Reducing Negative Transfer in Domain Adaptation for Vibration Fault Diagnosis

Pawel Knap[1], Urszula Jachymczyk[2]

[1,2] *AGH University of Krakow, Kraków, 30-059, Poland*
pknap@agh.edu.pl
ujachymczyk@agh.edu.pl

ABSTRACT

Unsupervised domain adaptation (UDA) for vibration-based fault diagnosis can improve transfer across changing operating conditions, but its reliability remains a practical concern. In particular, large domain shifts can lead to negative transfer, where an adapted model underperforms a source-only baseline. This motivates evaluation beyond average gains, with emphasis on worst-case behavior and failure modes relevant to deployment. This study proposes a lightweight safeguard for discrepancy-based UDA that does not require labeled target data. The approach augments standard adaptation with an unlabeled monitoring rule based on target prediction entropy and alignment-loss trends. When adaptation appears unstable, training is paused and the model is rolled back to a safer checkpoint. The safeguard is designed as a small reliability layer on top of existing UDA pipelines rather than as a new adaptation method. We evaluate source-only training, standard UDA, source-pretrained UDA, and safeguarded UDA on the Case Western Reserve University (CWRU) and Paderborn University (PU) bearing datasets under multiple cross-condition transfer tasks. Experiments include raw time-domain, FFT-based, and STFT-based representations with MMD- and CORAL-based adaptation. Results show that negative transfer is a repeatable phenomenon, particularly on more challenging CWRU shifts, while source-pretrained UDA substantially affects reliability. The safeguard shows partial mitigation of harmful adaptation in selected PU cases but does not consistently prevent degradation across all scenarios. Overall, the results highlight that monitoring adaptation dynamics can improve reliability in some settings, but that safe deployment of UDA for fault diagnosis still requires explicit consideration of worst-case behavior and baseline comparisons.

1. INTRODUCTION

Unsupervised domain adaptation (UDA) has become an attractive strategy for vibration-based fault diagnosis because it allows models trained on labeled source data to generalize to target domains while mitigating performance degradation caused by varying operating conditions (Wagner & Sommer, 2021; Azari, Flammini, Santini, & Caporuscio, 2023). This topic quickly evolved toward domain adaptation with deep feature learning in a single training process (Ganin & Lempitsky, 2015). In rotating machinery diagnostics, this setting is especially relevant because vibration signals vary with operating speed, load, sensor location, machine configuration, and fault progression, causing a distribution mismatch between the source and target domains (X. Chen et al., 2023; Z. Chen et al., 2021). As a result, a growing body of work has applied discrepancy-based and adversarial adaptation methods to cross-condition fault diagnosis problems (Ragab et al., 2021; Li et al., 2021; Ragab et al., 2023).

Despite this progress and the great potential of UDA methods, they are not guaranteed to improve performance. Negative transfer, i.e., the case where transferred knowledge reduces target-domain performance, is a well-recognized challenge in transfer learning (Zhang, Deng, Zhang, & Wu, 2023). In domain adaptation, severe source–target discrepancies can make alignment harmful rather than beneficial, especially when the learned representation becomes more transferable at the expense of discriminability (X. Chen, Wang, Long, & Wang, 2019). This concern is directly relevant to fault diagnosis under changing operating conditions, where several studies explicitly motivate method design or training control in order to avoid negative transfer (Wei, Han, Chu, & Zuo, 2021; Z. Wang et al., 2024). Cross-condition shifts may alter not only low-level signal statistics but also the class structure relevant for fault recognition, and adaptation methods may underperform non-transfer baselines in difficult transfer scenarios.

From a practical perspective, this means that average adaptation gains are not sufficient to establish trust in UDA. A prac-

tically useful UDA system should also limit worst-case degradation relative to a strong source-only baseline. A method that improves mean performance but occasionally causes large degradation may still be risky in deployment, particularly in predictive maintenance settings where false confidence can lead to missed faults or unnecessary interventions (Hoffmann & Lasch, 2025).

1.1. Research Gap

Although UDA for bearing fault diagnosis is now well established, most existing studies focus on improving adaptation performance rather than controlling adaptation risk. In particular, there is still limited work that treats *negative transfer* as the primary object of analysis and investigates whether harmful adaptation can be detected and prevented without target labels (Zhang et al., 2023; Z. Wang et al., 2024). Existing methods usually report target-domain gains after adaptation, but much more rarely ask whether adaptation should be trusted in the first place. Negative transfer can remain hidden if the evaluation focuses only on mean gains and does not compare adapted models against a source-only reference. For this reason, reliability-oriented UDA should not ask only whether adaptation helps on average, but also how often it becomes harmful, how severe the degradation is, and whether such harmful adaptation can be detected early without access to target labels.

In particular, there is still no widely adopted protocol that simultaneously provides:

- leakage-safe run-level or acquisition-level partitioning for UDA experiments,
- explicit comparison against a source-only baseline under fixed cross-domain scenarios,
- direct analysis of how often and how severely negative transfer occurs under large cross-condition shifts, and
- a practical safeguard capable of suppressing harmful adaptation without target labels.

1.2. Contributions

To address possible UDA-induced prediction degradation, we introduce a lightweight safeguard mechanism that monitors adaptation dynamics using only unlabeled target signals available during training. The mechanism tracks quantities such as prediction entropy and alignment-loss trends and, when it detects a high-risk trajectory, stops adaptation and reverts to the source-trained model. The main practical issue is not only average improvement, but also negative transfer, where adaptation falls below a source-only baseline. In this way, the proposed approach aims not simply to maximize adaptation gains, but to reduce the risk of negative transfer.

The main contributions of this work are as follows:

- We study source-only, plain UDA, source-pretrained UDA, and guarded UDA settings under fixed cross-condition transfer scenarios.
- We propose a lightweight safeguard mechanism that monitors unlabeled adaptation dynamics using target prediction entropy and alignment-loss trends.
- We implement an automatic rollback strategy that reverts to the source-trained model when adaptation is assessed as high-risk.
- We evaluate the approach on CWRU and Paderborn University datasets, analyzing both average-case gains and worst-case degradation.
- We identify scenarios in which the safeguard improves reliability, as well as cases where it remains unstable and requires further development.

The study is designed to answer the following research questions:

- **RQ1:** Under run-level evaluation, how often does negative transfer occur relative to a source-only baseline across cross-condition UDA scenarios?
- **RQ2:** How does source pretraining influence the interpretation of UDA performance gains and failures?
- **RQ3:** Can a lightweight unlabeled safeguard detect harmful adaptation dynamics and reduce worst-case performance degradation without target labels?
- **RQ4:** Where does the current guardrail remain unstable across datasets or representations?

This study compares source-only training, plain UDA, source-pretrained UDA, and a lightweight guardrail-based adaptation strategy across multiple cross-domain fault-diagnosis scenarios. The comparison is conducted using three input representation–model combinations, namely raw time-domain signals, FFT-based representations, and STFT-based time–frequency representations. In addition to comparing overall adaptation performance, the study examines the occurrence and severity of negative transfer and evaluates the effectiveness of the proposed guardrail in mitigating harmful adaptation.

2. METHODOLOGY

This section describes the methodological framework used in the study. It first formulates the unsupervised domain adaptation problem considered in the experiments, then presents the proposed guardrail mechanism, the signal representations and the model architectures used.

2.1. Problem Formulation

In UDA, a model is trained using a labeled source domain and an unlabeled target domain. The source domain is defined as:

$$\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}, \tag{1}$$

while the target domain is defined as:

$$\mathcal{D}_t = \{x_j^t\}_{j=1}^{N_t}. \tag{2}$$

where:
- x_i^s denotes the i-th source-domain input window,
- $y_i^s \in \{1, \ldots, C\}$ is a class label corresponding to i-th source-domain input window,
- N_s is the number of source samples,
- x_j^t denotes the j-th target-domain input window,
- N_t is the number of target samples.

Target labels are assumed unavailable during adaptation. The symbol C denotes the number of classes, which is fixed to four in all experiments.

The goal of UDA is to reduce the domain gap between these two domains while preserving discriminative performance on the source task. A general adaptation objective can be written as:

$$\mathcal{L}_{\text{UDA}} = \mathcal{L}_{\text{cls}}(\mathcal{D}_s) + \lambda \mathcal{L}_{\text{align}}(\mathcal{D}_s, \mathcal{D}_t), \tag{3}$$

where:
- $\mathcal{L}_{\text{cls}}$ denotes the supervised classification loss on the source domain,
- $\mathcal{L}_{\text{align}}$ is a domain alignment term, and
- λ controls the trade-off between classification and alignment.

In this work, $\mathcal{L}_{\text{align}}$ is instantiated using either CORAL (Sun, Feng, & Saenko, 2016; Sun & Saenko, 2016) or MMD (Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2006; W. Wang, Li, Ding, & Wang, 2020). CORAL reduces domain shift by aligning the second-order statistics, specifically the covariance structure, of source and target features. In contrast, MMD measures and minimizes the discrepancy between source and target feature distributions in a reproducing kernel Hilbert space. The model is trained to jointly minimize the supervised classification loss on the labeled source domain and the domain alignment loss between the source and target domains, so that domains become less separable.

In addition to direct adaptation, the research also includes evaluation of a source-pretrained regime. In this setting, the model is first trained using only the source-domain classification loss, $\mathcal{L}_{\text{cls}}(\mathcal{D}_s)$, and is then fine-tuned using the full objective in Eq. (3). This source-only checkpoint serves as a reference baseline and a safety point for evaluating the effect of domain adaptation.

2.2. Guardrail Algorithm

The final safeguard in this constrained setting is a lightweight unlabeled monitor applied during source-pretrained domain adaptation. It tracks target prediction entropy, alignment-loss progress, and an adaptation-strength proxy to detect potentially risky adaptation dynamics.

The entropy monitor is based on the intuition that, when adaptation moves target samples toward better separated class regions, the classifier should become more confident on unlabeled target data. For a four-class problem, a completely uniform prediction has entropy $\log(4) \approx 1.386$, whereas a near-deterministic prediction has entropy close to zero. Thus, decreasing target entropy can indicate that target samples are moving away from decision boundaries. However, entropy is only a proxy for target reliability and is not guaranteed to be monotonically related to target accuracy. A model can be confidently wrong, especially under severe domain shift or class-mismatched alignment. For this reason, the proposed guardrail does not treat low entropy alone as sufficient evidence of successful adaptation. Instead, it combines entropy regression with alignment-loss stagnation and an adaptation-strength condition before triggering rollback.

After source-only pretraining, a baseline target entropy is computed as:

$$h_0 = -\frac{1}{N_t} \sum_{j=1}^{N_t} \sum_{c=1}^{C} p_{\theta_0}(c \mid x_j^t) \log p_{\theta_0}(c \mid x_j^t), \tag{4}$$

where:
- θ_0 denotes the source-pretrained model parameters,
- $p_{\theta_0}(c \mid x_j^t)$ denotes the predicted probability of class c for the target sample x_j^t under the source-pretrained model and c indexes the classes from 1 to C.

Next, during the adaptation stage, the model is optimized according to the following equation:

$$\mathcal{L}_{\text{UDA}}^{(e)} = \mathcal{L}_{\text{cls}}^{(e)} + \lambda \mathcal{L}_{\text{align}}^{(e)}, \tag{5}$$

where:
- $\mathcal{L}_{\text{cls}}^{(e)}$ is the source-domain classification loss at epoch e, and
- $\mathcal{L}_{\text{align}}^{(e)}$ is the domain alignment loss at epoch e.

At each adaptation epoch e, the target prediction entropy is computed as:

$$h_e = -\frac{1}{N_t} \sum_{j=1}^{N_t} \sum_{c=1}^{C} p_{\theta_e}(c \mid x_j^t) \log p_{\theta_e}(c \mid x_j^t), \tag{6}$$

and the adaptation-strength proxy is defined as the ratio between domain alignment loss and source classification loss:

$$s_e = \frac{\lambda \mathcal{L}_{\text{align}}^{(e)}}{\mathcal{L}_{\text{cls}}^{(e)}}. \tag{7}$$

Lower target prediction entropy generally indicates that the model is making more confident predictions on unlabeled target samples. For this reason, the guardrail algorithm tracks the best entropy achieved across all epochs:

$$h_{\min}^{(e)} = \min(h_0, h_1, \ldots, h_{e-1}), \tag{8}$$

and computes the best achieved entropy gain relative to the source-pretrained baseline:

$$\Delta h_{\text{arm}}^{(e)} = h_0 - h_{\text{min}}^{(e)}. \tag{9}$$

The monitor is *armed* only if

$$\Delta h_{\text{arm}}^{(e)} > \tau_{\text{arm}}, \tag{10}$$

where τ_{arm} denotes the minimum entropy improvement required before rollback can be considered. In this research, the minimum safety improvement was set to $\tau_{\text{arm}} = 0.05$, so the guardrail is armed only after a non-trivial entropy improvement over the source-pretrained baseline.

Once armed, the guardrail measures, at each epoch, the regression of target entropy relative to the best target entropy achieved so far during adaptation:

$$\Delta h_{\text{reg}}^{(e)} = h_e - h_{\text{min}}^{(e)}, \tag{11}$$

A positive increase in entropy suggests that adaptation may be becoming less reliable. Entropy is therefore marked as risky only when:

$$\Delta h_{\text{reg}}^{(e)} > \tau_h, \tag{12}$$

where τ_h is the entropy regression tolerance margin.

Let W denote the epoch-level comparison window. At epoch e, the method computes the mean alignment loss over the W epochs immediately preceding the recent block,

$$\bar{d}_{\text{prev}}^{(e)} = \frac{1}{W} \sum_{i=e-2W}^{e-W-1} d_i, \tag{13}$$

and the mean alignment loss over the recent block consisting of the last W stored epochs together with the current epoch,

$$\bar{d}_{\text{recent}}^{(e)} = \frac{1}{W+1} \sum_{i=e-W}^{e} d_i, \tag{14}$$

where $d_e = \mathcal{L}_{\text{align}}^{(e)}$ denotes the alignment loss recorded at epoch i. The relative alignment improvement is then defined as

$$r_d^{(e)} = \frac{\bar{d}_{\text{prev}}^{(e)} - \bar{d}_{\text{recent}}^{(e)}}{\left|\bar{d}_{\text{prev}}^{(e)}\right| + \epsilon}. \tag{15}$$

Here, ϵ is a small positive constant used only for numerical stability and prevents division by zero.

The method treats alignment improvement as insufficient when

$$r_d^{(e)} < \tau_d, \tag{16}$$

where τ_d specifies the minimum relative decrease in alignment loss required for adaptation to be considered still making meaningful progress.

A bad epoch is declared only when all of the following hold:

$$e \geq E_{\text{start}}, \tag{17}$$

$$s_e \geq \tau_s, \tag{18}$$

$$\Delta h_{\text{arm}}^{(e)} > \tau_{\text{arm}}, \tag{19}$$

$$\Delta h_{\text{reg}}^{(e)} > \tau_h, \tag{20}$$

$$r_d^{(e)} < \tau_d, \tag{21}$$

where E_{start} is the first epoch at which guardrail decisions are allowed and τ_s is the minimum adaptation strength required to treat adaptation as active. In other words, an epoch is tagged as bad only when adaptation is active, the guardrail has been armed by a sufficient entropy improvement over the source-pretrained baseline, target entropy has regressed from its best value, and alignment improvement has become insufficient.

After the guardrail is armed, the method stores a *best adapted checkpoint*, but only from epochs that are both (i) at or after E_{start} and (ii) not already marked as bad epochs. Among those eligible epochs, it keeps the checkpoint with the lowest target entropy. If bad epochs persist for P consecutive epochs, where P is the patience, the guardrail triggers and restores the epoch with the lowest target entropy (best predictions on target data) if it is available, or source-pretrained epoch only, if not.

$$\theta_{\text{rollback}} = \begin{cases} \theta_{\text{best-adapted}}, & \text{if an eligible adapted checkpoint exists,} \\ \theta_{\text{source-pretrain}}, & \text{otherwise.} \end{cases} \tag{22}$$

In practical terms, the guardrail follows a two-stage logic:

- First, it waits until adaptation has produced a meaningful decrease in target entropy (confidence of model on target data increases) compared with the source-pretrained baseline, which arms the monitor and identifies that an adapted state may be useful. If entropy never improves enough to arm the guardrail, then the guardrail remains inactive — no rollback is triggered by this mechanism and the run would continue as ordinary source-pretrained UDA.
- Second, after the monitor is armed, it checks whether target entropy starts to increase again while the alignment loss no longer decreases sufficiently.

Such a combination suggests that adaptation may be moving away from a previously better target-domain state while no longer making useful alignment progress.

2.3. Signal Representations and Models

The study compares three models that share the same downstream classification objective but differ in the frontend representation of the input data. This keeps the supervision tar-

get fixed while allowing to assess how domain adaptation behaves under different encodings of the same vibration data.

All three models produce a learned feature representation that is mapped to class logits through a small classifier head. The classifier head is a two-layer multilayer perceptron with a hidden dimension of 64, dropout 0.4, and a final linear layer producing class logits.

Raw Time-Domain Model. The raw baseline uses a 1D convolutional neural network applied directly to the windowed vibration signal. The feature extractor consists of three convolutional blocks with channel widths 32, 64, and 128, followed by batch normalization, ReLU activations, and adaptive average pooling to a single temporal feature vector.

FFT-Based Model. The FFT model applies a real-valued fast Fourier transform directly along the temporal axis before classification. The resulting log-amplitude spectrum is then passed through the same 1D convolutional backbone pattern used by the raw model. This model is intended to test whether a frequency-domain representation improves robustness under sensor or operating-condition shift.

STFT-Based Model. The STFT model uses a time–frequency representation computed from input window. Its frontend applies the short-time Fourier transform, after which the resulting representation is processed by a 2D convolutional network with three convolutional stages, max-pooling in the first two stages, and adaptive average pooling before the classifier head. In the final implementation, the STFT front-end uses a Hann window with $n_{\mathrm{fft}} = 1024$ and the default hop length $h = n_{\mathrm{fft}}/4$.

3. EXPERIMENTAL SETUP

Because the main objective of the study is to assess reliability under domain shift and the occurrence of negative transfer, the experimental analysis is organized around three settings: source-only training, plain UDA, and a lightweight guardrail prototype. The analysis is conducted on two widely used bearing fault diagnosis datasets: the Case Western Reserve University (CWRU) dataset and the Paderborn University (PU) dataset. The CWRU dataset provides controlled vibration measurements under varying load and sensor configurations (Smith & Randall, 2015), while the PU dataset contains both artificially induced and naturally developed bearing faults acquired under a range of operating conditions (Lessmeier, Kimotho, Zimmer, & Sextro, 2016). The classification task is formulated as a four-class fault-category problem for both datasets, as summarized in Table 1. Fault size or damage severity is not used as an additional classification target in this study.

Table 1. Mapping of fault types to classes for the CWRU and PU datasets.

Class	CWRU fault type	PU fault type
0	No fault	No fault
1	Ball fault	Inner race fault
2	Inner race fault	Outer race fault
3	Outer race fault	Multiple faults

For the CWRU cross-load scenarios, the changing load also directly influences the rotational speed, what affects the spectral location of fault-related components and can therefore influence both feature distributions and adaptation behavior. Thus, these experiments should be interpreted as combined load/speed domain shifts.

3.1. Experiment Matrix

Table 2 summarizes the dataset and scenario matrix used in the study. It includes cross-sensor and cross-load experiments on the CWRU dataset, as well as cross-speed and cross-fault-generation (*artificial-to-real*) experiments on the PU dataset. Guardrail experiments were not conducted on the CWRU cross-sensor setting because it did not exhibit strong negative transfer on the primary metric. Instead, the guarded subset was chosen to include both clearly harmful and clearly beneficial cases, allowing the safeguard to be evaluated on failure cases and control cases.

Three model–representation pairs were evaluated: a raw time-domain 1D CNN, an FFT-based 1D CNN, and an STFT-based 2D CNN.

3.2. Data Split and Domain Definition

For the source-only setting, the model is trained on the source-domain subset and evaluated on the target-domain subset. For the UDA setting, the same source and target subsets are used, but the target-domain data are provided without labels during adaptation and used only for domain alignment; final performance is then measured on the target domain using the saved model checkpoint.

For the PU dataset, only the `vibration_1` channel from the `HostService` raster was retained to ensure consistency with the vibration-based study. The PU dataset contains measurements from both artificially induced defects and naturally developed fatigue damage. Accordingly, the *Artificial→Real* scenario uses a fixed set of healthy bearings (`K001`, `K002`, `K003`, `K004`, `K005`, `K006`) together with selected artificially damaged bearings (`KA01`, `KA03`, `KA05`, `KA06`, `KA07`, `KA08`, `KA09`, `KI01`, `KI03`, `KI05`, `KI07`, `KI08`) and fatigue-damaged bearings (`KA04`, `KA15`, `KA16`, `KA22`, `KA30`, `KI04`, `KI14`, `KI16`, `KI17`, `KI18`, `KI21`), all acquired under the same operating condition, `N15_M07_F10`, corresponding to a rotational speed of 1500 rpm, an applied torque of 0.7 Nm, and a radial load of 1000 N on the bearing.

Table 2. Experiment matrix used in the reliability study.

Dataset	Scenario	Source domain	Target domain	Role in study
CWRU	Cross-sensor	DE sensor, 12 kHz, 1750 rpm	FE sensor, 12 kHz, 1750 rpm	baseline benchmark
CWRU	Load $0 \rightarrow 3$	DE sensor, 12 kHz, 1797 rpm	DE sensor, 12 kHz, 1730 rpm	baseline + guardrail subset
CWRU	Load $3 \rightarrow 0$	DE sensor, 12 kHz, 1730 rpm	DE sensor, 12 kHz, 1797 rpm	baseline + guardrail subset
PU	Cross-speed	vibration_1, N15_M07_F10	vibration_1, N09_M07_F10	baseline + guardrail subset
PU	Artificial→Real	vibration_1, artificial faults at N15_M07_F10	vibration_1, fatigue faults at N15_M07_F10	baseline + guardrail subset

3.3. Training Hyperparameters

The experimental setup uses a common training budget across all models and datasets. Specifically, all experiments use a window size of 2048 samples, a training overlap of 0.25, a test overlap of 0.25, 50 training epochs, a batch size of 256, and a learning rate of 5×10^{-4}.

Optimization is performed with AdamW using a weight decay of 10^{-4} and an exponential learning-rate scheduler. Other shared settings include label smoothing of 0.05, `num_workers`=0, mixed precision disabled, and a fixed random seed of 42.

For plain UDA, two alignment methods are considered. MMD uses $\lambda_{\mathrm{DA}} = 1.0$ with `kernel_num`=5 and `kernel_mul`=2.0, whereas CORAL uses $\lambda_{\mathrm{DA}} = 500.0$, a value chosen to prevent the alignment term from becoming effectively negligible.

3.4. Source-Pretrained and Guardrail Settings

In the source-pretrained UDA setting, training is carried out in two stages. First, the model is trained only on the labeled source-domain data, using the standard supervised classification objective. This produces a source-trained initialization that serves as the starting point for adaptation. In the second stage, the model is further optimized with the full UDA objective, which combines source-domain classification with source–target domain alignment. In the guardrail experiments, the source-pretraining stage lasts for 50 epochs before adaptation begins.

The guardrail configuration used in the final experiments is as follows:

- `start_epoch` ($E_{\mathrm{start}} = 5$): guardrail decisions are enabled from epoch 5 onward,
- `patience` ($P = 2$): rollback is triggered after two consecutive degraded epochs,
- `compare_window` ($W = 2$): defines the window for comparing alignment-loss trends,
- `entropy_margin` ($\tau_h = 0.02$): tolerance for entropy regression,
- `min_entropy_improvement_to_arm` ($\tau_{\mathrm{arm}} = 0.05$): minimum entropy improvement required to arm the guardrail,
- `min_adaptation_strength` ($\tau_s = 0.01$): threshold for considering adaptation active, and
- `min_relative_domain_improvement` ($\tau_d = 0.02$): minimum relative improvement in alignment loss required to avoid declaring stalled progress.

These parameters were chosen to make the guardrail conservative rather than highly reactive. The arming threshold τ_{arm} requires a non-trivial entropy improvement before rollback can even be considered, while τ_h defines how much regression from the best observed target entropy is tolerated before the epoch is treated as risky. The activity threshold τ_s and the alignment-progress threshold τ_d prevent rollback from being triggered when adaptation is either effectively inactive or still making meaningful progress.

The guardrail thresholds and stopping criteria should be interpreted as heuristic hyperparameters of the prototype rather than as theoretically optimal constants resulting from a leakage-safe hyperparameter optimization. The final values were fixed after iterative prototype development on a small guarded subset and then kept unchanged in the experiments. They were chosen to make the mechanism conservative: rollback is allowed only after a non-trivial entropy improvement, only when adaptation is active, and only when alignment progress appears insufficient. Therefore, the results may be sensitive to these values, especially across datasets, signal representations, and adaptation losses. The present study does not claim that the selected thresholds generalize universally.

Once the guardrail is armed, the rollback policy restores the best eligible adapted checkpoint. If no eligible adapted checkpoint is available, the method falls back to the source-pretrained checkpoint.

4. RESULTS

The primary evaluation metric is macro-F1, with accuracy and balanced accuracy reported as supporting metrics. Macro-F1 is used as the primary metric because it gives equal weight to each class and is therefore more informative than accuracy when class-wise performance differs. For class $c \in \{1, \ldots, C\}$, the F1-score is defined as:

$$\mathrm{F1}_c = \frac{2TP_c}{2TP_c + FP_c + FN_c}, \tag{23}$$

where TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c, respectively. The

macro-F1 score is then computed as

$$\mathrm{MacroF1} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{F1}_c. \tag{24}$$

Results are presented at two levels. First, the full baseline experiment set is reported for each scenario and summarized using average-case and worst-case performance relative to the source-only reference, in order to assess whether adaptation is beneficial or harmful under domain shift. Second, for the selected guardrail subset, results are reported as performance differences relative to plain UDA and source-only, together with trigger and rollback information, to evaluate whether the safeguard can reduce harmful adaptation without unnecessarily suppressing beneficial transfer.

Across the final experiment set, three high-level patterns emerge. First, unsupervised domain adaptation is not uniformly beneficial: negative transfer appears clearly on CWRU and in isolated PU settings, with strong dependence on scenario and representation. Second, source-pretrained UDA materially changes the interpretation of several guarded runs and must therefore be treated as a separate comparison regime rather than an implementation detail. Third, the current guardrail prototype shows encouraging evidence on PU but does not consistently rescue harmful CWRU cases, so its present value is best interpreted as prototype evidence rather than a generally reliable safeguard.

4.1. Baseline Negative Transfer

Tables 3 and 4, together with Figures 1 and 2, compare adaptation gains relative to the source-only baseline. The baseline results show that transfer quality depends strongly on both the domain-shift scenario and the signal representation. In the final benchmark, CWRU acts as the stronger stress test for negative transfer, whereas PU is more favorable on average but still not uniformly safe. Taken together, these results show that adaptation cannot be treated as inherently beneficial even within standard fault-diagnosis benchmarks, and that worst-case behavior is essential for interpreting reliability under domain shift.

4.1.1. CWRU

On CWRU, the FFT representation is the most reliable under adaptation. As shown in Table 3, FFT improves average macro-F1 from 0.758 to 0.842 with CORAL and to 0.836 with MMD, while also maintaining positive worst-case deltas of $+0.072$ and $+0.074$, respectively. This makes FFT the only representation on CWRU that is consistently beneficial across both average-case and worst-case views.

The raw and STFT models are less stable. For the raw model, CORAL reduces average macro-F1 from 0.807 to 0.792 and produces a worst-case delta of -0.040, while MMD is nearly neutral on average ($+0.006$) but still exhibits a harmful worst-case delta of -0.028. STFT is also vulnerable on CWRU despite being competitive in some individual settings. Its average macro-F1 drops slightly under both CORAL (0.756) and MMD (0.755), and its worst-case deltas are the most negative among the three representations (-0.069 and -0.065).

The scenario-level deltas in Fig. 1 clarify where these failures occur. Under `Cross-sensor`, FFT improves macro-F1 by $+0.087$ with CORAL and $+0.076$ with MMD, while the raw model remains near neutral and STFT is mixed. Under `Load 0→3`, FFT again remains positive with both methods ($+0.072$ and $+0.074$), but STFT becomes clearly harmful (-0.069 and -0.065). Under `Load 3→0`, FFT remains beneficial ($+0.094$ and $+0.086$), whereas the raw model becomes harmful for both CORAL and MMD (-0.040 and -0.028).

Overall, CWRU confirms that adaptation quality is highly representation-dependent and that some combinations are reliably harmful even when others are beneficial. This makes CWRU, especially under load transfer with raw or STFT inputs, a useful stress test for harmful adaptation rather than simply a benchmark for mean gains.

4.1.2. PU

PU shows a different pattern. Table 4 indicates that FFT and STFT are clearly beneficial on average, while raw adaptation is weaker but mostly non-catastrophic. Compared with CWRU, PU is therefore the more favorable benchmark overall, although it still contains practically relevant harmful cases.

For FFT on PU, average macro-F1 increases from 0.440 for source-only to 0.536 with CORAL and 0.535 with MMD. However, the worst-case behavior remains method-dependent: CORAL stays positive in the worst case ($+0.022$), whereas MMD still exhibits a small harmful case (-0.016). This is important for diagnostics interpretation because a favorable average result can still hide isolated but practically relevant failures.

STFT is the strongest overall representation on PU. Its average macro-F1 rises from 0.487 to 0.561 with CORAL and 0.577 with MMD, and both methods remain positive in the worst case ($+0.051$ and $+0.046$). In contrast to CWRU, STFT on PU is therefore both strong on average and stable in the worst case.

The scenario-level deltas in Fig. 2 explain this behavior. Under `Cross-speed`, FFT and STFT receive large gains, with FFT reaching $+0.169$ for CORAL and $+0.204$ for MMD, and STFT reaching $+0.097$ and $+0.136$.
Under `Artificial→Real`, most settings remain positive, but `FFT / MMD` becomes harmful with a delta of -0.016, which makes this scenario a useful negative-transfer case for the guardrail study.

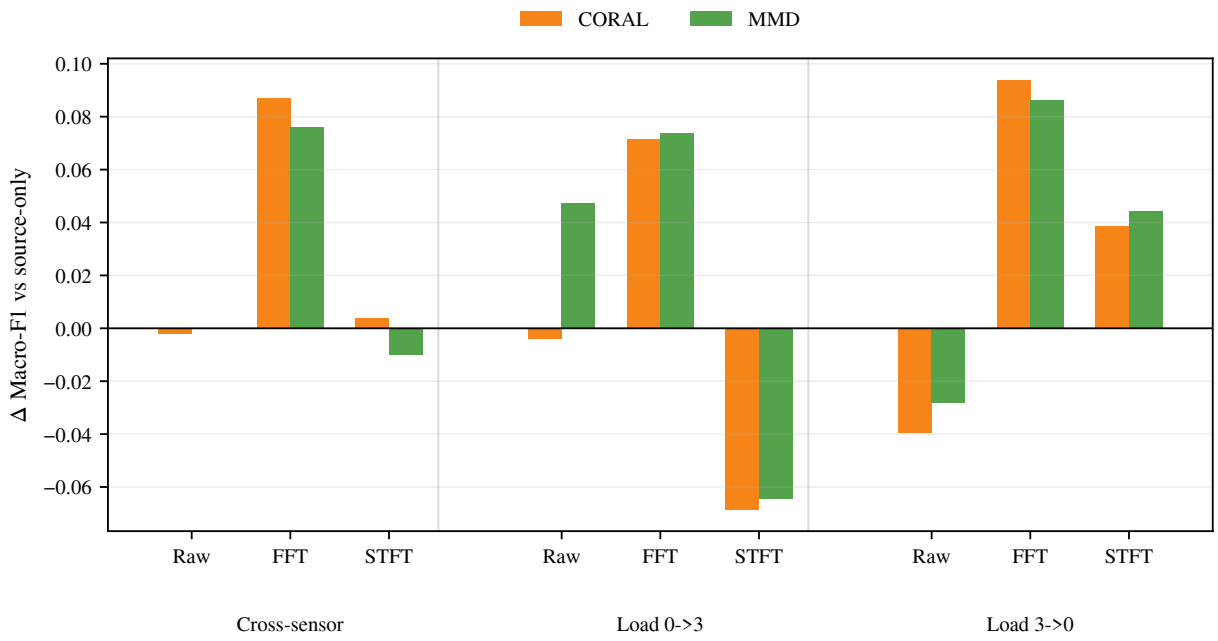

Figure 1. CWRU delta macro-F1 relative to source-only shows strong representation dependence and clear negative-transfer cases.

Table 3. CWRU average-case and worst-case baseline summary.

Model	Method	Avg Macro-F1	Worst Macro-F1	Avg Delta Macro-F1	Worst Delta Macro-F1	Avg Accuracy	Avg Balanced Acc.
Raw	source-only	0.807	0.559	+0.000	+0.000	0.874	0.813
Raw	CORAL	0.792	0.556	-0.015	-0.040	0.862	0.805
Raw	MMD	0.813	0.559	+0.006	-0.028	0.882	0.828
FFT	source-only	0.758	0.694	+0.000	+0.000	0.788	0.757
FFT	CORAL	0.842	0.788	+0.084	+0.072	0.861	0.844
FFT	MMD	0.836	0.781	+0.079	+0.074	0.858	0.841
STFT	source-only	0.765	0.738	+0.000	+0.000	0.775	0.779
STFT	CORAL	0.756	0.708	-0.009	-0.069	0.793	0.758
STFT	MMD	0.755	0.712	-0.010	-0.065	0.787	0.767

Overall, PU suggests that adaptation can be beneficial under more realistic transfer settings, but it also confirms that positive average transfer does not imply uniformly safe adaptation. Worst-case analysis therefore remains necessary even in the more favorable benchmark.

4.2. Source-Pretrained UDA

Source-pretrained UDA materially changes the reliability picture and must therefore be treated as a separate comparison regime rather than an implementation detail. Appendix Tables 7 and 8 report the absolute results for the guarded subset, and the final delta tables show that warm-start effects can be large.

On CWRU, the effect of source pretraining is mixed but substantial. For `Load 0→3 / STFT / MMD`, source-pretrained UDA reaches a macro-F1 of 0.780 and improves by $+0.067$ over plain UDA, while remaining only slightly above source-only ($+0.003$). For `Load 3→0 / FFT / CORAL`, source-pretrained UDA reaches 0.869 and improves by $+0.081$ over plain UDA and by $+0.175$ over source-only.

The raw CWRU case shows the opposite behavior. For `Load 3→0 / Raw / CORAL`, source-pretrained UDA reaches 0.827 but remains worse than plain UDA by -0.046 and worse than source-only by -0.085. This shows that warm start can materially affect adaptation outcomes, but it is not a universal remedy for negative transfer.

On PU, source-pretrained UDA is particularly important for `Artificial→Real / FFT / MMD`. In that case, the final results show a macro-F1 of 0.710, corresponding to a gain of $+0.070$ over plain UDA and $+0.054$ over source-only. This is markedly stronger than the plain UDA result for the same setting and changes the interpretation of any later rollback-based improvement.

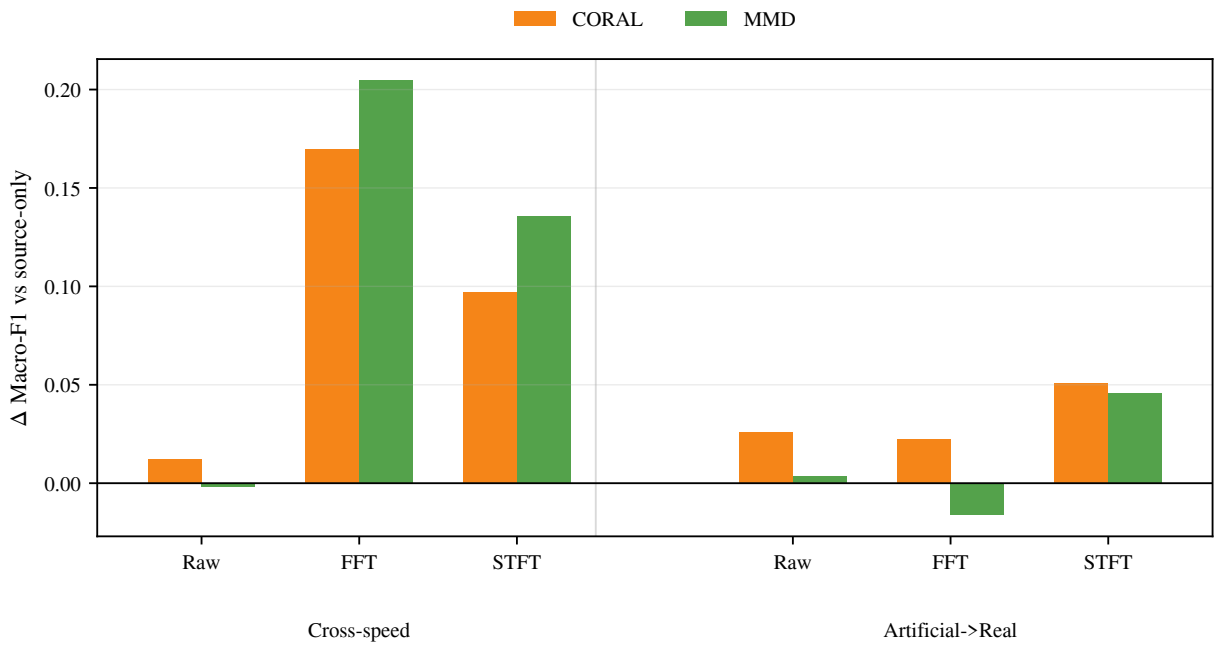

Figure 2. PU delta macro-F1 relative to source-only is more favorable overall, but still not uniformly safe.

Table 4. PU average-case and worst-case baseline summary from the final export.

Model	Method	Avg Macro-F1	Worst Macro-F1	Avg Delta Macro-F1	Worst Delta Macro-F1	Avg Accuracy	Avg Balanced Acc.
Raw	source-only	0.565	0.352	+0.000	+0.000	0.607	0.640
Raw	CORAL	0.584	0.365	+0.019	+0.012	0.622	0.655
Raw	MMD	0.565	0.350	+0.001	-0.002	0.605	0.639
FFT	source-only	0.440	0.225	+0.000	+0.000	0.474	0.517
FFT	CORAL	0.536	0.394	+0.096	+0.022	0.542	0.568
FFT	MMD	0.535	0.429	+0.094	-0.016	0.537	0.566
STFT	source-only	0.487	0.318	+0.000	+0.000	0.520	0.562
STFT	CORAL	0.561	0.415	+0.074	+0.051	0.577	0.611
STFT	MMD	0.577	0.454	+0.091	+0.046	0.595	0.621

For `Cross-speed / FFT / MMD`, source-pretrained UDA is slightly below plain UDA (-0.012) but remains strongly above source-only ($+0.193$). This indicates that the warm-start regime is already competitive in the beneficial PU setting and should be considered part of the reliability analysis rather than a minor optimization.

Overall, these results show that source pretraining is a meaningful reliability factor in its own right. Any apparent guardrail benefit must therefore be interpreted relative not only to plain UDA, but also to the already strong source-pretrained regime.

4.3. Guardrail Evaluation

The final guardrail results remain mixed across datasets, so the safeguard should be described as a prototype rather than a universally stable mechanism. Because the guardrail operates on top of an adapted training trajectory, its effect should be interpreted primarily as damage control on the adaptation process rather than as an independent classifier-improvement mechanism. Therefore, the guardrail effect should not be interpreted simply as the difference between guarded UDA and plain UDA. Because guarded runs start from a source-pretrained checkpoint, part of the observed difference may come from warm-started adaptation itself. For this reason, Appendix Tables 7 and 8 report source-pretrained UDA without rollback. The strongest current evidence comes from PU, where the final rollback policy restores the best eligible adapted checkpoint instead of reverting unconditionally to the source-pretrained model.

On PU, the final guardrail provides the most convincing positive result in the final artifact set. In Table 6, the safeguard remains inactive for `Cross-speed / FFT / MMD`, so the final result stays close to plain UDA, with a delta of only -0.014 macro-F1 relative to plain UDA, while still remaining strongly above source-only in the final comparison. This is an encouraging control case because it suggests that the

Table 5. Final CWRU guardrail effect relative to plain UDA on the guarded subset.

Scenario	Model	Method	Delta Macro-F1	Delta Accuracy	Delta Balanced Acc.	Triggered	Rollback	Trigger Epoch
Load $0 \rightarrow 3$	STFT	MMD	-0.178	-0.126	-0.155	yes	yes	11
Load $3 \rightarrow 0$	Raw	CORAL	-0.047	-0.027	-0.028	yes	yes	31
Load $3 \rightarrow 0$	FFT	CORAL	+0.085	+0.068	+0.090	no	no	–

Table 6. Final PU guardrail effect relative to plain UDA on the guarded subset.

Scenario	Model	Method	Delta Macro-F1	Delta Accuracy	Delta Balanced Acc.	Triggered	Rollback	Trigger Epoch
Cross-speed	FFT	MMD	-0.014	-0.014	-0.019	no	no	–
Artificial→Real	FFT	MMD	+0.020	+0.036	+0.028	yes	yes	40

guardrail does not automatically suppress a beneficial adapted state.

The guarded `Artificial→Real / FFT / MMD` run is the clearest positive safeguard case. It triggers at epoch 40, rolls back to the best eligible adapted checkpoint, and finishes $+0.020$ macro-F1 above plain UDA, together with supporting gains of $+0.036$ in accuracy and $+0.028$ in balanced accuracy. This indicates that the rollback logic can, at least in some cases, recover a better adapted solution than the final plain-UDA checkpoint.

The CWRU results are weaker. Table 5 shows that the final guardrail correctly leaves the helpful `Load 3→0 / FFT / CORAL` control untouched, and this case remains positive at $+0.085$ macro-F1 relative to plain UDA. However, the two harmful CWRU cases are not rescued by the final guardrail. For `Load 0→3 / STFT / MMD`, the final guarded result remains -0.178 macro-F1 below plain UDA, and for `Load 3→0 / Raw / CORAL`, it remains -0.047 below plain UDA.

These mixed outcomes are important for interpretation. The final prototype is no longer a purely destructive early-stop rule, because it can preserve useful adapted states on PU, but it still fails to deliver stable improvements across the guarded CWRU cases. Accordingly, the present results support the guardrail as prototype evidence for partial mitigation of harmful transfer, not as a generally validated safe-adaptation method. In addition, because several guarded runs are initialized from source-pretrained models, improvements over plain UDA should not automatically be interpreted as improvements attributable to the guardrail alone.

4.4. Overall Interpretation and Discussion

Three conclusions follow from the final results. First, negative transfer is a real and repeatable phenomenon in this benchmark, especially on CWRU under load transfer, and it depends strongly on representation rather than appearing uniformly across all settings. Second, source-pretrained UDA is a strong ablation that materially changes the interpretation of several guarded runs and should therefore be reported explicitly in reliability-focused studies. Third, the current guardrail prototype provides encouraging evidence on PU but is not yet robust enough to justify a broad cross-dataset claim of safer adaptation.

From a diagnostics perspective, these results show that average adaptation gains alone are not sufficient for deployment-oriented conclusions. Depending on the representation and transfer scenario, UDA can provide either meaningful improvement or harmful degradation, and on CWRU the degradation is large enough that a source-only reference remains necessary for reliable interpretation.

A second important observation is that source-pretrained UDA is not a minor implementation detail. In several guarded cases, warm-started adaptation materially changes the final outcome, which means that any apparent safeguard benefit must be interpreted relative not only to plain UDA but also to the pretrained adaptation regime. Without this comparison, rollback effects can easily be overstated.

The guardrail prototype is therefore best understood as a reliability-oriented mechanism for monitoring adaptation trajectories rather than as a validated safe-adaptation method. Its current behavior is encouraging on PU, where it can preserve a useful adapted checkpoint in selected cases, but the harmful CWRU cases show that the mechanism is not yet stable across datasets and representations. This may happen because target entropy can decrease even when the model becomes confidently biased toward incorrect target classes, or because alignment-loss reduction may improve marginal feature alignment without preserving class-conditional structure. Such effects are especially problematic for load/speed shifts, where fault-related spectral patterns change with operating condition. Moreover, if the entropy improvement required for arming is never reached, the guardrail remains inactive and harmful adaptation from the beginning may not be detected by this particular rule. Therefore, the CWRU failures indicate that future guardrails should include additional unlabeled risk signals, such as prediction diversity, class-balance drift, prototype consistency, or agreement between multiple representations.

The results also show that negative transfer is strongly representation-

dependent. Across the evaluated settings, the FFT-based representation was the most consistently reliable, especially on CWRU, where it improved both average and worst-case macro-F1. Raw time-domain models may preserve useful waveform information but can remain sensitive to phase, sensor placement, and operating-condition changes, while STFT representations introduce additional time–frequency structure that does not guarantee more stable adaptation and whose benefit appears to be dataset- and shift-dependent.

Overall, the main lesson is not that a universal safeguard has already been achieved, but that harmful adaptation is observable, non-trivial, and worth monitoring explicitly. The present results support the value of source-only baselines, worst-case reporting, and rollback-oriented safeguards as practical tools for studying trustworthy transfer in vibration-based fault diagnosis under domain shift.

5. CONCLUSION

This study showed that negative transfer is a real reliability issue in vibration-based unsupervised domain adaptation and that its severity depends strongly on both the transfer scenario and the signal representation. The results also showed that source-pretrained UDA is an important comparison regime, since part of the apparent safeguard benefit can otherwise be falsely attributed to rollback rather than to warm-started adaptation.

The proposed guardrail prototype produced encouraging results on selected PU scenarios, but remained unstable on the guarded CWRU subset. Accordingly, the present evidence should be interpreted as preliminary prototype evidence rather than as a demonstration of uniformly robust safe adaptation. Even though the magnitude of gains is modest, the practical relevance of the guardrail is not in producing large average improvements over plain UDA, but in limiting harmful adaptation and preserving safer checkpoints when the adaptation trajectory becomes unstable.

The main value of the study is therefore not a claim that safe adaptation has already been achieved, but a clearer reliability-oriented picture of when adaptation helps, when it harms, and why source-only baselines, worst-case reporting, and rollback-oriented safeguards are useful tools for studying trustworthy transfer in vibration-based fault diagnosis. Future work should focus on stricter run-level evaluation, leakage-safe threshold selection, broader cross-dataset validation, and more reliable unlabeled risk signals for rollback decisions.

Acknowledgment. This work was supported by the National Science Centre, Poland (NCN), under the PRELUDIUM 24 grant no. 2025/57/N/ST8/03313.

REFERENCES

Azari, M. S., Flammini, F., Santini, S., & Caporuscio, M. (2023). A systematic literature review on transfer learning for predictive maintenance in industry 4.0. *IEEE Access*, *11*, 12887-12910. doi: 10.1109/ACCESS.2023.3239784

Chen, X., Wang, S., Long, M., & Wang, J. (2019). Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 1081–1090).

Chen, X., Yang, R., Xue, Y., Huang, M., Ferrero, R., & Wang, Z. (2023). Deep transfer learning for bearing fault diagnosis: A systematic review since 2016. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–21. doi: 10.1109/TIM.2023.3244237

Chen, Z., He, G., Li, J., Liao, Y., Gryllias, K., & Li, W. (2021). Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11. doi: 10.1109/TIM.2020.2995441

Ganin, Y., & Lempitsky, V. (2015, 07–09 Jul). Unsupervised domain adaptation by backpropagation. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (Vol. 37, pp. 1180–1189). Lille, France: PMLR. Retrieved from https://proceedings.mlr.press/v37/ganin15.html

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006). A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems* (Vol. 19). MIT Press. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf

Hoffmann, M. A., & Lasch, R. (2025). Unlocking the potential of predictive maintenance for intelligent manufacturing: A case study on potentials, barriers, and critical success factors. *Schmalenbach Journal of Business Research*, *77*(1), 27–55. doi: 10.1007/s41471-024-00204-3

Lessmeier, C., Kimotho, J. K., Zimmer, D., & Sextro, W. (2016). Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *Phm society european conference* (Vol. 3). doi: 10.36001/phme.2016.v3i1.1577

Li, Y., Song, Y., Jia, L., Gao, S., Li, Q., & Qiu, M. (2021). Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble

learning. *IEEE Transactions on Industrial Informatics*, *17*(4), 2833–2841. doi: 10.1109/TII.2020.3008010

Ragab, M., Chen, Z., Wu, M., Li, H., Kwoh, C.-K., Yan, R., & Li, X. (2021). Adversarial multiple-target domain adaptation for fault classification. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11. doi: 10.1109/TIM.2020.3009341

Ragab, M., Eldele, E., Tan, W. L., Foo, C.-S., Chen, Z., Wu, M., ... Li, X. (2023). Adatime: A benchmarking suite for domain adaptation on time series data. *ACM Transactions on Knowledge Discovery from Data*, *17*(8), 106:1–106:18. doi: 10.1145/3587937

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing*, *64-65*, 100-131. doi: https://doi.org/10.1016/j.ymssp.2015.04.021

Sun, B., Feng, J., & Saenko, K. (2016). *Correlation alignment for unsupervised domain adaptation.*

Sun, B., & Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In G. Hua & H. Jégou (Eds.), *Computer vision – eccv 2016 workshops* (pp. 443–450). Cham: Springer International Publishing.

Wagner, T., & Sommer, S. (2021, 07). Feature based bearing fault detection with phase current sensor signals under different operating conditions. In (Vol. 6). doi: 10.36001/phme.2021.v6i1.2852

Wang, W., Li, H., Ding, Z., & Wang, Z. (2020). *Rethink maximum mean discrepancy for domain adaptation.* Retrieved from `https://arxiv.org/abs/2007.00689`

Wang, Z., Ragab, M., Yang, W., Wu, M., Jialin Pan, S., Zhang, J., & Chen, Z. (2024). Overcoming negative transfer by online selection: Distant domain adaptation for fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1-9. doi: 10.1109/TIM.2024.3472786

Wei, D., Han, T., Chu, F., & Zuo, M. J. (2021). Weighted domain adaptation networks for machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *158*, 107744. doi: 10.1016/j.ymssp.2021.107744

Zhang, W., Deng, L., Zhang, L., & Wu, D. (2023). A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, *10*(2), 305–329. doi: 10.1109/JAS.2022.106004

A. ADDITIONAL RESULTS

This appendix provides the regime-specific results used to interpret the main-text findings. In particular, it separates the effect of source-pretrained UDA from the effect of the rollback mechanism and summarizes the trigger behavior of the final guardrail runs.

The first two tables report source-pretrained UDA without rollback on the guarded subsets for CWRU and PU. These results are included to distinguish warm-start effects from the effect of the guardrail itself.

Table 7. Final CWRU source-pretrained UDA results on the guarded subset.

Scenario	Model	Method	Regime	Macro-F1	Accuracy	Balanced Acc.
Load $0 \rightarrow 3$	STFT	MMD	UDA+source-pretrain	**0.780**	0.793	0.773
Load $3 \rightarrow 0$	Raw	CORAL	UDA+source-pretrain	**0.827**	0.850	0.866
Load $3 \rightarrow 0$	FFT	CORAL	UDA+source-pretrain	**0.869**	0.871	0.886

Table 8. Final PU source-pretrained UDA results on the guarded subset.

Scenario	Model	Method	Regime	Macro-F1	Accuracy	Balanced Acc.
Cross-speed	FFT	MMD	UDA+source-pretrain	**0.417**	0.408	0.466
Artificial→Real	FFT	MMD	UDA+source-pretrain	**0.710**	0.729	0.715

The next two tables summarize the trigger behavior of the final guardrail runs. They complement the main-text safeguard results by showing trigger rate, rollback rate, and mean trigger epoch on the guarded subsets.

Table 9. Final CWRU guardrail trigger summary.

Method	N	Trigger Rate	Rollback Rate	Mean Trigger Epoch	Median Trigger Epoch
CORAL	2	0.500	0.500	31.000	31.000
MMD	1	1.000	1.000	11.000	11.000

Table 10. Final PU guardrail trigger summary.

Method	N	Trigger Rate	Rollback Rate	Mean Trigger Epoch	Median Trigger Epoch
MMD	2	0.500	0.500	40.000	40.000