

ATCN was implemented in PyTorch and trained using CUDA. Main comparisons and ablations used random seeds 42, 123, 2024, 3407, and 7777; sensitivity analysis used seed 42. Bayesian optimization [45] guided the search ranges in Table 5, within which task-specific values of λ_{mda}, K, and λ_{con} were selected by grid search using source-validation ACC. Table 4 lists the parameters shared across tasks. Training lasted at most 100 epochs.

Table 7. Hyperparameter search settings for the six comparison methods.

Method	Learning Rate	Method-Specific Parameter	Parameter Range	Weight Decay
CNN-C	0.0003–0.003	Center loss weight	0.01–1	0, 0.0001, 0.001
D-CORAL	0.001–0.01	Covariance alignment weight	0.1–10	0, 0.0001, 0.001
DDTLN	0.0003–0.003	Class-conditional alignment weight	0.01–1	0, 0.0001, 0.001
DXDA	0.0003–0.003	Feature preservation weight	0.1–10	0, 0.0001, 0.001
PKMTCL	0.0002–0.001	Contrastive loss weight	0.3–1.5	0, 0.0001, 0.001
DANN	0.0003–0.003	Domain classification weight	0.1–10	0, 0.0001, 0.001

The evaluation metrics are ACC, Macro-F1, and Macro-AUROC. ACC is the proportion of correctly classified samples. Macro-F1 averages the F1 scores of the four classes with equal weights, assessing their combined precision and recall. Macro-AUROC uses a one-versus-rest approach. It measures how well each class score ranks positive samples above samples from the other classes, then averages the results equally across the four classes. The metrics describe classification accuracy, balance in class recognition, and probability ranking, respectively.

Aggregated ACC, Macro-F1, and Macro-AUROC results are reported as the mean and sample standard deviation across five runs, with 5−1 degrees of freedom used to calculate the sample standard deviation. Six-task averages are unweighted averages of the task means. Reported means, standard deviations, and performance differences are displayed to two decimal places. Differences are expressed in percentage points. Confusion matrices and ROC curves shown in the main text represent individual runs; the aggregated metrics report five-seed means and sample standard deviations.

To assess run-to-run variation in performance differences, ATCN was compared with the baseline having the highest five-seed mean ACC on each transfer task. ACC differences (ATCN minus baseline) were paired by random seed and shared data split. We calculated the mean paired difference, its approximate 95% confidence interval, and a two-sided paired t-test p-value for each task. The Holm procedure adjusted the p-values across the 12 comparisons. Calculations used the available ACC values rounded to two decimal places; confidence intervals were not adjusted for multiple comparisons. The analysis describes variation across runs under the current evaluation protocol.

3.4. Cross-Speed Fault Diagnosis Results

Figure 6 compares ACC for the seven methods on T1–T6. ATCN achieved a mean ACC of 99.36%, exceeding the closest method, DXDA (90.71%), by 8.65 percentage points. Its mean Macro-F1 was 99.36%, exceeding DXDA (90.53%) by 8.83 percentage points. Mean ACC values for D-CORAL, PKMTCL, and DANN were 88.02%, 89.64%, and 87.52%, respectively. Figure 7 presents Macro-F1 and its distribution across five runs, and Figure 8 compares ROC curves.

ATCN achieved ACC values of 99.06–99.82% across T1–T6. The best-performing baselines on the respective tasks were D-CORAL, DANN, DXDA, DXDA, D-CORAL, and DXDA. ATCN exceeded them by 9.16, 10.45, 9.59, 4.69, 10.70, and 1.19 percentage points, respectively. Baseline rankings varied with transfer direction. T3 (Self600 → Self1200) and T4 (Self1200 → Self600) form an opposite-direction pair, and DXDA was the strongest baseline in both. ATCN exceeded DXDA by 9.59 and 4.69 percentage points, respectively, indicating that the performance gap differed between the two directions. Table 8 presents the individual ACC results across five runs for each cross-speed task, together with the means, sample standard deviations, and approximate 95% confidence intervals.

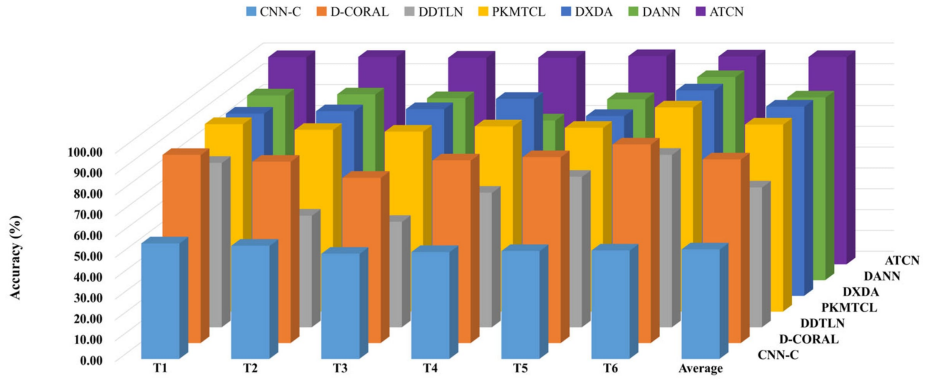


Figure 6. ACC on cross-speed tasks. Bar heights show five-seed means. Average denotes the reported six-task aggregate.

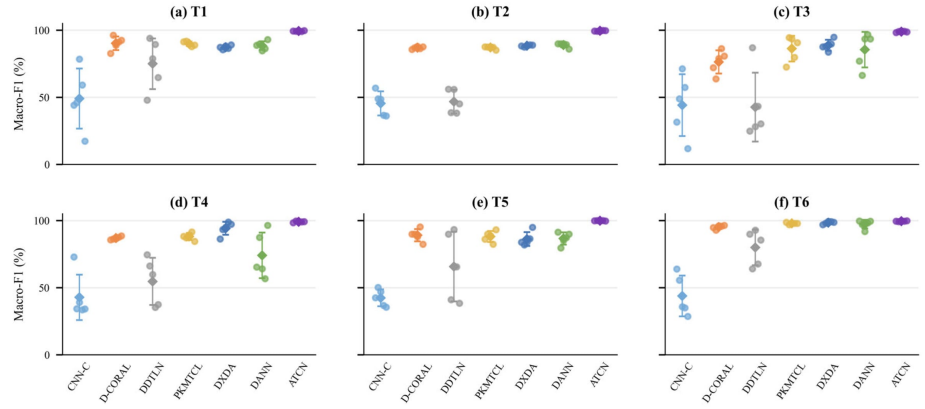


Figure 7. Five-seed distributions of cross-speed Macro-F1. Points represent individual runs; diamonds and error bars indicate means and one sample standard deviation. All panels share the same vertical-axis range.

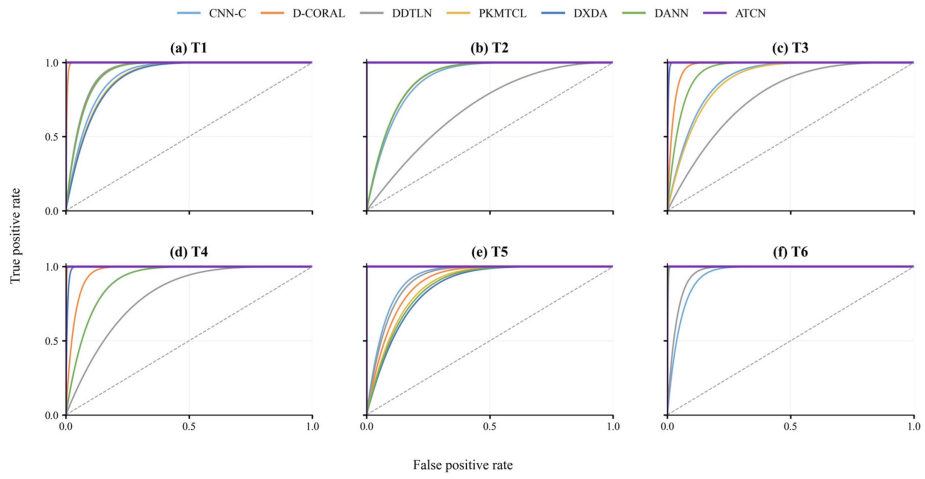


Figure 8. Single-run ROC curves for the six cross-speed tasks. Panels (**a**–**f**) correspond to T1–T6. Colors distinguish the seven methods. The dashed diagonal line in each panel represents chance-level performance (AUROC = 0.5).

ACC is based on 1024 test samples. Summary statistics were calculated before rounding and are displayed to two decimal places. The approximate 95% intervals use Student's t distribution with four degrees of freedom and describe uncertainty in the mean across runs

under a normality assumption. The symmetric T5 interval extends above 100% because it is unconstrained.

Table 8. Individual cross-speed ACC results for ATCN and approximate 95% confidence intervals (%, $n = 5$).

Task	42	123	2024	3407	7777	Mean ± SD	95% CI
T1	99.32	99.22	99.41	99.02	99.71	99.34 ± 0.25	99.02–99.65
T2	99.41	99.02	99.22	99.80	99.71	99.43 ± 0.33	99.03–99.84
T3	98.83	99.41	99.32	99.12	98.63	99.06 ± 0.33	98.65–99.47
T4	99.12	99.71	99.22	99.02	98.44	99.10 ± 0.45	98.54–99.67
T5	99.80	100.00	99.71	100.00	99.61	99.82 ± 0.17	99.61–100.04
T6	99.61	99.22	99.90	99.71	99.80	99.65 ± 0.26	99.32–99.98

Across the 30 runs, ACC ranged from 98.44% to 100.00%, and task-specific sample standard deviations ranged from 0.17 to 0.45 percentage points. These results show stable five-seed performance under the evaluation protocol described in Section 3.2. The cross-machine results in Section 3.5 further assess transfer between test rigs, and future work will evaluate independently acquired target recordings.

In Figure 7, ATCN achieved mean Macro-F1 and ACC values of 99.36%. Task-specific Macro-F1 standard deviations were 0.20–0.46 percentage points, indicating relatively stable recognition across classes and random seeds. Macro-F1 values for DXDA and PKMTCL were 90.53% and 89.51%, close to their respective ACC values of 90.71% and 89.64%. CNN-C and DDTLN had Macro-F1 values 7.78 and 6.23 percentage points below their ACC values, respectively, reflecting differences in class recognition. CNN-C trains with source classification supervision and center loss without adapting to the target distribution. Class structures learned in the source domain may therefore become unsuitable after speed changes. DDTLN achieved 42.66% ± 25.69% Macro-F1 on T3. Its predicted pseudo-labels directly contribute to class-conditional alignment and target classification loss, so early prediction changes may influence subsequent training through feedback. DANN had a standard deviation of 16.99 percentage points on T4, indicating substantial variation in domain-adversarial training for that direction. These results show differences in both class recognition and run-to-run variability across the evaluated methods.

Figure 8 presents ROC curves, illustrating the distinction between probability ranking and final class decisions. Across five runs, ATCN achieved a mean Macro-AUROC that rounds to 100.00%. The strongest baselines on T1, T3, and T4 were D-CORAL (99.61%), DXDA (99.70%), and PKMTCL (99.84%), respectively. ATCN exceeded them by 0.38, 0.30, and 0.15 percentage points. On T2 and T5, the strongest baselines were DXDA (89.99%) and CNN-C (92.71%). ATCN exceeded them by approximately 10.01 and 7.29 percentage points, respectively. On T6, Macro-AUROC values for both ATCN and PKMTCL approached 100%. On T1, CNN-C achieved 91.18% AUROC but 55.27% ACC; the corresponding D-CORAL values were 99.61% and 90.17%. Thus, strong probability ranking can coexist with class decision errors. DDTLN achieved Macro-AUROC values of 69.42% and 76.89% on T2 and T3. Its predicted pseudo-labels contribute to class-conditional alignment, so errors in class information may also affect ranking. ATCN showed high ranking performance across the six transfer directions.

The displayed value of 100.00% is rounded; it does not imply that every run achieved exactly 100% Macro-AUROC or that every test sample was classified correctly.

Figures 9 and 10 show class-specific errors in the illustrated runs. On T1, ATCN's diagonal counts for the four classes were 256, 254, 254, and 253. CNN-C correctly identified only 24/256 IR8 samples and 83/256 OR5 samples, while DDTLN correctly identified 70/256 OR5 samples. On T3, ATCN's diagonal counts were 252, 252, 252, and 256. CNN-C correctly

identified no Normal samples and 144/256 OR5 samples, and DDTLN correctly identified no Normal samples. These concentrated class errors are consistent with the lower Macro-F1 values of the corresponding methods, whereas ATCN retained high recognition across all four classes in the illustrated T1 and T3 runs.

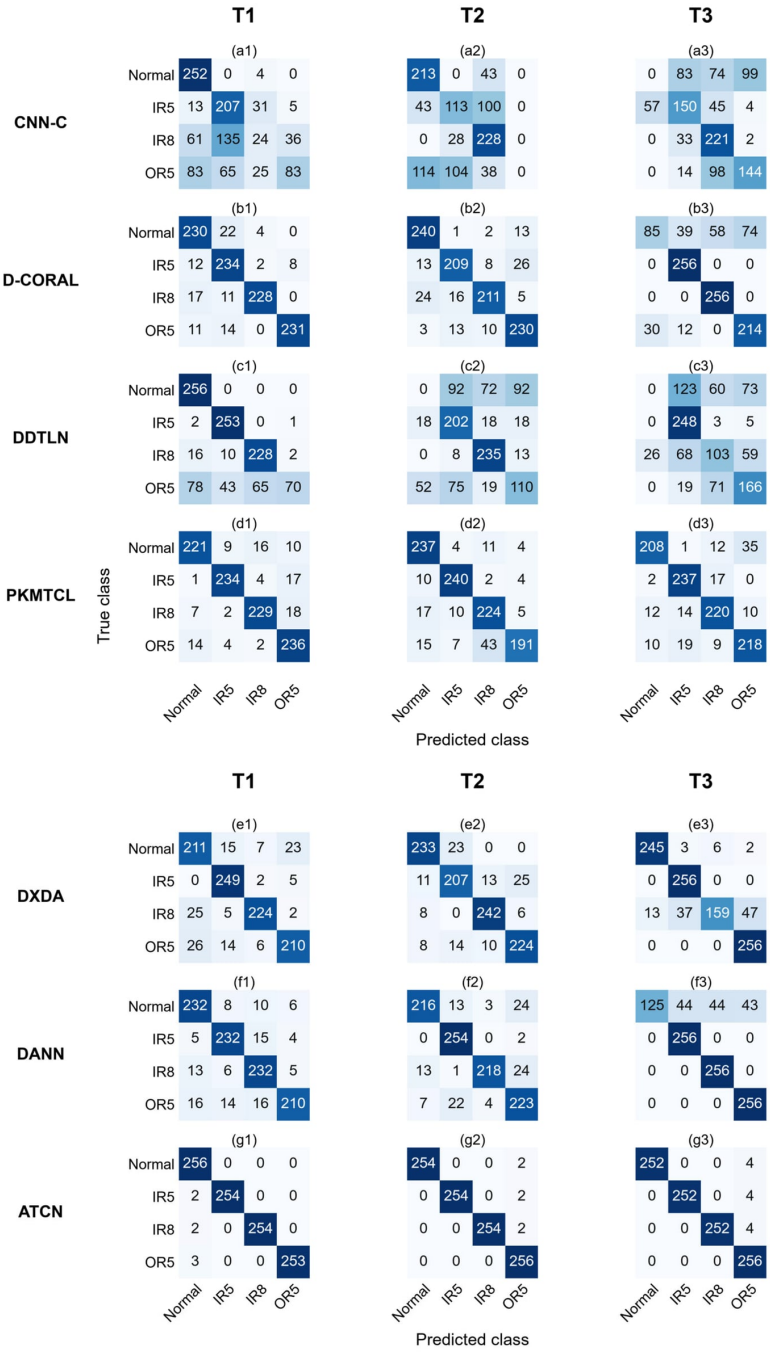


Figure 9. Single-run confusion matrices for cross-speed tasks T1–T3: (a1) CNN-C on T1; (a2) CNN-C on T2; (a3) CNN-C on T3; (b1) D-CORAL on T1; (b2) D-CORAL on T2; (b3) D-CORAL on T3; (c1) DDTLN on T1; (c2) DDTLN on T2; (c3) DDTLN on T3; (d1) PKMTCL on T1; (d2) PKMTCL on T2; (d3) PKMTCL on T3; (e1) DXDA on T1; (e2) DXDA on T2; (e3) DXDA on T3; (f1) DANN on T1; (f2) DANN on T2; (f3) DANN on T3; (g1) ATCN on T1; (g2) ATCN on T2; (g3) ATCN on T3. Matrix rows denote true classes and columns denote predicted classes, both ordered Normal, IR5, IR8, and OR5. Numbers in the cells indicate test-sample counts, with darker blue indicating larger counts.

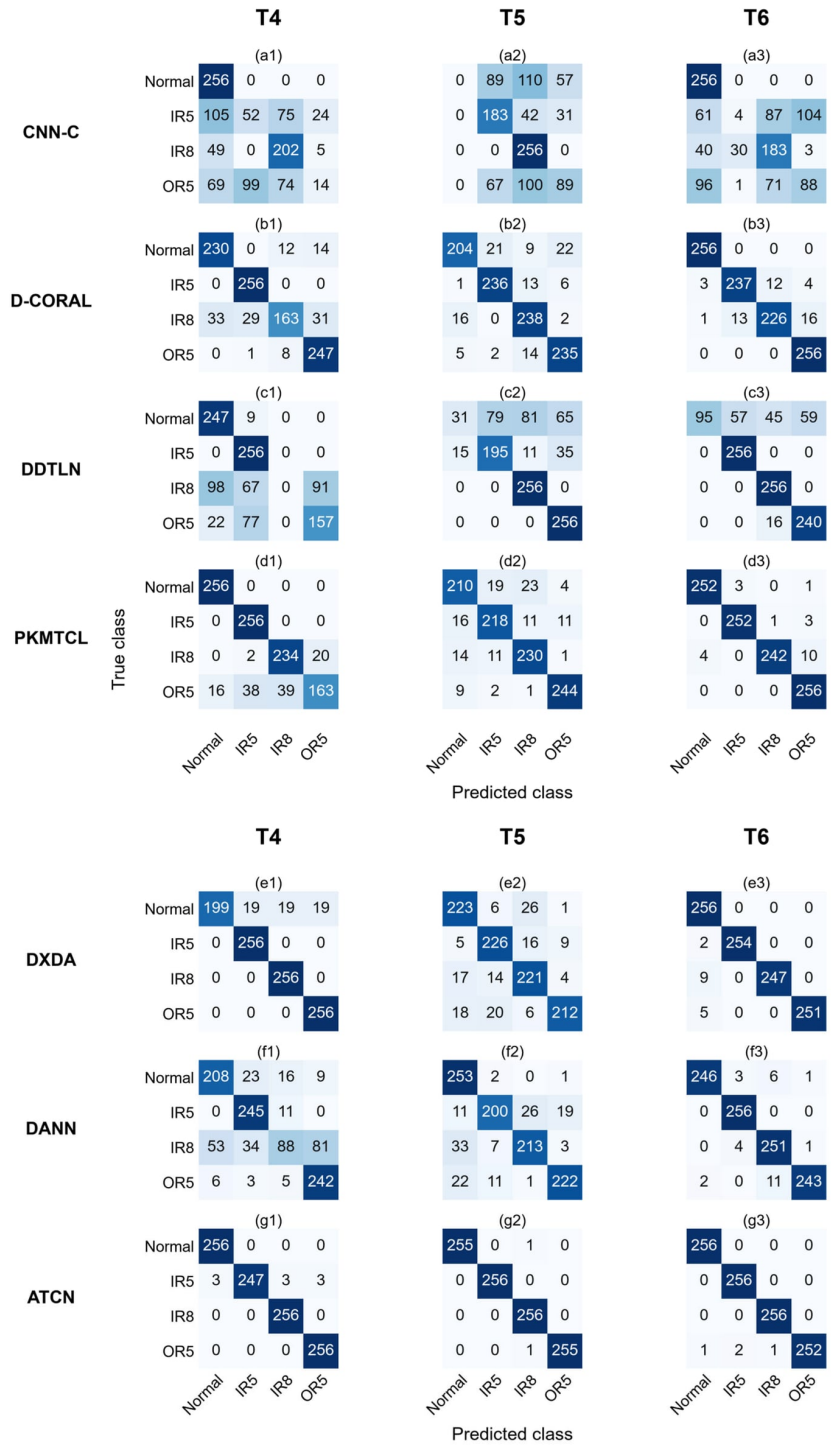


Figure 10. Single-run confusion matrices for cross-speed tasks T4–T6: (a1) CNN-C on T4; (a2) CNN-C on T5; (a3) CNN-C on T6; (b1) D-CORAL on T4; (b2) D-CORAL on T5; (b3) D-CORAL on T6; (c1) DDTLN on T4; (c2) DDTLN on T5; (c3) DDTLN on T6; (d1) PKMTCL on T4; (d2) PKMTCL on T5; (d3) PKMTCL on T6; (e1) DXDA on T4; (e2) DXDA on T5; (e3) DXDA on T6; (f1) DANN on T4; (f2) DANN on T5; (f3) DANN on T6; (g1) ATCN on T4; (g2) ATCN on T5; (g3) ATCN on T6. Matrix rows denote true classes and columns denote predicted classes, both ordered Normal, IR5, IR8, and OR5. Numbers in the cells indicate test-sample counts, with darker blue indicating larger counts.

Across Figures 6–10, ATCN achieved high ACC, Macro-F1, and probability ranking performance on cross-speed tasks. On T1, almost all samples from the four classes fell on the diagonal. On T3, a few Normal, IR5, and IR8 samples were still assigned to OR5. Overall

performance improvements accompanied fewer class errors, while the remaining errors varied by task and class.

To illustrate domain alignment in an additional cross-speed setting, Figure 11 visualizes the source- and target-domain feature distributions for the MFS 600-to-1200 r/min transfer task using t-SNE.

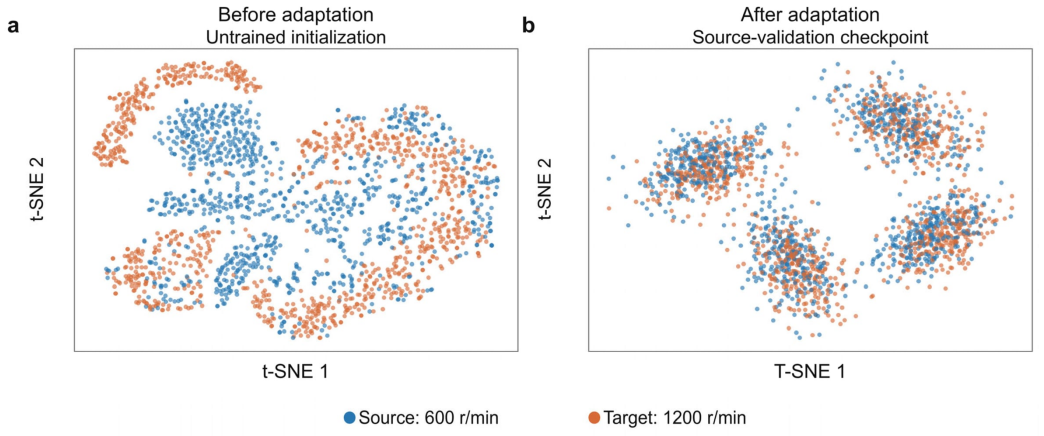


Figure 11. Two-dimensional t-SNE visualization of source- and target-domain features for MFS 600-to-1200 r/min transfer. (**a**) Before adaptation; (**b**) after adaptation. Blue points represent the source domain (600 r/min), and orange points represent the target domain (1200 r/min).

In the post-adaptation panel, source- and target-domain points overlap within several compact groups. Figure 11 provides a qualitative visualization of domain alignment in the displayed feature representation.

In a separate single-run comparison on the MFS 600-to-1200 r/min transfer task, ATCN and the six comparison methods were evaluated using seed 42. Figure 12 presents their diagnostic accuracies.

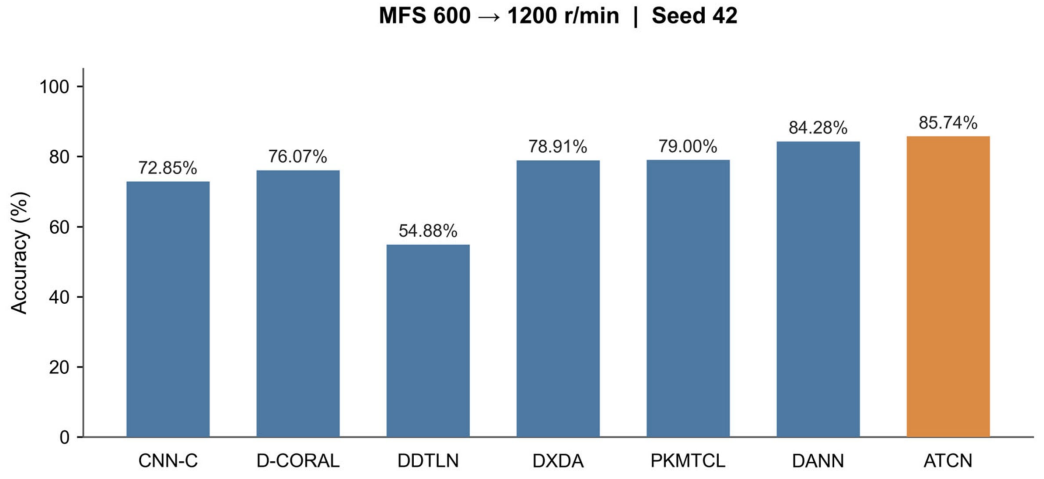


Figure 12. Diagnostic accuracy comparison on the MFS 600-to-1200 r/min transfer task. The bars show single-run results using seed 42.

ATCN achieved an accuracy of 85.74%, followed by DANN at 84.28%, PKMTCL at 79.00%, DXDA at 78.91%, D-CORAL at 76.07%, CNN-C at 72.85%, and DDTLN at 54.88%. ATCN exceeded the strongest comparison method, DANN, by 1.46 percentage points. These results show that ATCN achieved the highest accuracy among the evaluated methods on this cross-speed task.

3.5. Cross-Machine Fault Diagnosis Experiments

3.5.1. Hyperparameter Sensitivity Analysis

Task D6 (MFS1200 → Self1200) was used to examine the global alignment weight λ_{mda}, per-class pseudo-label limit K, and output consistency weight λ_{con}. These parameters control global adaptation strength, pseudo-label acceptance counts, and output constraint strength, respectively. One parameter was fixed while the other two were scanned to examine local responses and parameter interactions. Table 5 lists the evaluated values and fixed settings.

Each of the three 7 × 7 scan planes contains 49 configurations. Shared configurations were trained only once, leaving 127 distinct configurations after deduplication. All configurations used seed 42, with checkpoints selected using source-validation metrics. Figure 13 shows the source-validation ACC for each configuration. These results describe sensitivity and are used for hyperparameter selection.

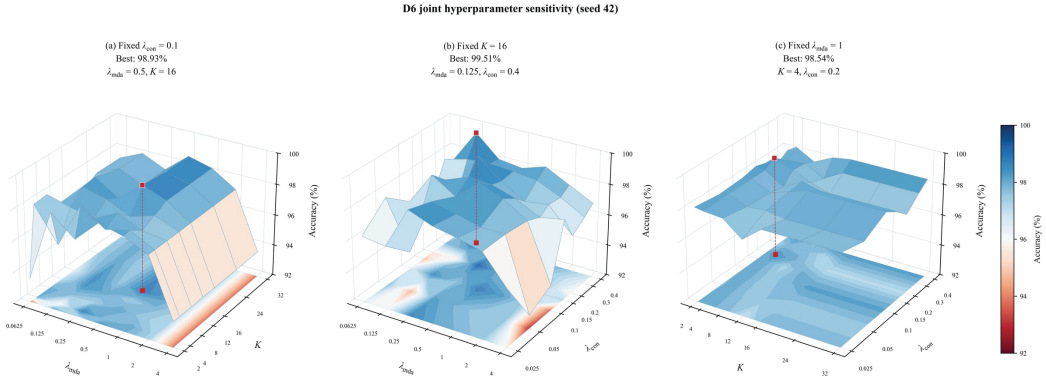


Figure 13. Joint hyperparameter sensitivity on D6. (**a**) $\lambda_{con} = 0.1$, varying λ_{mda} and K. (**b**) K = 16, varying λ_{mda} and λ_{con}. (**c**) $\lambda_{mda} = 1$, varying K and λ_{con}. Each panel contains 49 measured configurations, with 127 distinct configurations after deduplication, all using seed 42. Red squares mark the highest observed ACC in each plane. Surfaces and projections connect the observed grid, representing responses of the measured configurations. Weight axes use $\log_2$ spacing, while K uses a linear scale.

With $\lambda_{con} = 0.1$, ACC ranged from 93.55% to 98.93%, a difference of 5.37 percentage points. The maximum occurred at $\lambda_{mda} = 0.5$ and K = 16. The global alignment weight controls the strength of interdomain feature matching. K limits the target samples per class available for supervision and centroid computation. Both therefore influence class alignment. Actual acceptance counts also depend on the dynamic threshold, sample confidence, and predicted class distribution.

With K = 16, ACC ranged from 92.87% to 99.51%, a difference of 6.64 percentage points. At $\lambda_{mda} = 0.125$ and $\lambda_{con} = 0.4$, 1019/1024 samples were correctly classified, yielding the highest value in this scan. This shows that global alignment and output consistency weights require joint configuration. The D6 configuration was selected using source-validation ACC. The D6 comparison and ablation experiments used $\lambda_{mda} = 0.125$, K = 16, and $\lambda_{con} = 0.4$, with five random seeds.

With $\lambda_{mda} = 1$, ACC ranged from 96.97% to 98.54%, a difference of 1.56 percentage points. The maximum occurred at K = 4 and $\lambda_{con} = 0.2$. The three planes had different maxima, indicating that parameter effects depend on the fixed values of the other parameters. The value $\lambda_{con} = 0.4$ is the upper boundary of the scanned range. Task-specific values for the other transfer tasks were selected within the ranges in Table 5 using the same source-validation ACC criterion. The same candidate planes were applied to all eleven tasks. Table 9 reports representative selected configurations.

Table 9. Representative hyperparameter configurations for the twelve transfer tasks.

Task	λ_{mda}	K	λ_{con}
T1	0.0625	2	0.10
T2	0.0625	2	0.10
T3	0.0625	2	0.10
T4	0.0625	2	0.10
T5	0.0625	2	0.10
T6	0.0625	2	0.10
D1	0.0625	2	0.10
D2	0.0625	8	0.10
D3	0.0625	2	0.10
D4	0.25	8	0.10
D5	0.0625	2	0.10
D6	0.125	16	0.40

For T1–T6 and D1–D6, the search used seed 42. Where multiple configurations shared the highest source-validation ACC, one representative was selected by ascending λmda, K, and λcon.

3.5.2. Cross-Machine Comparative Experiments

ATCN achieved mean cross-machine ACC, Macro-F1, and Macro-AUROC values of 84.33%, 83.67%, and 90.14%, respectively. DXDA had the highest mean ACC and Macro-F1 among the comparison methods, at 60.95% and 58.74%. ATCN exceeded these by 23.38 and 24.93 percentage points. D-CORAL had the highest baseline mean Macro-AUROC, at 80.77%, which ATCN exceeded by 9.37 percentage points. Figures 14–16 show ACC, bidirectional Macro-F1, and ROC curves, respectively.

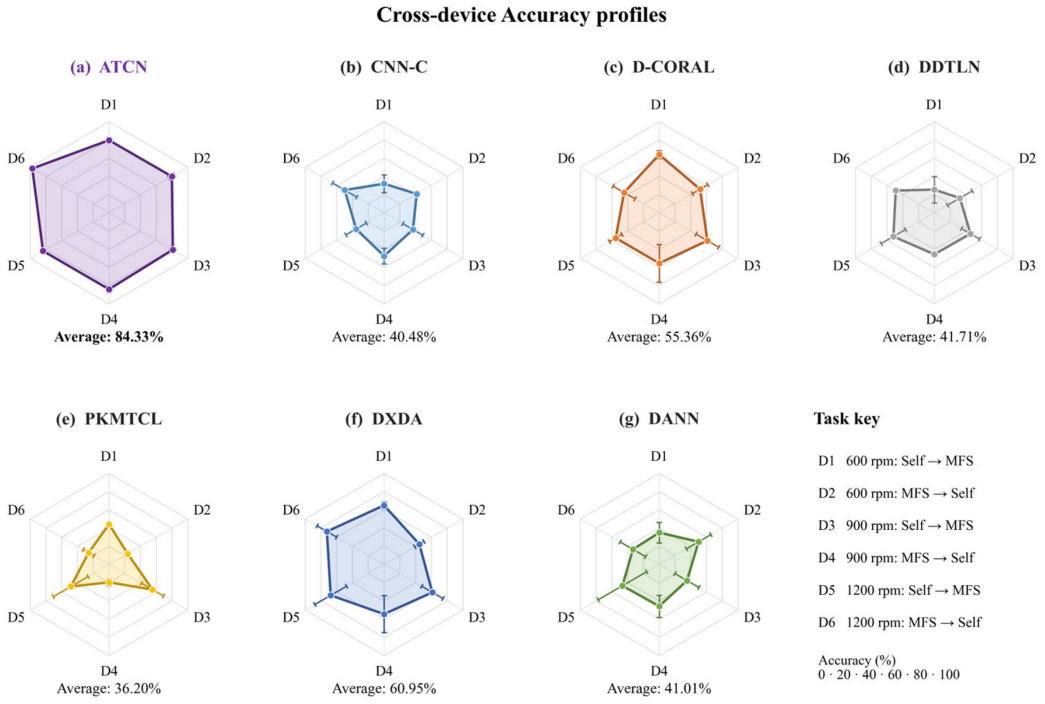


Figure 14. Cross-machine ACC. Points and polygons show five-seed means, with radial error bars indicating one sample standard deviation. All panels span 0–100%, with rings spaced by 20 percentage points. Average denotes the reported six-task aggregate.

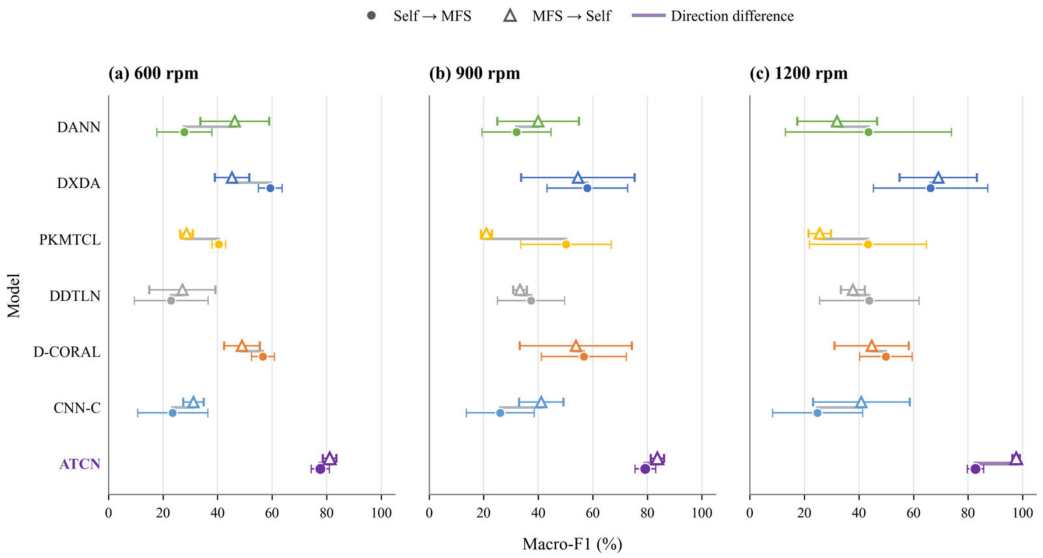


Figure 15. Bidirectional comparison of cross-machine Macro-F1. Panels correspond to 600, 900, and 1200 r/min. Points and error bars show five-seed means and sample standard deviations. Lines connect the two transfer directions for each method.

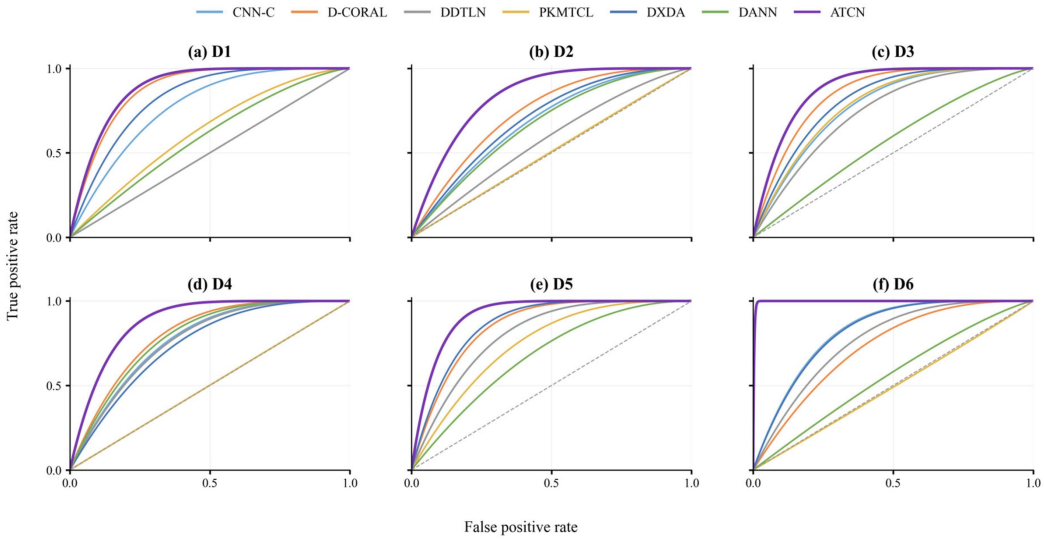


Figure 16. Single-run ROC curves for the six cross-machine tasks. Panels (**a**–**f**) correspond to D1–D6. Colors distinguish the seven methods. The dashed diagonal line in each panel represents chance-level performance (AUROC = 0.5).

In Figure 14, ATCN achieved ACC values of 79.79%, 79.82%, 81.02%, 83.96%, 83.71%, and 97.29% on D1–D6, respectively. These exceeded the strongest baselines on each task by 14.80, 27.93, 19.79, 28.57, 15.96, and 24.73 percentage points. At 600 and 900 r/min, DXDA was the strongest baseline for Self → MFS, whereas D-CORAL led in the reverse direction. ATCN outperformed the corresponding baselines in both directions. Its ACC values on D1–D5 ranged from 79.79% to 83.96%, whereas D6 reached 97.29%. The confusion matrices below show how these task differences relate to class-specific errors.

In Figure 15, ATCN achieved Macro-F1 values of 77.65%, 81.12%, 79.22%, 83.68%, 82.69%, and 97.65% on D1–D6, respectively. These exceeded the strongest directional baselines by 18.36, 32.16, 21.28, 29.13, 16.47, and 28.57 percentage points. At 600 r/min, D1 and D2 differed by only 0.03 percentage points in ACC but by 3.47 percentage points in Macro-F1. In the illustrated runs, D1 mainly misclassified OR5, whereas errors for Normal, IR8, and OR5 in D2 more frequently mapped to IR5. At 900 and 1200 r/min, Macro-F1 differences between opposite directions were 4.46 and 14.96 percentage points, respectively. The latter reflects IR8 confusion in D5 and the nearly diagonal structure of D6. DXDA was the strongest Macro-F1 baseline in five directions, while D-CORAL led on D2. The larger D5–D6 gap reflects the effect of transfer direction on class recognition.

Cross-machine tasks also show differences between ranking and classification decisions. D-CORAL had high Macro-AUROC values on D1–D4 but a mean ACC of 55.36%. On D6, CNN-C achieved 82.40% Macro-AUROC and 49.98% ACC. These results show that strong class-score ranking can coexist with misaligned decision boundaries. ATCN achieved higher mean ACC and Macro-AUROC than the strongest baselines on these cross-machine tasks.

In Figure 16, the ATCN ROC curve on D6 approaches the upper-left corner, while curves for D1–D5 vary with transfer direction. Mean Macro-AUROC on D6 across five runs was 99.68%.

Figures 17 and 18 show the class errors in the illustrated single-seed runs. On D1, ATCN's diagonal counts were 253, 204, 256, and 104 for the four classes. Only 104/256 OR5 samples were correctly identified, primarily accounting for the lower Macro-F1 on this task. On D2, Normal, IR8, and OR5 each had 69 samples assigned to IR5, producing class bias. On D3, 108/256 Normal samples were correctly identified, compared with 256/256 for both IR5 and OR5. Correct counts were 186/256 for OR5 on D4 and 146/256 for IR8 on D5, showing that difficult classes vary with direction. On D5, DXDA correctly identified 232/256 IR8 samples, more than ATCN, but performed less well on IR5 and OR5. ATCN recognized the other classes more evenly and therefore achieved higher Macro-F1. Its diagonal counts on D6 were 251, 245, 256, and 248. The matrices show that the main error patterns vary with transfer direction: OR5 confusion on D1, IR5 prediction bias on D2, and IR8 confusion on D5.

Cross-machine performance varies with transfer direction. D1 and D2 have similar overall accuracies but different class error distributions, whereas D6 recognizes the four classes relatively evenly. Mean cross-machine ACC is lower than cross-speed ACC, indicating that global and class-level adaptation is more challenging when the test rig changes.

Table 10 shows that, across all 12 transfer tasks, ATCN had a higher mean ACC than the strongest baseline selected for each task. Mean paired differences ranged from 1.19 to 10.70 percentage points for cross-speed tasks and from 14.80 to 28.57 percentage points for cross-machine tasks. After Holm correction, the differences for T2, T3, and D1 met the 0.05 significance threshold. The other tasks also showed positive mean differences, but their adjusted p-values did not meet this threshold. These findings indicate an observed performance advantage across the tasks, with statistical evidence varying by task.

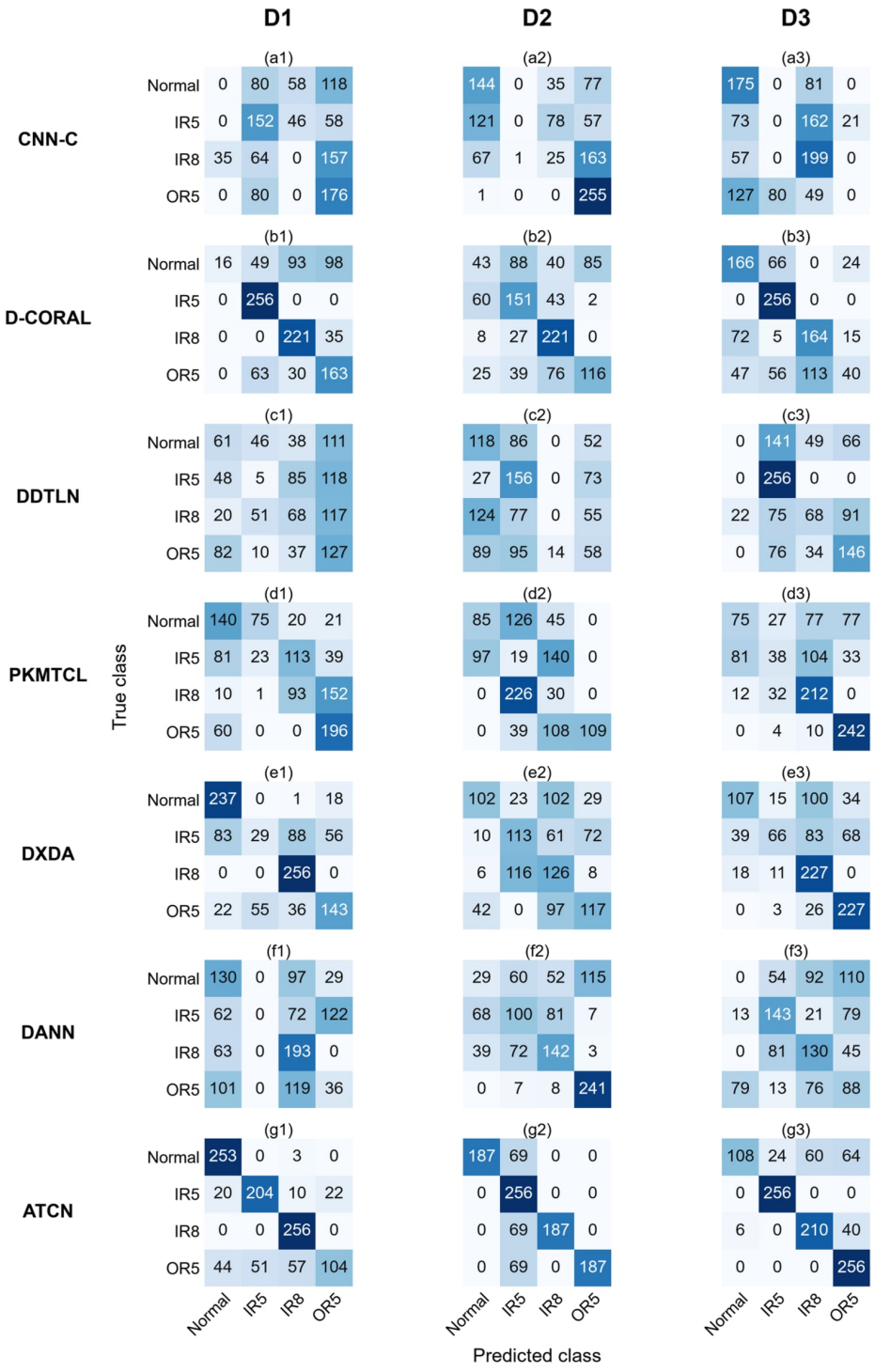


Figure 17. Single-run confusion matrices for cross-machine tasks D1–D3: (a1) CNN-C on D1; (a2) CNN-C on D2; (a3) CNN-C on D3; (b1) D-CORAL on D1; (b2) D-CORAL on D2; (b3) D-CORAL on D3; (c1) DDTLN on D1; (c2) DDTLN on D2; (c3) DDTLN on D3; (d1) PKMTCL on D1; (d2) PKMTCL on D2; (d3) PKMTCL on D3; (e1) DXDA on D1; (e2) DXDA on D2; (e3) DXDA on D3; (f1) DANN on D1; (f2) DANN on D2; (f3) DANN on D3; (g1) ATCN on D1; (g2) ATCN on D2; (g3) ATCN on D3. Matrix rows denote true classes and columns denote predicted classes, both ordered Normal, IR5, IR8, and OR5. Numbers in the cells indicate test-sample counts, with darker blue indicating larger counts.

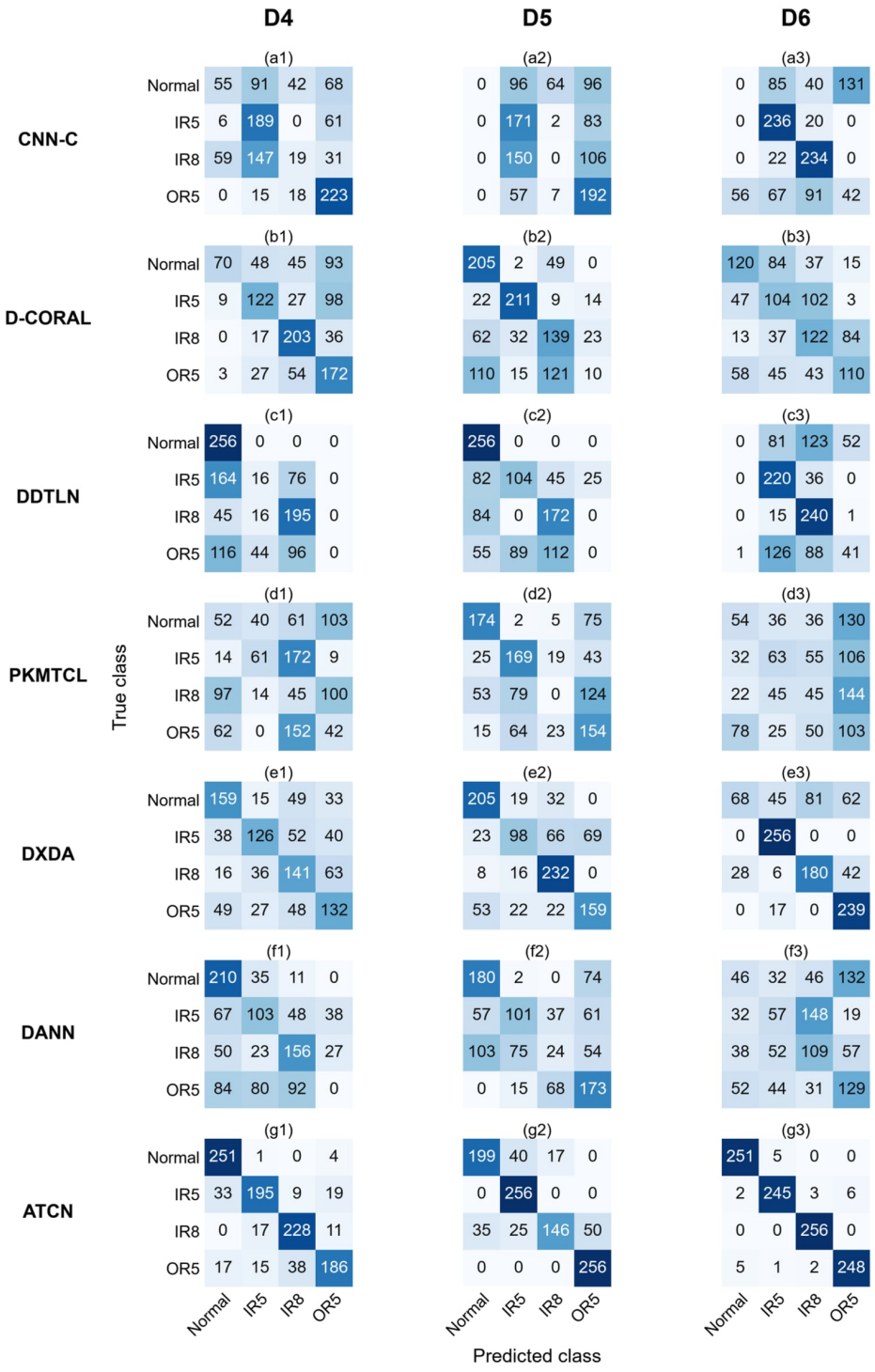


Figure 18. Single-run confusion matrices for cross-machine tasks D4–D6: (a1) CNN-C on D4; (a2) CNN-C on D5; (a3) CNN-C on D6; (b1) D-CORAL on D4; (b2) D-CORAL on D5; (b3) D-CORAL on D6; (c1) DDTLN on D4; (c2) DDTLN on D5; (c3) DDTLN on D6; (d1) PKMTCL on D4; (d2) PKMTCL on D5; (d3) PKMTCL on D6; (e1) DXDA on D4; (e2) DXDA on D5; (e3) DXDA on D6; (f1) DANN on D4; (f2) DANN on D5; (f3) DANN on D6; (g1) ATCN on D4; (g2) ATCN on D5; (g3) ATCN on D6. Matrix rows denote true classes and columns denote predicted classes, both ordered Normal, IR5, IR8, and OR5. Numbers in the cells indicate test-sample counts, with darker blue indicating larger counts.

Table 10. Paired ACC comparison between ATCN and the strongest baseline on each transfer task.

Task	Strongest Baseline	Mean Difference	95% CI	*p*-Value	Holm *p*-Value
T1	D-CORAL	9.16	[2.77, 15.55]	0.0164	0.0982
T2	DANN	10.45	[8.04, 12.86]	0.0003	0.0033
T3	DXDA	9.59	[4.99, 14.19]	0.0044	0.0442
T4	DXDA	4.69	[−1.02, 10.39]	0.0847	0.1693
T5	D-CORAL	10.70	[5.14, 16.27]	0.0059	0.0510
T6	DXDA	1.19	[0.10, 2.28]	0.0386	0.1542
D1	DXDA	14.80	[8.68, 20.92]	0.0026	0.0281
D2	D-CORAL	27.93	[13.59, 42.27]	0.0057	0.0510
D3	DXDA	19.79	[4.57, 35.01]	0.0226	0.1128
D4	D-CORAL	28.57	[0.62, 56.53]	0.0470	0.1542
D5	DXDA	15.96	[−8.09, 40.00]	0.1392	0.1693
D6	DXDA	24.73	[8.75, 40.71]	0.0127	0.0888

Mean differences and confidence intervals are in percentage points; positive values favor ATCN. Confidence intervals are unadjusted. *p*-values are from two-sided paired *t*-tests; Holm-adjusted *p*-values account for the 12 task-level comparisons. Each comparison uses five paired runs.

3.5.3. Module Ablation Experiments

The D6 ablation study compared full ATCN with three removal variants using the same five random seeds. The w/o PMLA variant disables global distribution alignment and class-centroid alignment while retaining pseudo-label classification. The w/o PBKM variant disables pseudo-label classification and its dependent class-centroid alignment while retaining global alignment. The w/o OCR variant disables target entropy and perturbation consistency terms while retaining source noise supervision.

In Table 11, full ATCN achieved a mean ACC of 97.29% ± 0.73%. Removing PMLA, the PBKM-dependent losses, and OCR reduced ACC to 77.23% ± 0.32%, 77.68% ± 4.21%, and 85.98% ± 1.79%, respectively. Removing PMLA decreased mean ACC by 20.06 percentage points, with relatively little variation across seeds. Figure 19a presents the ACC distributions across the five runs.

Table 11. Aggregated diagnostic results of module ablations on D6 (%, five-seed mean ± sample standard deviation).

Variant	ACC	Macro-F1	Macro-AUROC
Full ATCN	97.29 ± 0.73	97.65 ± 1.36	99.68 ± 0.38
w/o PMLA	77.23 ± 0.32	76.86 ± 0.59	81.17 ± 0.39
w/o PBKM	77.68 ± 4.21	77.27 ± 4.44	79.14 ± 4.52
w/o OCR	85.98 ± 1.79	85.96 ± 1.78	89.67 ± 0.48

After PBKM removal, Macro-F1 and Macro-AUROC were 77.27% and 79.14%, with paired mean decreases of 20.38 and 20.55 percentage points, respectively. The ACC standard deviation increased from 0.73 percentage points for the full model to 4.21 percentage points. The mean ACC decrease was 19.61 percentage points. Disabling pseudo-label supervision and its dependent class-centroid alignment reduces target class information. The resulting performance decline and increased run variability illustrate the role of this information pathway.

After OCR removal, the paired mean decreases in ACC, Macro-F1, and Macro-AUROC were 11.31, 11.69, and 10.02 percentage points, respectively. Disabling target output constraints reduced all three metrics despite retaining source noise supervision. This indicates different roles for source augmentation and target output regularization.

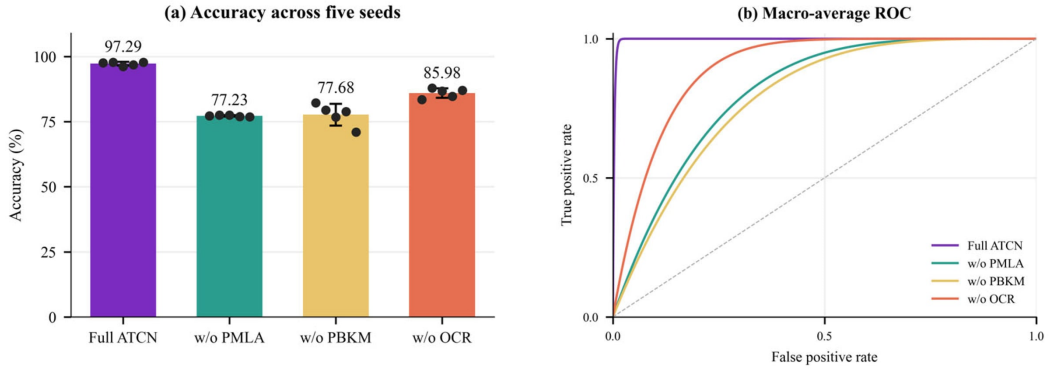


Figure 19. Module ablations on D6. (**a**) Bars and error bars show mean ACC and sample standard deviation across five runs; points represent individual runs. (**b**) Single-run macro-average ROC curves for each variant. The w/o PBKM variant also disables class-centroid alignment. The dashed diagonal line in panel (b) represents chance-level performance (AUROC = 0.5).

Figure 20 shows class changes after module removal in the illustrated runs. Removing PMLA reduced correct Normal and IR5 counts from 251 and 245 to 173 and 166. Errors spread across other fault classes, illustrating the role of global and class-centroid alignment in cross-domain class correspondence. Removing PBKM left 153 correctly identified Normal samples, with another 56 assigned to IR8. Without selected class information and its centroid alignment pathway, these two classes became more easily confused. Removing OCR reduced the correctly identified OR5 count to 205, with 27 samples assigned to Normal. Thus, class decision errors remain when only feature adaptation and pseudo-label supervision are retained. Together with the ranking decline in Figure 19, these results indicate that output constraints complement feature-level adaptation.

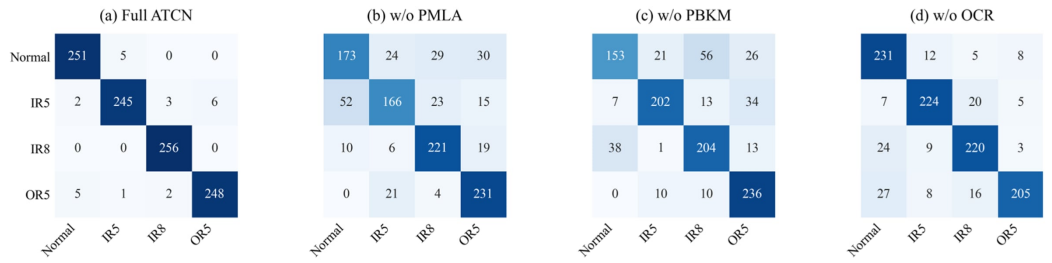


Figure 20. Single-run confusion matrices for the D6 ablation variants: (**a**) full ATCN; (**b**) w/o PMLA; (**c**) w/o PBKM; (**d**) w/o OCR. Rows indicate true classes and columns indicate predicted classes. Class order is Normal, IR5, IR8, and OR5. The w/o PBKM variant also disables class-centroid alignment. Panel (**a**) shows the same full ATCN confusion matrix for D6 as that presented in Figure 18.

To examine the individual contributions of the two PMLA levels and the centroid distance, additional D6 ablations were conducted using the same data protocol, training settings, and five seeds as the full model. The global-only variant retains global alignment and disables centroid alignment; the centroid-only variant does the reverse. Pseudo-label classification and OCR remain active. The Euclidean-only and cosine-only variants retain both alignment levels and use the respective single distance for centroid alignment. Full ATCN uses the learnable hybrid distance, while w/o PMLA disables both alignment terms.

As shown in Table 12, full ATCN achieved 97.29% ± 0.73% ACC, exceeding the global-only and centroid-only variants by 16.35 and 17.89 percentage points, respectively, and w/o PMLA by 20.06 points. These results support the benefit of combining global distribution matching with class-centroid alignment. Global matching addresses overall domain shift, while centroid alignment uses selected class information to maintain correspondence

between fault classes. ACC summarizes overall recognition across the fault classes; feature-level analysis can further examine class separation.

Table 12. Component and distance ablations of PMLA on D6 (ACC, %).

Variant	42	123	2024	3407	7777	Mean ± SD
Global only	78.52	77.54	82.52	81.45	84.67	80.94 ± 2.92
Centroid only	76.86	77.25	83.79	77.54	81.54	79.39 ± 3.10
Euclidean only	84.57	86.52	86.52	85.64	84.77	85.61 ± 0.93
Cosine only	85.55	84.47	82.71	80.96	84.67	83.67 ± 1.83
Full ATCN	97.66	97.85	96.19	96.88	97.85	97.29 ± 0.73
w/o PMLA	77.25	77.54	77.54	76.95	76.86	77.23 ± 0.32

Each result is based on five runs with seeds 42, 123, 2024, 3407, and 7777; SD denotes sample standard deviation. The individual ACC values are displayed to two decimal places. Summary statistics and mean differences were calculated before rounding.

The Euclidean-only and cosine-only variants achieved 85.61% ± 0.93% and 83.67% ± 1.83%, respectively. Full ATCN exceeded them by 11.68 and 13.61 percentage points, supporting the adaptive combination of centroid position and direction. Progressive global weighting and reliable pseudo-label selection help coordinate the alignment objectives when pseudo-labels are inaccurate or class structures differ across domains; these conditions may otherwise introduce conflicting updates.

3.6. Comparison of Diagnostic Accuracy and Inference Cost

To examine the relationship between diagnostic accuracy and inference cost, Table 13 compares ATCN with the six baseline methods. Cross-speed and cross-machine ACC values are the unweighted means across their respective six tasks and five random seeds. Inference cost was measured using the classification path of each model with four output classes and a 1024-point input.

Table 13. Comparison of diagnostic accuracy and inference cost for the seven methods.

Method	Cross-Speed ACC (%)	Cross-Machine ACC (%)	Inference Params (M)	Total Params (M)	Conv/Linear FLOPs (M)	Throughput (Samples/s)
CNN-C	52.38	40.48	0.9969	0.9969	87.33	28,363
D-CORAL	88.02	55.36	0.9969	0.9969	87.33	28,254
DDTLN	67.01	41.71	0.7920	0.7920	6.69	71,343
DXDA	90.71	60.95	0.7920	0.9238	6.69	72,669
PKMTCL	89.64	36.20	0.2298	0.2720	88.47	30,693
DANN	87.52	41.01	0.6596	0.7920	6.42	71,624
ATCN	99.36	84.33	0.7920	0.7920	6.69	71,535

Inference parameters count the trainable parameters used by the classification path, whereas total parameters also include auxiliary components used during training. FLOPs measure the computational cost of convolutional and linear layers, counting each multiply–accumulate operation as 2 FLOPs. Throughput is the number of samples diagnosed per second.

Throughput was measured on an NVIDIA GeForce RTX 5060 Laptop GPU using PyTorch 2.11.0 and CUDA 12.8, with FP32 inputs resident on the GPU and a batch size of 64. Each model was warmed up for 30 batches and timed over ten trials of 100 batches, with CUDA synchronized before and after each trial. Because its initial timing varied more, PKMTCL received 30 additional trials after a separate 50-batch warm-up; the reported throughput combines all 40 trials.

As shown in Table 13, ATCN achieved mean ACC values of 99.36% and 84.33% on the cross-speed and cross-machine tasks, respectively, exceeding the highest corresponding results among the comparison methods by 8.65 and 23.38 percentage points. ATCN used 0.7920 million inference parameters and 6.69 million convolutional and linear-layer FLOPs, matching DDTLN. Its throughput was 71,535 samples per second, close to those of DDTLN, DXDA, and DANN.

Among the evaluated methods, PKMTCL had the fewest inference parameters, while DANN had the lowest convolutional and linear-layer FLOPs. ATCN achieved the highest mean diagnostic accuracy in both transfer task families while maintaining inference costs comparable to those of DDTLN, DXDA, and DANN. Under the experimental and measurement conditions considered here, these results indicate a favorable balance between diagnostic accuracy and inference efficiency.

Inference cost was measured using each model's classification path; training and domain adaptation were excluded from the inference cost comparison.

4. Conclusions

This study proposes an Adaptive Transfer Consistency Network (ATCN) framework for multilevel cross-domain fault diagnosis of wind turbine bearings. The framework integrates progressive global alignment and adaptive class-centroid matching with selected pseudo-label supervision, target entropy minimization, and perturbation consistency. Across twelve HIT-SM transfer tasks, mean cross-speed ACC, Macro-F1, and Macro-AUROC were 99.36%, 99.36%, and 100.00% after rounding, respectively. The corresponding cross-machine values were 84.33%, 83.67%, and 90.14%. Mean ACC exceeded that of the strongest baseline by 8.65 and 23.38 percentage points for the two task families, respectively. On D6, removing PMLA, the PBKM-dependent losses, and OCR decreased mean ACC by 20.06, 19.61, and 11.31 percentage points, respectively. The comparative and ablation results support the benefit of coordinating alignment, target class information, and output regularization across the two task families.

The laboratory test-rig experiments provide a controlled evaluation of the framework. Future work will validate the framework using field-acquired vibration data under varying loads, environmental conditions, and naturally occurring bearing damage, and examine joint speed–machine changes. Class-wise pseudo-label acceptance statistics and prediction distributions will be used to refine sample selection and centroid alignment, with a focus on improving recognition of easily confused fault classes in cross-machine tasks. The target-domain split uses separate time segments of the same recording; leave-one-recording-out evaluation across independent target acquisitions remains future work. The present evaluation considers four bearing health states and uses offline adaptation with an unlabeled target-domain segment. Future work will extend the evaluation to additional fault conditions and sequentially acquired target data.

Author Contributions: Methodology, software, validation, formal analysis, data curation, visualization, writing—original draft, Z.Z.; supervision, writing—review and editing, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ATCN	Adaptive Transfer Consistency Network
PMLA	Progressive Multi-Level Adaptive Alignment
PBKM	Progressive Balanced Knowledge Mining
OCR	Output-Level Consistency Regularization
UDA	Unsupervised domain adaptation
1D-CNN	One-dimensional convolutional neural network
ACC	Classification accuracy
Macro-F1	Macro-averaged F1 score
Macro-AUROC	Macro-averaged area under the receiver operating characteristic curve

References

1. Peng, H.; Zhang, H.; Fan, Y.; Shangguan, L.; Yang, Y. A Review of Research on Wind Turbine Bearings' Failure Analysis and Fault Diagnosis. *Lubricants* **2023**, *11*, 14. [CrossRef]
2. Hart, E.; Raby, K.; Keller, J.; Sheng, S.; Long, H.; Carroll, J.; Brasseur, J.; Tough, F. *Main Bearing Replacement and Damage: A Field Data Study on 15 Gigawatts of Wind Energy Capacity*; NREL/TP-5000-86228; National Renewable Energy Laboratory: Golden, CO, USA, 2023. [CrossRef]
3. Zhao, W.; Jiang, Z.; Zhang, P.; Huang, X. Reliability Sensitivity Analysis of Main Shaft Bearings of Wind Turbines Subject to Subsurface Stress. *Machines* **2023**, *11*, 681. [CrossRef]
4. He, Y.; Huang, Y.; Hu, X.; Shen, W. A comprehensive review of compound fault diagnosis for industrial machinery: Object hierarchies, application scenarios and emerging perspectives on foundation models. *Adv. Eng. Inform.* **2027**, *77*, 105163. [CrossRef]
5. Zhao, Z.; Chen, F.; He, X.; Hu, X.; Zhao, Z.; Yin, X. Abnormal flow pattern identification for pumped-storage units toward intelligent operation and maintenance: Methods, system, and application validation. *Energy* **2026**, *362*, 142144. [CrossRef]
6. Chen, F.; Zhao, Z.; Hu, X.; Liu, D.; Yin, X. Multiscale root mean square for hydropower generation systems: A scalable feature extraction method for condition-based maintenance. *Renew. Energy* **2026**, *273*, 126106. [CrossRef]
7. Chen, J.; Pan, J.; Li, Z.; Zi, Y.; Chen, X. Generator bearing fault diagnosis for wind turbine via empirical wavelet transform using measured vibration signals. *Renew. Energy* **2016**, *89*, 80–92. [CrossRef]
8. Zhang, C.; Zhang, X.; Xu, Z.; Dai, W.; Lu, J. An Improved Denoising Method for Fault Vibration Signals of Wind Turbine Gearbox Bearings. *Machines* **2023**, *11*, 1004. [CrossRef]
9. Wang, X.; He, Y.; Wang, H.; Hu, A.; Zhang, X. A novel hybrid approach for damage identification of wind turbine bearing under variable speed condition. *Mech. Mach. Theory* **2022**, *169*, 104629. [CrossRef]
10. Ma, Z.; Zhao, M.; Luo, M.; Gou, C.; Xu, G. An integrated monitoring scheme for wind turbine main bearing using acoustic emission. *Signal Process.* **2023**, *205*, 108867. [CrossRef]
11. Wu, Z.; Yang, W.; Song, X.; Wei, K. Study on Condition Monitoring of Pitch Bearings Based on Stress Measurement. *Machines* **2024**, *12*, 154. [CrossRef]
12. Jiang, G.; Jia, C.; Nie, S.; Wu, X.; He, Q.; Xie, P. Multiview enhanced fault diagnosis for wind turbine gearbox bearings with fusion of vibration and current signals. *Measurement* **2022**, *196*, 111159. [CrossRef]
13. Bach-Andersen, M.; Rømer-Odgaard, B.; Winther, O. Deep learning for automated drivetrain fault detection. *Wind Energy* **2018**, *21*, 29–41. [CrossRef]
14. Chang, Y.; Chen, J.; Qu, C.; Pan, T. Intelligent fault diagnosis of Wind Turbines via a Deep Learning Network Using Parallel Convolution Layers with Multi-Scale Kernels. *Renew. Energy* **2020**, *153*, 205–213. [CrossRef]
15. Zhang, C.; Zhang, L. Wind turbine pitch bearing fault detection with Bayesian augmented temporal convolutional networks. *Struct. Health Monit.* **2024**, *23*, 1089–1106. [CrossRef]
16. Jiang, C.; Hu, X.; Huang, Y.; Wang, H.; Chen, Z.; Han, T. RJADNet: A Structure-Aware Multijoint Network with Topology-Constrained Aggregation for Industrial Robot Anomaly Detection with Compositional Generalization Under Imperfect Sensing. *IEEE Trans. Reliab.* **2026**, *75*, 2805–2819. [CrossRef]
17. Zhang, H.; Hu, X.; Cao, J.; Huang, Y.; Zhang, X.; Qi, J.; Chen, Z. Shapley-Proxy: Sensor Configuration for Railway Turnout Actuator Condition Monitoring Under Imperfect Sensing. *IEEE Trans. Instrum. Meas.* **2026**, *75*, 2513016. [CrossRef]
18. Yao, J.; Hu, X.; Jiang, C.; Huang, Y.; Wang, H.; Chen, Z. NPLNet: A Noise-Prior Lightweight Network for On-Device Acoustic Fault Diagnosis of Railway Point Machines From Noisy Measurements. *IEEE Trans. Instrum. Meas.* **2026**, *75*, 2511915. [CrossRef]
19. Tavares, T.M.; Giesbrecht, M. Deep Learning-Based Fault Diagnosis in Wind Turbine Bearings and Gearboxes Using Vibration Signals: Survey, Challenges, and Recommendations. *IEEE Access* **2025**, *13*, 207013–207032. [CrossRef]

20. Liang, P.; Wang, B.; Jiang, G.; Li, N.; Zhang, L. Unsupervised fault diagnosis of wind turbine bearing via a deep residual deformable convolution network based on subdomain adaptation under time-varying speeds. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105656. [CrossRef]
21. Jin, Y.; Feng, Q.; Zhang, X.; Lu, P.; Shen, J.; Tu, Y.; Wu, Z. Deep domain adversarial residual neural network for sustainable wind turbine cyber-physical system fault diagnosis. *Softw. Pract. Exp.* **2021**, *51*, 2128–2142. [CrossRef]
22. Wang, Q.; Taal, C.; Fink, O. Integrating Expert Knowledge with Domain Adaptation for Unsupervised Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
23. Su, H.; Yao, Q.; Li, X.; Xiang, L.; Hu, A. A Novel Transfer Learning Framework Based on Deep Contrastive Convolution for Wind Turbine Bearing Fault Localization. *IEEE Trans. Reliab.* **2025**, *74*, 5579–5591. [CrossRef]
24. Yang, Y.; Yin, J.; Zheng, H.; Li, Y.; Xu, M.; Chen, Y. Learn Generalization Feature via Convolutional Neural Network: A Fault Diagnosis Scheme Toward Unseen Operating Conditions. *IEEE Access* **2020**, *8*, 91103–91115. [CrossRef]
25. Sun, B.; Saenko, K. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Computer Vision—ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III*; Hua, G., Jégou, H., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9915, pp. 443–450.
26. Qian, Q.; Qin, Y.; Luo, J.; Wang, Y.; Wu, F. Deep discriminative transfer learning network for cross-machine fault diagnosis. *Mech. Syst. Signal Process.* **2023**, *186*, 109884. [CrossRef]
27. Wang, Q.; Michau, G.; Fink, O. Missing-Class-Robust Domain Adaptation by Unilateral Alignment. *IEEE Trans. Ind. Electron.* **2021**, *68*, 663–671. [CrossRef]
28. Liu, Z.; Huang, W.; Huang, Y.; Zhang, Y. Prior knowledge-informed multi-task collaborative learning for few-shot fault diagnosis of aero-engines. *Aerosp. Sci. Technol.* **2026**, *174*, 111864. [CrossRef]
29. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
30. Luo, H.; Wang, X.; Zhang, L. Normalization-Guided and Gradient-Weighted Unsupervised Domain Adaptation Network for Transfer Diagnosis of Rolling Bearing Faults Under Class Imbalance. *Actuators* **2025**, *14*, 39. [CrossRef]
31. Liu, C.; Hu, R.; Fang, X.; Luo, W.; Zhu, C. Enhancing Unsupervised Bearing Fault Diagnosis through Structured Prediction in Latent Subspace. *Sci. Rep.* **2025**, *15*, 42146. [CrossRef]
32. Zhang, S.-C.; Li, W.; Dong, L.; Liu, Z.; Yang, C.; Wang, X.; Jiang, C. Time-Frequency Hybrid Multilevel Feature Alignment Domain Adaptation Network for Cross-Condition Fault Diagnosis of Rolling Bearings. *Adv. Mech. Eng.* **2026**, *18*, 16878132261493401. [CrossRef]
33. Li, X.; Hu, B.; Dong, J.; Dong, Z. Industrial Internet-Oriented Unsupervised Hydro-Turbine Bearing Fault Diagnosis via Prototype-Disentangled Conditional Wasserstein Domain Adaptation. *Future Internet* **2026**, *18*, 428. [CrossRef]
34. Zhao, Z.; Xu, L.; Cai, J.; Zhu, X.; Bian, X. Multimodal Gated Fusion and Domain Adaptation for Cross-Condition High-Speed Train Bearing Fault Diagnosis. *Meas. Control* **2026**, *59*, 1436–1458. [CrossRef]
35. Ruan, H.; Pan, J.; Yang, J.; Zhu, H.; Huang, F. Domain-Specific Structured Reliability-Aware Open-Set Domain Adaptation for Cross-Condition Rolling Bearing Fault Diagnosis. *Appl. Sci.* **2026**, *16*, 7246. [CrossRef]
36. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep Subdomain Adaptation Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 1713–1722. [CrossRef]
37. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
38. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.-L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 596–608.
39. Zou, Y.; Yu, Z.; Vijaya Kumar, B.V.K.; Wang, J. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11207, pp. 297–313.
40. Grandvalet, Y.; Bengio, Y. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*; Saul, L.K., Weiss, Y., Bottou, L., Eds.; MIT Press: Cambridge, MA, USA, 2005; Volume 17, pp. 529–536.
41. Laine, S.; Aila, T. Temporal Ensembling for Semi-Supervised Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
42. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2818–2826.
43. HIT Sensing & Measurement Lab. Bearing-Datasets. Available online: https://github.com/hitwzc/Bearing-datasets (accessed on 13 October 2026).

44. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A Discriminative Feature Learning Approach for Deep Face Recognition. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9911, pp. 499–515.
45. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C.J., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25, pp. 2951–2959.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.