Specification Document

1. Code Framework for DTL Paradigms

1) It is crucial to note that the backbone network (Backbone), serving as the feature extraction module, significantly impacts test accuracy. Different backbone networks exhibit varying feature representation capabilities, making it difficult to definitively determine which architecture is superior. Consequently, directly comparing the three fundamental DTL paradigms using results reported in disparate published papers would be both unfair and inappropriate. Therefore, to ensure a fair comparison, we employ an identical 1D-CNN-based backbone network to evaluate the performance of different DTL paradigms on the same dataset.

Fig. 1 illustrates the architecture of this backbone network. It comprises four one-dimensional convolutional layers. Each convolutional layer is equipped with a 1D batch normalization (BN) layer and a ReLU activation function. Additionally, the second convolutional layer incorporates a 1D max-pooling layer, while the fourth convolutional layer utilizes a 1D adaptive max-pooling layer. The output vector from the convolutional layers is subsequently flattened and passed through a fully connected (Fc) layer, a ReLU activation function, and a dropout layer before being fed into the classifier. Detailed parameters are provided in Table 1.

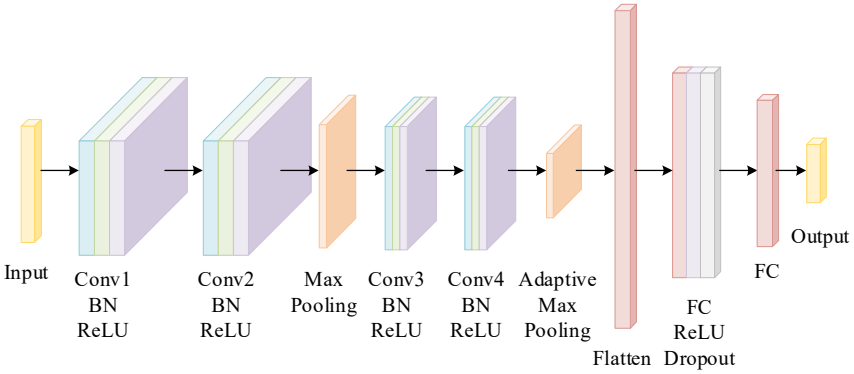


Fig 1. Structure of the basic model.

Tab 1. Parameters of the basic model.

	Parameter	Activation	Output shape
Conv1+BN	(1,16,15)	ReLU	(64,16,1010)
Conv2+BN	(16,32,3)	ReLU	(64,32,1008)
MaxPool1d	(2,2)	-	(64,32,504)
Conv3+BN	(32,64,3)	ReLU	(64,64,502)
Conv4+BN	(64,128,3)	ReLU	(64,128,500)
AdaptiveMaxPool1d	4	-	(64,128,4)
Flatten	-	-	(64,512)
Linear	(512,256)	ReLU	(64,256)
Dropout	p=0.5	-	(64,256)
Linear	(256,10)	SoftMax	(64,10)

Guided by the principle of "how the model reduces inter-domain generalization error," the algorithmic workflows for each of the three fundamental DTL paradigms were designed accordingly.

a) Pre-training－Finetuning (PTFT)

First, backbone network_1 undergoes supervised learning on the source domain training set for 200 epochs. We employ the minibatch AdamW optimizer with a batch size of 64. The "step" learning

rate annealing strategy in PyTorch is adopted, with an initial learning rate of 1e-3 decaying (multiplied by 0.1) at epochs 100 and 150, respectively. Subsequently, the network weights and bias parameters updated at the 200th epoch are saved and transferred to backbone network_2 to construct the target domain pre-trained network. Next, using the predefined "set_freeze_by_id" function, we unfreeze the parameter update mechanism for specified layers in the backbone network (excluding the classifier) (as illustrated in Fig. 2). The target domain labeled training set is then used to perform supervised training on backbone network_2 for 100 epochs.

Considering the distinct requirements of the pre-training versus fine-tuning stages—where pre-training focuses on learning generic features while fine-tuning adapts the model to specific tasks—we configure different optimizer and learning rate parameters for fine-tuning. We utilize the minibatch SGD optimizer with a batch size of 64. The "step" learning rate annealing strategy is applied, with an initial learning rate of 1e-4 decaying (multiplied by 0.1) at epochs 30 and 60, respectively. This configuration is chosen because AdamW, as an adaptive optimizer combining momentum and adaptive learning rates (RMSprop), offers rapid convergence and low sensitivity to hyperparameters, making it suitable for efficient initialization. In contrast, SGD with momentum enables "escaping" local minima near optimal solutions, yielding superior generalization performance—particularly advantageous for pre-trained models with initially converged parameters.

========== Freeze Status ==========	
Layer: layer1.0.weight, Requires Grad: False	Layer: layer1.0.bias, Requires Grad: False
Layer: layer1.1.weight, Requires Grad: False	Layer: layer1.1.bias, Requires Grad: False
Layer: layer2.0.weight, Requires Grad: False	Layer: layer2.0.bias, Requires Grad: False
Layer: layer2.1.weight, Requires Grad: False	Layer: layer2.1.bias, Requires Grad: False
Layer: layer3.0.weight, Requires Grad: False	Layer: layer3.0.bias, Requires Grad: False
Layer: layer3.1.weight, Requires Grad: False	Layer: layer3.1.bias, Requires Grad: False
Layer: layer4.0.weight, Requires Grad: False	Layer: layer4.0.bias, Requires Grad: False
Layer: layer4.1.weight, Requires Grad: False	Layer: layer4.1.bias, Requires Grad: False
Layer: layer5.0.weight, Requires Grad: True	Layer: layer5.0.bias, Requires Grad: True
==========	

Fig 2. The requires_grad state of parameters at each layer of the basic network (the fine-tuning layer is the last layer).

b) Statistical Moment Matching (SMM)

For this fundamental DTL paradigm centered on "feature adaptation," we employ a phased training strategy to enhance the stability and effectiveness of transfer learning.

Phase 1: Initially, the backbone network undergoes supervised learning only on the source domain training set. This stage focuses on learning the feature representations and classifier parameters specific to the source domain.

Phase 2: Subsequently, unlabeled training data from the target domain is introduced. Transfer learning methods are then applied to align the feature distributions between the source and target domains, facilitating knowledge transfer.

Rationale: This approach ensures the model first masters source domain knowledge, establishing foundational feature extraction and classification capabilities. It thus prevents interference from the unknown target domain distribution during later adaptation stages caused by unstable model parameters. Leveraging the knowledge acquired from the source domain, the model is then adapted using target domain data to adjust to its distribution. Crucially, this phased strategy mitigates the risk of "negative transfer" (where source domain knowledge is corrupted by target domain noise), which can occur with direct mixed training.

Implementation: Within the SMM code framework, the total training is set to 300 epochs. The first 50 epochs constitute Phase 1, while epochs beyond 50 define Phase 2. We employ the minibatch AdamW optimizer with a batch size of 64. The PyTorch "step" learning rate annealing strategy is used, starting at an initial rate of 1e-3 and decaying (multiplied by 0.1) at epochs 100 and 150, respectively.

Distribution Metric Selection: Various metrics can quantify distribution differences in SMM, including Maximum Mean Discrepancy (MMD), CORrelation Alignment (CORAL), and Kullback-Leibler Divergence (KLD). Among these, Multi-Kernel MMD (MK-MMD) offers distinct advantages: by combining multiple kernel functions, it flexibly captures nonlinear distribution discrepancies between domains, exhibits stronger adaptability to complex data, and has demonstrated high robustness and effectiveness in cross-domain tasks in prior research.

Framework Design: Consequently, within our code framework, **we utilize MK-MMD as a representative case study to design the SMM algorithm flow**. MK-MMD is integrated into the loss function to achieve feature alignment. The final loss function is defined as follows,

$$\mathcal{L} = \begin{cases} \mathcal{L}_c & epoch \leqslant 50 \\ \mathcal{L}_c + \lambda_{\mathrm{MK-MMD}} \mathcal{L}_{\mathrm{MK-MMD}}\left(\mathcal{D}_s, \mathcal{D}_t\right) & epoch > 50 \end{cases} \tag{1}$$

where $\lambda_{\text{MK-MMD}}$ is a tradeoff parameter. We implement a sigmoid-based gradual increase for $\lambda_{\text{MK-MMD}}$, smoothly scaling it from 0 to 1 over the course of training. This design dynamically adjusts the influence of the domain adaptation loss. Target domain information is introduced gradually after the model has matured in Phase 1. The domain adaptation loss progressively dominates in the later stages of training. This results in a more stable adaptation process and aligns precisely with the overall logic of the phased training strategy: "Learn the task first, then adapt to the domain."

$$\lambda_{\mathrm{MK-MMD}} = \frac{1 - e^{-10 \cdot \left(\frac{epoch-50}{300-50}\right)}}{1 + e^{-10 \cdot \left(\frac{epoch-50}{300-50}\right)}} \tag{2}$$

c) Domain Adversarial Training (DAT)

Since DAT shares the core concept of "feature adaptation" with SMM, the training strategy, number of epochs (300), optimizer (AdamW), and learning rate parameters in the code framework adopt identical configurations to SMM.

Method Selection Rationale: Among domain-adversarial deep transfer learning methods, prominent approaches include Domain Adversarial Neural Network (DANN), Adversarial Discriminative Domain Adaptation (ADDA), and Conditional Domain Adversarial Network (CDAN). DANN, as one of the earliest methods introducing adversarial learning into transfer learning, established the foundation for subsequent research. Its implementation relies on a single Gradient Reversal Layer (GRL) and domain discriminator to construct the adversarial mechanism—eliminating the need for complex generators or multi-branch structures. This design offers low computational cost and facilitates integration within diverse neural network frameworks. Therefore, **we employ DANN as the representative case study to construct our code framework**, enhancing the comprehensibility of adversarial-based transfer learning workflows while providing a foundation for future improvements.

Framework Implementation: Within the code framework, we construct DANN by augmenting the backbone network with a Gradient Reversal Layer (GRL) and a domain discriminator, as detailed in Fig. 3. The GRL reverses the sign of gradients received from upstream layers during backpropagation. This forces the optimization direction of the domain classifier to oppose that of the feature extractor, establishing adversarial training. Consequently, the feature extractor learns

features that "confound the domain classifier," eliminating the need for explicit design of complex adversarial optimization procedures. The domain discriminator is implemented as a three-layer fully connected (Fc) binary classifier. The output features of these Fc layers are 1024 (FC1), 1024 (FC2), and 2 (FC3), respectively. The parameter of a dropout layer is p = 0.5.

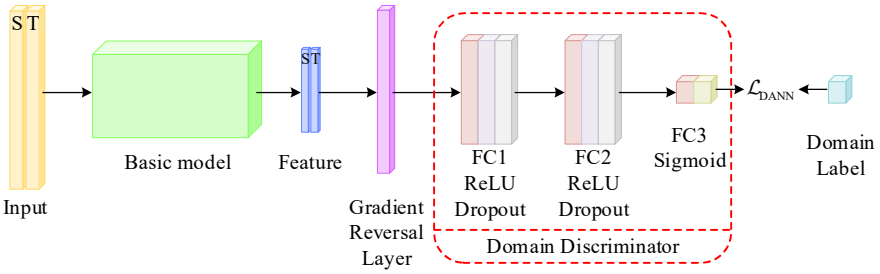


Fig 3. Structure of the DANN.

Adversarial Loss Integration: The adversarial concept is inherently integrated into the loss function to achieve feature alignment. The final loss function is defined as follows,

$$\mathcal{L}=\begin{cases}\mathcal{L}_c & epoch\leqslant 50\\ \mathcal{L}_c-\lambda_{\mathrm{DANN}}\mathcal{L}_{\mathrm{DANN}}\left(\mathcal{D}_s,\mathcal{D}_t\right) & epoch>50\end{cases} \tag{3}$$

where λ_{DANN} is a tradeoff parameter.

Tradeoff Parameter Distinction: While both λ_{DANN} and $\lambda_{\mathrm{MK\text{-}MMD}}$ employ progressive weighting mechanisms, their implementation details differ significantly.

$\lambda_{\mathrm{MK\text{-}MMD}}$ dynamically scales the distribution discrepancy loss weight, updated per epoch. All batches within an epoch share the same weight, constituting a coarse-grained adjustment.

λ_{DANN} dynamically adjusts the gradient reversal coefficient (GRL intensity), acting directly on gradient computation. Updated per iteration (batch), it progressively intensifies adversarial strength batch-wise. This fine-grained adaptation responds effectively to potential distribution fluctuations between mini-batches during training, aligning with the training dynamics. Critically, DANN's adversarial loss weight fundamentally controls gradients via the GRL coefficient rather than loss function scaling.

$$\lambda_{\mathrm{DANN}}=\frac{1-e^{-10\cdot\left(\frac{\text{iter\_num}}{(300\text{-}50)\times len(\text{dataloader})}\right)}}{1+e^{-10\cdot\left(\frac{\text{iter\_num}}{(300\text{-}50)\times len(\text{dataloader})}\right)}} \tag{4}$$

2. Regarding Data Selection Issues

Through comprehensive investigation of currently available open-source datasets for health state anomalies in electric powertrain systems, we find that existing public data cannot fully replicate the complex operating conditions encountered in actual electric vehicle operation. Based on this, our study evaluates the usability of open-source data according to fundamental principles for transfer learning validation. The core value of transfer learning lies in addressing "model failure caused by domain discrepancy," whose verification requires satisfying two criteria:

a) Controllability of domain shift: Cross-domain data should effectively characterize distribution differences between source and target domains.

b) Universality of fault mechanisms: The involved fault modes must reflect the degradation essence of real-world systems.

According to these criteria, significant data barriers and validation feasibility disparities exist in electric powertrain systems health maintenance.

Motor systems—Data: As shown in Fig. 4, motor systems contain fault-prone subcomponents (stator, rotor, bearings). Their data availability analysis is as follows:

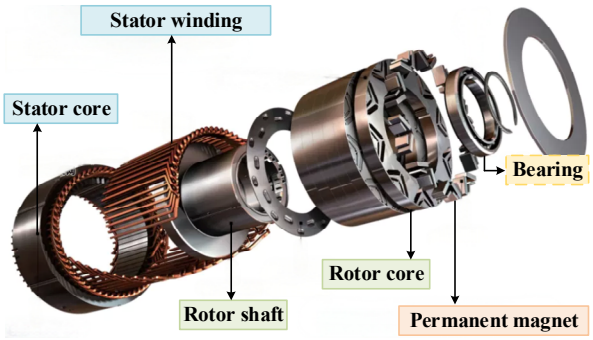


Fig 4. Components of the motor system (Taking PMSM as an example).

➢ **Stator Faults**

The stator winding short-circuit fault dataset released by the Korea Advanced Institute of Science and Technology (KAIST) in 2023 records vibration and current signals from three power-rated PMSMs (1.0 kW, 1.5 kW, 3.0 kW) under normal, inter-coil short-circuit, and inter-turn short-circuit conditions. While attempting to simulate multi-level fault severity, **this dataset exhibits three critical limitations**:

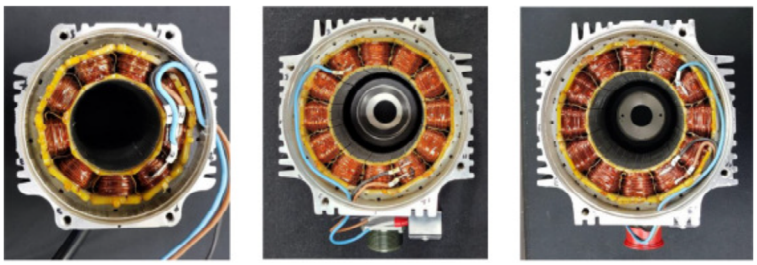
i) Fault simulation was achieved by connecting shunt resistors of varying resistances in parallel between adjacent turns in Phase A to represent different severity levels. Fault severity was defined as the ratio of current in the bypass circuit to that in the normal circuit. This approach contradicts the academic consensus that the severity of an inter-turn short circuit primarily depends on the short-circuit current amplitude under specific operating speeds, which can be approximated as a combination of the shorted turn percentage and the short-circuit contact resistance.

$$FI = \begin{cases} \dfrac{R_{it}}{R_{it} + R_{\text{bypass}}} \\ \dfrac{\mu}{\mu R_a + R_{\text{bypass}} - \mu^2 R_a} \end{cases} \tag{5}$$

ii) All data were collected under a single operating condition (3000 rpm, 1.5 N·m), preventing the construction of cross-condition transfer tasks.

iii) Shunt resistors of identical resistance values represent inequivalent physical severity levels across motors of different power ratings. Consequently, cross-domain transfer settings between motor systems of varying power ratings under the same stator winding fault severity level are inherently infeasible.

Therefore, this open-source dataset fundamentally fails to satisfy the "Controllability of domain shift."



(a) (b) (c)

Fig 5. Artificial fault seeded on A-phase of PMSMs with different powers: (a) 1.0 kW, (b) 1.5 kW, and (c) 3.0 kW.

- **Rotor Faults:**

The 2018 dataset from Wuhan University on four rotor states—normal, unbalanced, misaligned, and rubbing—records vibration signals from a 148W DC motor operating at 1200 rpm. Constrained by **a single constant operating condition**, it lacks data distribution shifts necessary to construct the "source-target domain" structure required for transfer learning. This inherently violates the core problem transfer learning aims to solve and fails to meet the rigid constraint of "Controllability of domain shift".

- **Bearing Faults:**

As one of the most prevalent faults in motor systems, bearing faults account for 40% or more of all failures. Furthermore, bearing fault diagnosis is a classic problem in motor (rotating machinery) system health monitoring, with research history traceable to the establishment of the Mechanical Failure Prevention Group (MFPG) in 1967. Decades of development have yielded multiple authoritative open-source datasets, such as: Case Western Reserve University (CWRU) Dataset, Paderborn University (PU) Dataset, JiangNan University (JNU) Dataset. **Their core advantages lie in**:

i) Most test conditions in these datasets involve varying operating conditions (speed and load), enabling straightforward construction of the "source-target domain" structure required for transfer learning. This satisfies the constraint of "Controllability of domain shift".

ii) The operational mechanisms of operating conditions on vibration signals under fault states are well-defined: speed variations manifest as differences in fault component rotational frequencies and vibration amplitudes, while load changes manifest as vibration amplitude differences induced by fault impact energy. This aligns with the requirement for "Universality of fault mechanisms".

In summary, under current data constraints, bearing fault datasets emerge as the optimal open-source resource with verifiable feasibility for motor systems, owing to their adjustable operating conditions and clear physical mechanisms. See Table 2 for specifics.

Tab 2. Comparison of validation feasibility for open-source data in motor system.

	Dataset Source	Satisfies DTL Validation Criteria
Stator	Vibration and current dataset of stator faults	×
Rotor	Vibration dataset of rotor faults	×
Bearing	CWRU Dataset、PU Dataset、JNU Dataset	√

To scientifically validate the cross-domain performance of DTL paradigms, **we ultimately selected the CWRU Dataset as the benchmark**, based on the following rationale:

i) Widely Recognized Validation Benchmark: This dataset has been extensively utilized in top-tier journals (e.g., IEEE Transactions) for research on motor system health management. It incorporates explicit variations in operating conditions, supporting the construction of 12 standardized transfer tasks. This perfectly matches the "cross-operating-condition migration" scenario.

ii) Engineering Universality of Physical Mechanisms: The impact of rotational speed and load variations on vibration signals under fault states adheres to explicit physical laws. Although CWRU data is not specifically from electric vehicles, its underlying fault physical mechanisms are fundamentally consistent with in-vehicle scenarios. Furthermore, its simulation of fault damage progression (e.g., fault diameter gradations), while originating from industrial settings, aligns with

the degradation mechanisms of bearings in vehicle-mounted motors.

iii) Universality Verification of Paradigms: Quantitative accuracy comparison across multiple transfer tasks clearly reveals the performance differences among the three DTL paradigms.

Therefore, we contend that this approach, within the constraints of available data, maximizes the rigor of the experimental comparison of fundamental DTL paradigm performance.

3. Quantitative Results of Supplementary Experiments

Following the above analysis, we ultimately adopted the CWRU dataset provided by the Case Western Reserve University (CWRU) Bearing Data Center. Consistent with other relevant studies, we utilized drive-end bearing fault data sampled at 12 kHz. As shown in Table 3, based on fault diameters, our dataset is divided into ten categories, including one normal bearing state (NA) and nine fault states. The fault states encompass three fault types: Inner Race Faults (IF), Outer Race Faults (OF), and Balling element Faults (BF). Different fault diameters equivalently simulate the deterioration trend of fault damage. Furthermore, as shown in Table 4, the test conditions of the CWRU dataset include four distinct operating conditions (corresponding to different combinations of rotational speed and load). In the cross-domain maintenance task setting, these conditions can be regarded as distinct domains. Consequently, we can construct the "source domain-target domain" structure required for deep transfer learning between any two different operating conditions, forming a total of twelve cross-domain tasks. Among these, Task1(0→1) denotes migration from the source domain with a rotational speed of 1797 rpm and a load of 0 HP to the target domain with a rotational speed of 1772 rpm and a load of 1 HP.

Tab 3. Description of class labels of CWRU.

Class Label	0	1	2	3	4	5	6	7	8	9
Fault Location	NA	IF	BF	OF	IF	BF	OF	IF	BF	OF
Fault Size (mils)	0	7	7	7	14	14	14	21	21	21

Tab 4. Operating conditions and cross-domain tasks of CWRU.

Operating conditions	0			1			2			3		
Speed (rpm)	1797			1772			1750			1730		
Load (HP)	0			1			2			3		
Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Task 11	Task 12
0→1	0→2	0→3	1→0	1→2	1→3	2→0	2→1	2→3	3→0	3→1	3→2

We directly utilized the raw time-domain signals as network input. Considering that the minimum rotational frequency among the four operating conditions is 28.83 Hz (corresponding to approximately 417 samples per shaft revolution), to ensure each sample contains information from at least two rotor rotation cycles under different conditions while guaranteeing a sufficient number of training samples, we set the sliding window length to 1024 data points and the sliding step size to 256 for sample extraction. Subsequently, to prevent interference from class imbalance on the performance comparison of different DTL fundamental paradigms, we randomly selected 256 samples from all available samples within each category, ensuring a balanced number of samples per sub-category. Next, we preprocessed the time-domain input samples using Z-score normalization and allocated 70% of the total samples to the training set and 30% to the test set to prevent test data leakage. The aforementioned data processing pipeline was applied to both the source domain and target domain data.

To systematically evaluate the generalization capability of different DTL fundamental paradigms in cross-domain maintenance tasks and ensure the stability, robustness, and fairness of experimental

results and model comparisons, we implemented the following triple safeguards to eliminate interference from random factors. Firstly, to ensure reproducibility, we fixed all potential random sources (including PyTorch, NumPy, Python's native random module, and the CUDA environment) within the code framework via a custom "setup_seed" function. Each experiment used a unique and fixed random seed (seed=43, 44, 45, 46, 47), guaranteeing that all randomized procedures—such as data partitioning (e.g., train/test set splits), model parameter initialization, and data augmentation methods—were completely identical. Secondly, each model was trained for 300 epochs, and the accuracy for each epoch was recorded to observe long-term convergence. Given that diagnostic accuracy may fluctuate significantly during early training due to optimizer oscillations or overfitting/underfitting, results from later stages (near convergence) better reflect the model's true performance. Therefore, we selected the average accuracy over the last 10 epochs as the performance metric for a single experiment, thereby obtaining a "stable performance estimate" of the model under that specific random seed. Finally, we repeated the experiment independently five times (each time using a different fixed seed) and performed a secondary averaging of the stable performance estimates from each single experiment to obtain the final average accuracy. This method effectively assessed the algorithm's overall performance under different initialization conditions, reduced interference from randomness, and ensured the generality and reliability of the results. It is worth noting that all experiments were executed under Windows 10 and PyTorch 2.3.1 running on a computer with an Intel Core i7-12700KF CPU, an NVIDIA GeForce RTX 4060Ti GPU, and 16 GB of RAM.

1) Domain Shift Vulnerability of Baseline Models

As illustrated in Fig. 6, the baseline model (i.e., the model without deep transfer learning) achieved diagnostic accuracies exceeding 99% on the source domain test sets across all cross-domain maintenance tasks, demonstrating its excellent within-domain diagnostic capability. However, when directly applying the source-domain pre-trained model to target-domain test sets, its performance degraded to varying degrees in all 12 cross-domain tasks, with the degradation severity positively correlating with domain discrepancy. In mild-shift tasks (e.g., T0→T1: 1 HP load difference between domains), accuracy decreased by approximately 3% (99.94% → 97.03%). In severe-shift tasks (e.g., T0→T3: 3 HP load difference), accuracy plummeted to 76.41% (T10) or even 73.13% (T11), representing a maximum decline of 27%. Furthermore, Fig. 7 displays box plots of diagnostic accuracy from multiple tests on target-domain test sets. These reveal drastic performance fluctuations in the baseline model, with Min-Max differences reaching 18.47% (73.44%-91.91%) in certain tasks (e.g., T6), indicating extreme sensitivity to target-domain data distribution. **Collectively, these results confirm that domain distribution differences severely degrade the generalization capability of baseline models.**

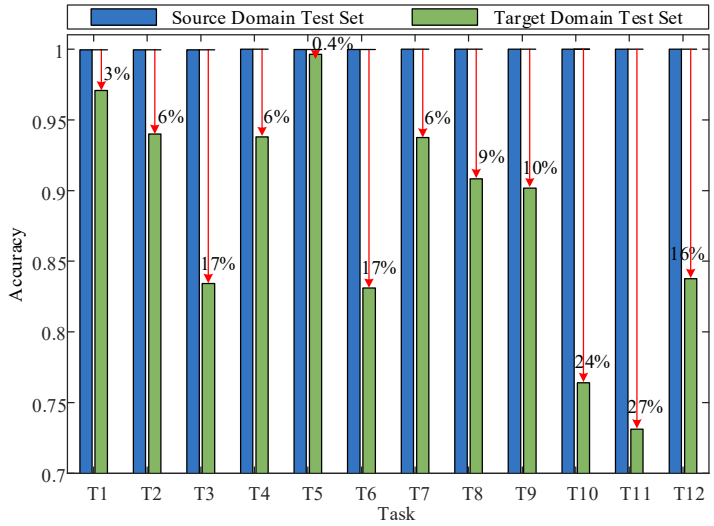


Fig 6. Mean diagnostic accuracy of the baseline model on source domain and target domain test sets in different cross-domain tasks.

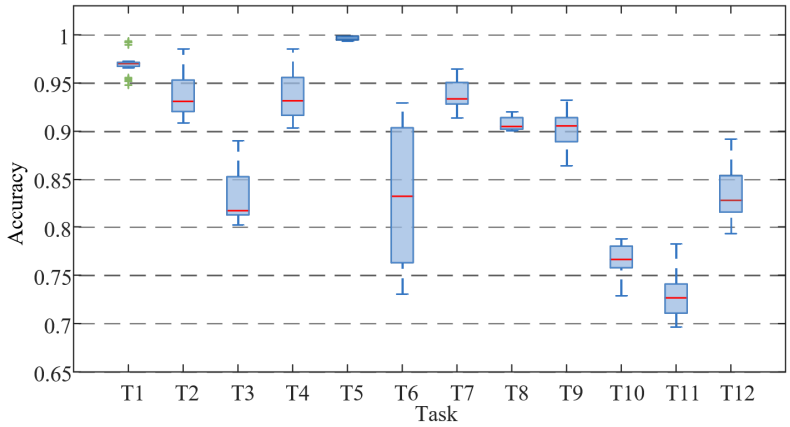


Fig 7. Box plot of diagnostic accuracy of the baseline model on target domain test sets of 12 cross-domain tasks.

2) Necessity Verification of DTL Paradigms

Table 5 summarizes the average, maximum, and minimum diagnostic accuracies of the baseline model versus three fundamental DTL paradigms (PTFT, SMM, DAT) across all cross-domain tasks over five randomized tests. The data demonstrate that all three DTL paradigms significantly outperform the baseline model in every cross-domain task. For severe-shift tasks (e.g., T11), DTL methods (PTFT:98.77%, SMM:97.06%, DAT:98.49%) improved average accuracy by over 23% compared to the baseline (73.13%). Even in mild-shift tasks (e.g., T1), DTL increased average accuracy from 97.07% to over 99.56%. A 3D confusion matrix visualizes classification results for the 10 categories in cross-domain task T10 (Fig. 8). The baseline model shows extremely poor accuracy for Class_label_3, Class_label_5, and Class_label_9: 41.56% of Class_label_3 samples were misclassified as Class_label_7; 93.51% of Class_label_5 samples were misclassified as Class_label_7; all Class_label_9 samples were misdiagnosed. In contrast, fundamental DTL paradigms (whether via parameter transfer or feature adaptation) substantially improved accuracy for these severely misclassified categories. Additionally, the Min-Max differences for DTL paradigms were generally $\leqslant$2% (e.g., PTFT in T12: 100%-100%), far

below the baseline's maximum difference (18.47%). **These results conclusively demonstrate that DTL effectively mitigates distribution shift impacts through domain adaptation mechanisms, validating its necessity for cross-domain diagnostics**.

Tab 5. Diagnostic accuracy of three basic paradigms of DTL in different cross-domain tasks.

Method	T1-Accuracy(%)			T2-Accuracy(%)			T3-Accuracy(%)			T4-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	97.07	99.21	95.23	93.99	98.40	91.16	83.41	88.66	80.29	93.78	97.79	90.63
PTFT	**99.92**	100	99.87	99.87	100	99.79	**99.97**	100	99.87	**99.22**	99.74	98.76
SMM	99.56	99.74	99.19	99.79	100	99.41	99.86	100	99.71	98.99	99.51	98.66
DAT	99.84	100	99.65	**99.91**	100	99.75	99.74	99.97	99.48	98.94	99.61	98.61
Method	T5-Accuracy(%)			T6-Accuracy(%)			T7-Accuracy(%)			T8-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	99.63	99.87	99.44	83.10	91.91	73.44	93.74	95.49	91.97	90.83	91.85	90.23
PTFT	99.95	100	99.87	**99.99**	100	99.96	97.92	99.23	96.29	**99.09**	99.30	98.80
SMM	99.95	100	99.87	99.76	99.90	99.58	97.85	98.92	96.55	98.95	99.47	98.22
DAT	**99.97**	100	99.87	99.67	99.87	99.27	**97.94**	98.66	97.12	98.48	98.72	98.19
Method	T9-Accuracy(%)			T10-Accuracy(%)			T11-Accuracy(%)			T12-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	90.17	92.68	87.98	76.41	78.19	73.39	73.13	78.15	69.78	83.77	88.75	80.04
PTFT	**100**	100	100	96.24	97.04	95.47	**98.77**	99.23	97.89	**100**	100	100
SMM	99.98	100	99.9	97.15	98.48	95.57	97.06	98.54	96.12	99.93	100	99.82
DAT	99.95	100	99.87	**97.91**	98.19	97.27	98.49	98.83	98.26	99.78	100	99.38

Remarks: T1-Accuracy (%) represents the accuracy in the Task 1. Mean is the mean accuracy of the 5 repeated experiments; Max is the maximum accuracy and Min is the minimum accuracy.

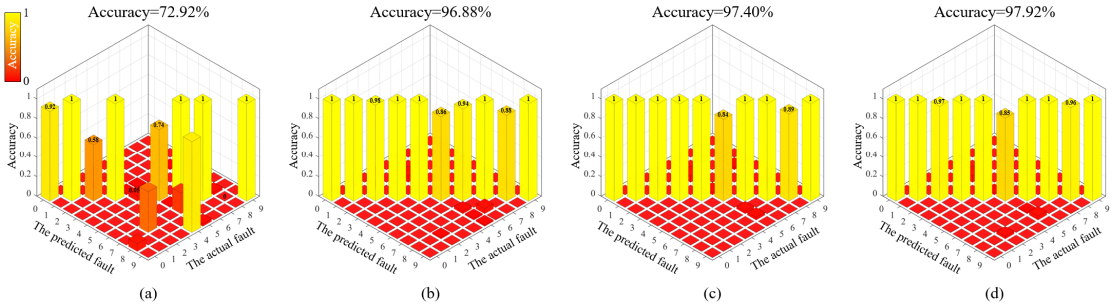


Fig 8. Three-dimensional confusion matrix in the T10. (a)Base; (b) PTFT; (c) SMM; (d) DAT.

3) Necessity Verification of DTL Paradigms

Fig. 9 presents the average diagnostic accuracies of the baseline model and three fundamental DTL paradigms (PTFT, SMM, DAT) across 12 cross-domain tasks over five randomized tests. **The three DTL paradigms exhibit varying performance across tasks, with no universally optimal paradigm**. This divergence may stem from task-specific characteristics, intrinsic data relationships, and directionality of domain shifts, causing optimal paradigm selection to vary with transfer direction. Thus, **the effectiveness of fundamental DTL paradigms is highly dependent on specific cross-domain scenarios** (e.g., shift type, fault-operating condition coupling), **necessitating paradigm selection aligned with task characteristics**.

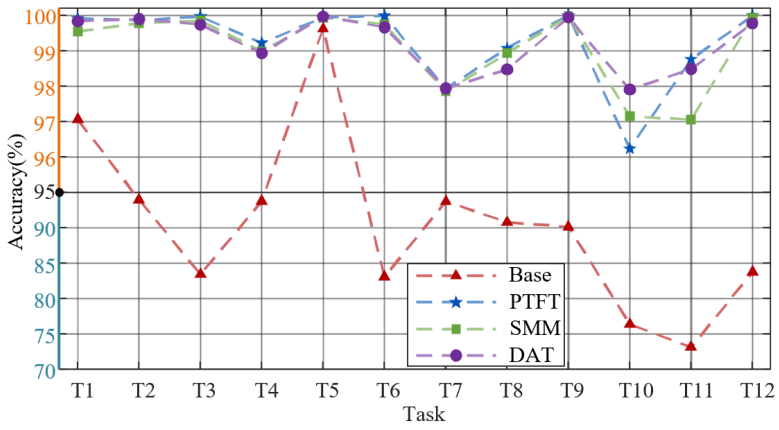


Fig 9. Mean diagnostic accuracy of three basic paradigms of DTL in different cross-domain tasks.

Experimental results demonstrate:

✓ The critical necessity of DTL over baseline models;

✓ Paradigm selection must account for intrinsic task properties.

Critically, **this analysis primarily validates the necessity and universal effectiveness of fundamental DTL paradigms in cross-domain tasks—not to identify a single optimal solution for all tasks**. This conclusion aligns with engineering realities: cross-domain scenarios in electric vehicle electric powertrain systems (battery-motor-power electronics) exhibit high heterogeneity, requiring flexible adaptation of DTL paradigms. Furthermore, though validated on CWRU bearing data, the deep transfer learning mechanisms (addressing data distribution shifts/feature space discrepancies) share identical cross-domain essence with EV system maintenance. **The core validation lies in algorithmic robustness against distribution shifts—not replication of specific scenarios—and the implemented code framework is readily extensible to other systems**. Thus, this study provides both methodological support and performance benchmarks for deploying deep transfer learning in electric powertrain systems maintenance of electric vehicles.

操作文档

1. DTL 范式的代码框架

1）考虑到，主干网络（Backbone）作为特征提取模块，对测试精度具有显著影响。不同主干网络的特征表示能力存在差异，难以判定何种主干网络更优，而直接采用已发表论文中所列结果对三种 DTL 基础范式进行对比既不公平也不恰当。因此，我们采用相同的基于 1D-CNN 的主干网络来验证不同 DTL 基础范式在同一数据集上的性能，以确保公平。图 1 展示了主干网络的架构，其中包含四个一维卷积层，每层均配备一维批量归一化（batch normalization, BN）层和 ReLU 激活函数。此外，第二个卷积层配备一维最大池化层（Max-pooling layer），第四个卷积层配备一维自适应最大池化层（adaptive max-pooling layer）。卷积层的输出向量随后将被展平，并经过全连接（Fc）层、ReLU 激活函数和 dropout 层，送入分类器。详细参数见表 1。

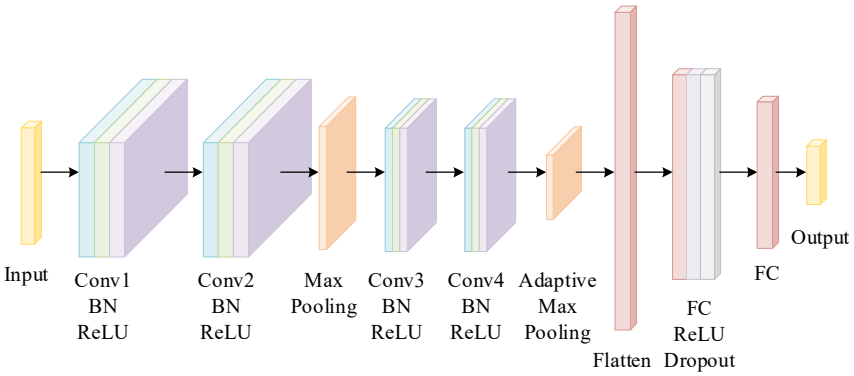


Fig 1. Structure of the basic model.

Tab 1. Parameters of the basic model.

	Parameter	Activation	Output shape
Conv1+BN	(1,16,15)	ReLU	(64,16,1010)
Conv2+BN	(16,32,3)	ReLU	(64,32,1008)
MaxPool1d	(2,2)	-	(64,32,504)
Conv3+BN	(32,64,3)	ReLU	(64,64,502)
Conv4+BN	(64,128,3)	ReLU	(64,128,500)
AdaptiveMaxPool1d	4	-	(64,128,4)
Flatten	-	-	(64,512)
Linear	(512,256)	ReLU	(64,256)
Dropout	p=0.5	-	(64,256)
Linear	(256,10)	SoftMax	(64,10)

按照“模型降低域间泛化误差”的作用机制，分别对三类 DTL 基础范式的算法流程进行设计。

a）Pre-training - Finetuning (PTFT)

首先，主干网络_1 在源域训练集上进行监督学习，epoch 为 200。采用小批量 AdamW 优化器，批量大小设为 64。学习率退火方法采用 PyTorch 中的“step”策略，初始学习率为 1e-3，分别在第 100 轮和第 150 轮时衰减（乘以 0.1）。然后，将第 200 次 epoch 训练更新后的网络权重及偏置参数进行保存，并将这些参数传递给主干网络_2，构建目标域预训练网络。其次，利用预定义的“set_freeze_by_id”函数将主干网络（分类器除外）中的指定层的参数更新机制进行解冻（如图 2 所示），并利用目标域的有标签训练集对主干网络_2 进行监督训练，epoch 为 100。考虑到，预训练和微调阶段面临的需求不同，预训练阶段是让模型学习

通用特征，而微调阶段则是让模型适应特定任务。因此，我们在这里对微调阶段的优化器与学习率参数进行了与预训练阶段不一样的设置。采用小批量 SGD 优化器，批量大小设为 64。学习率退火方法采用 PyTorch 中的“step”策略，初始学习率为 1e-4，分别在第 30 轮和第 60 轮时衰减（乘以 0.1）。这样选择的原因是，AdamW 作为自适应优化器，结合了动量（Momentum）和自适应学习率（RMSprop），收敛速度快且对超参数敏感度较低，适合快速初始化。而 SGD 配合动量能在局部最优解附近“冲出”低谷，获得更优泛化性能，尤其适合参数已初步收敛的预训练模型。

Freeze Status	
Layer: layer1.0.weight, Requires Grad: False	Layer: layer1.0.bias, Requires Grad: False
Layer: layer1.1.weight, Requires Grad: False	Layer: layer1.1.bias, Requires Grad: False
Layer: layer2.0.weight, Requires Grad: False	Layer: layer2.0.bias, Requires Grad: False
Layer: layer2.1.weight, Requires Grad: False	Layer: layer2.1.bias, Requires Grad: False
Layer: layer3.0.weight, Requires Grad: False	Layer: layer3.0.bias, Requires Grad: False
Layer: layer3.1.weight, Requires Grad: False	Layer: layer3.1.bias, Requires Grad: False
Layer: layer4.0.weight, Requires Grad: False	Layer: layer4.0.bias, Requires Grad: False
Layer: layer4.1.weight, Requires Grad: False	Layer: layer4.1.bias, Requires Grad: False
Layer: layer5.0.weight, Requires Grad: True	Layer: layer5.0.bias, Requires Grad: True

Fig 2. The requires_grad state of parameters at each layer of the basic network (the fine-tuning layer is the last layer).

b）Statistical Moment Matching (SMM)

对于这种以“特征适配”为核心思想的 DTL 基础范式，我们采用分阶段训练策略来提升迁移学习的稳定性和有效性。首先，在初始阶段，主干网络仅在源域训练集上进行监督学习，学习源域的特征表示和分类器参数。然后，在后续阶段，引入目标域的无标签训练数据，通过迁移学习方法对齐源域与目标域的特征分布，实现知识迁移。这样做的目的是让模型先掌握源域的知识，形成基础的特征提取能力和分类能力，避免在后续迁移中因模型参数不稳定而受到目标域未知分布的干扰。进而利用源域已学习的知识，结合目标域数据调整模型，使其适应目标域的分布，同时避免直接混合训练导致的“负迁移”（即源域知识被目标域噪声污染）。因此，我们在 SMM 的代码框架中，将总的 epoch 设置为 300 次，前 50 次 epoch 定义为第一阶段，第 50 次 epoch 之后定义为第二阶段。采用小批量 AdamW 优化器，批量大小设为 64。学习率退火方法采用 PyTorch 中的“step”策略，初始学习率为 1e-3，分别在第 100 轮和第 150 轮时衰减（乘以 0.1）。

由于 SMM 中可以使用的分布差异度量方法多样，包括 Maximum Mean Discrepancy (MMD)、CORrelation Alignment、Kullback-Leibler Divergence 等。其中，MK-MMD 通过组合多个核函数能够灵活捕捉源域与目标域之间的非线性分布差异，对复杂数据具有更强的适应性，且在过往研究中被证明在跨域任务中表现出较高的鲁棒性和有效性。因此，**我们在代码框架中以 MK-MMD 作为典型案例**，对 SMM 的算法流程进行设计，并将其引入损失函数以实现特征对齐，从而最终的损失函数可定义如下，

$$\mathcal{L}=\begin{cases}\mathcal{L}_c & epoch\leqslant 50\\ \mathcal{L}_c+\lambda_{\text{MK-MMD}}\mathcal{L}_{\text{MK-MMD}}\left(\mathcal{D}_s,\mathcal{D}_t\right) & epoch>50\end{cases} \tag{1}$$

其中，$\lambda_{\text{MK-MMD}}$ 是一个权衡参数。

我们对 $\lambda_{\text{MK-MMD}}$ 采用了 sigmoid 函数进行平滑增长，从 0 逐渐增加到 1。这种设计通过动态调整 $\lambda_{\text{MK-MMD}}$，在模型成熟后再逐渐引入目标域信息，使域适应损失在训练后期占据主导地位，适应过程更加稳定，符合“先学习任务，再适应领域”的分阶段训练总体逻辑。

$$\lambda_{\mathrm{MK-MMD}}=\frac{1-e^{-10\cdot\left(\frac{epoch-50}{300-50}\right)}}{1+e^{-10\cdot\left(\frac{epoch-50}{300-50}\right)}} \tag{2}$$

c）Domain Adversarial Training (DAT)

由于 DAT 的核心思想同样为“特征适配”，因此代码框架中的训练策略、epoch 次数、优化器与学习率参数均采用与 SMM 一致的设置。此外，在基于领域对抗的深度迁移学习方法中，常见的方法主要包括 domain adversarial neural network (DANN)、adversarial discriminative domain adaptation (ADDA)、conditional domain adversarial network (CDAN)等。其中，DANN 是最早将对抗学习引入迁移学习的方法之一，为后续研究奠定了基础。DANN 通过单一的梯度反转层和领域判别器构建对抗机制，无需复杂的生成器或多分支结构，计算成本低，便于在各类神经网络框架中集成。因此，**我们以 DANN 作为典型案例构建代码框架**，便于理解基于对抗的迁移学习流程，并为后续改进提供基础。

在代码框架中，我们通过在主干网络的基础上增加梯度反转层和领域判别器来构建 DANN，具体如图 3 所示。其中，梯度反转层在反向传播过程中对上游传来的梯度进行符号反转，让领域分类器的优化方向与特征提取器的优化方向完全相反，形成对抗训练，迫使特征提取器学习到“混淆领域分类器”的特征，从而无需显式设计复杂的对抗优化流程。领域判别器则由 three-layer Fc binary classifier 构成，The output features of these Fc layers are 1024 (FC1), 1024 (FC2), and 2 (FC3), respectively. The parameter of a dropout layer is p = 0.5。

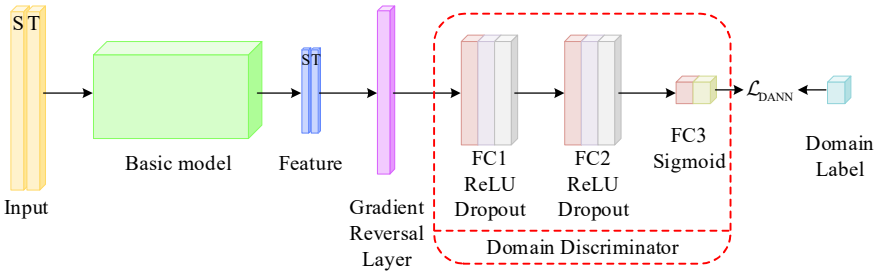


Fig 3. Structure of the DANN.

与此同时，自动将对抗思想引入损失函数，以实现特征对齐，从而最终的损失函数可定义如下，

$$\mathcal{L}=\begin{cases}\mathcal{L}_c & epoch\leqslant 50\\ \mathcal{L}_c-\lambda_{\mathrm{DANN}}\mathcal{L}_{\mathrm{DANN}}\left(\mathcal{D}_s,\mathcal{D}_t\right) & epoch>50\end{cases} \tag{3}$$

其中，λ_{DANN} 是一个权衡参数。

值得注意的是，λ_{DANN} 虽然与 $\lambda_{\mathrm{MK\text{-}MMD}}$ 都采用渐进式动态权重机制，但具体实施细节存在差异。$\lambda_{\mathrm{MK\text{-}MMD}}$ 是动态调整分布差异损失权重，且基于 epoch 进行更新，每个 epoch 内所有 batch 使用相同权重，属于粗粒度调整。而 DANN 的对抗损失权重本质上是通过 GRL 系数直接控制梯度，而非在损失函数中缩放。因此，λ_{DANN} 是动态调整梯度反转系数（GRL 强度），直接作用于梯度计算，且基于迭代次数进行更新，每个 batch 更新一次，使每个 batch 的对抗强度逐步增加，适应训练动态，以有效应对训练过程中批次数据间可能存在的分布波动。

$$\lambda_{\mathrm{DANN}}=\frac{1-e^{-10\cdot\left(\frac{\mathrm{iter\_num}}{(300\text{-}50)\times len(\mathrm{dataloader})}\right)}}{1+e^{-10\cdot\left(\frac{\mathrm{iter\_num}}{(300\text{-}50)\times len(\mathrm{dataloader})}\right)}} \tag{4}$$

2. 数据选择说明

通过深入调研当前公开可用的三电系统（电池、电机、电驱）健康状态异常数据集，我们发现现有开源数据均无法完全复现电动汽车实际运行中的复杂工况。基于此，本研究依据迁移

学习验证的根本原则对开源数据的可用性进行评估。迁移学习的核心价值在于解决“领域差异导致的模型失效”问题，其基础范式的验证需满足两个准则，

a）领域偏移的可控性：跨域数据应能有效表征源域与目标域的分布差异；

b）故障机制的普适性：所涉及的故障模式需反映实际系统的退化本质。

根据上述准则，三电系统健康维护领域存在显著的数据壁垒与验证可行性差异。

电机数据：如图 4 所示，电机系统存在定子、转子及轴承等故障易发子部件，其数据可用性分析如下，

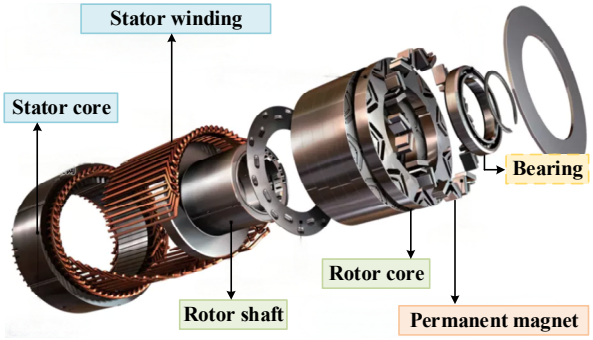


Fig 4. Components of the motor system (Taking PMSM as an example).

a）定子故障，Korea Advanced Institute of Science and Technology 在 2023 年公布的定子绕组短路故障数据集记录了三种功率等级的 PMSM（1.0 kW、1.5 kW 和 3.0 kW）在正常、线圈间短路及匝间短路状态下的振动与电流信号。该数据集虽尝试模拟多级故障严重程度，但**存在三重缺陷**：

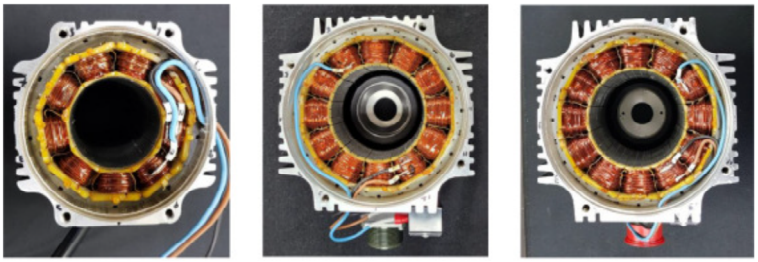
i）故障模拟方式是通过在 A 相相邻的两匝线圈间并联不同阻值的旁路电阻来等效故障的严重等级。将故障严重程度定义为旁路电路中的电流与正常电路中的电流之比。此方法与学界共识相悖，即匝间短路的故障严重程度主要取决于具体转速工况下的短路电流幅值，可近似表示为短路匝数百分比与短路接触电阻的组合。

ii）所有数据均在单一工况（3000rpm，1.5N.m）下采集，无法构建跨工况迁移任务。

iii）相同阻值的旁路电阻在不同功率等级电机中表征的物理严重程度不等价。因此，同样也就无法进行相同故障严重程度定子绕组短路故障模式下不同功率等级电机系统之间的跨域迁移设置。

因此，该开源数据集从本质上就无法满足“领域偏移的可控性”。

$$FI = \begin{cases} \dfrac{R_{it}}{R_{it} + R_{\text{bypass}}} \\ \dfrac{\mu}{\mu R_a + R_{\text{bypass}} - \mu^2 R_a} \end{cases} \tag{5}$$



(a) (b) (c)

Fig 5. Artificial fault seeded on A-phase of PMSMs with different powers: (a) 1.0 kW, (b) 1.5 kW,

and (c) 3.0 kW.

b）转子故障，Wuhan University 在 2018 年分布的关于正常、不平衡、不对中、摩擦四种转子状态（unbalanced, misaligned, and rubbing）的数据集记录了 148W 直流电机运行在 1200rpm 转速下的振动信号。受限于单一恒定工况，其缺乏数据分布偏移，无法构建迁移学习所需的“源域-目标域”结构，背离迁移学习解决的核心问题，从本质上便满足“领域偏移的可控性”这一硬性约束。

c）轴承故障，作为电机系统最常见的故障之一，占比高达 40%及以上。而且，轴承故障诊断作为电机（旋转机械）系统健康监测的经典问题，其研究历史可追溯至 1967 年美国机械故障预防小组(MFPG)的成立。经数十年发展，已形成多组权威开源数据集，比如，Case Western Reserve University (CWRU) Dataset、Paderborn University (PU) Dataset、JiangNan University (JNU) Dataset 等，其核心优势在于，

i）这些数据集的测试条件中大部分都涉及不同的运行工况（转速和载荷），可以轻松构建迁移学习所需的“源域-目标域”结构，满足“领域偏移的可控性”的约束条件。

ii）运行工况对故障状态下的振动信号的作用机制明确，即转速变化在振动信号上体现为由故障部件转频和振动幅值变化造成的差异，载荷变化在振动信号上则体现为由故障冲击能量导致的振动幅值差异，符合“故障机制的普适性”的要求。

综上，在现有数据约束下，轴承故障数据集因其工况可调性与物理机制明确，成为电机系统具备验证可行性的最佳开源资源，具体参见表 2。

Tab 2. Comparison of validation feasibility for open-source data in motor system.

	Dataset Source	Satisfies DTL Validation Criteria
Stator	Vibration and current dataset of stator faults	×
Rotor	Vibration dataset of rotor faults	×
Bearing	CWRU Dataset、PU Dataset、JNU Dataset	√

为科学验证 DTL 范式的跨域性能，我们最终决定选用 CWRU Dataset 作为基准，依据如下：

- 学界公认的验证基准：该数据集已被 IEEE Transactions 等顶级期刊广泛用于电机系统健康管理研究，其包含明确工况变化，支持构建 12 种标准化迁移任务，完美匹配“跨工况迁移”场景；
- 物理机制的工程普适性：转速和载荷变动对故障状态振动信号的影响具有明确物理规律。CWRU 虽非电动汽车用专用数据，但其故障物理机制与车载场景基本一致。而且，其故障损伤演进模拟（如故障直径分级）虽源自工业场景，但与车载电机轴承退化机制一致；
- 范式的普适性验证：通过多种迁移任务下的 Accuracy 定量对比，清晰揭示三种 DTL 范式的性能差异。

因此，我们认为此方案在现有数据局限下，最大程度保证了 DTL 基础范式性能对比实验的严谨性。

3. 补充实验的定量结果

经过上述分析，我们最终采用由 CWRU 轴承数据中心提供的 CWRU 数据集。与其他相关研究一致，我们采用采样频率为 12kHz 的驱动端轴承故障数据。如表 3 所示，根据故障尺寸差异，我们数据集划分为十个类别，包括 1 种正常轴承状态（NA）和 9 种故障状态。故障状态涵盖三种故障类型：内圈故障（IF）、外圈故障（OF）和滚珠故障（BF），其中不同的故障尺寸等效模拟了故障损伤的恶化趋势。此外，如表 4 所示，CWRU 数据集的测试条件包含 4 种不同的运行工况（对应不同的转速和载荷组合）。在跨域维护任务设定中，这些工况可视为不同的域。因此，我们可以在任意两种不同工况间构建深度迁移学习所需的“源域-

目标域”结构，共计形成 12 项跨域任务。其中，Task1(0➞1)表示从转速为 1797rpm、载荷为 0HP 的源域迁移至转速为 1772rpm、载荷为 1HP 的目标域。

Tab 3. Description of class labels of CWRU.

Class Label	0	1	2	3	4	5	6	7	8	9
Fault Location	NA	IF	BF	OF	IF	BF	OF	IF	BF	OF
Fault Size (mils)	0	7	7	7	14	14	14	21	21	21

Tab 4. Operating conditions and cross-domain tasks of CWRU.

Operating conditions	0			1			2			3	
Speed (rpm)	1797			1772			1750			1730	
Load (HP)	0			1			2			3	
Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Task 11	Task 12
0➞1	0➞2	0➞3	1➞0	1➞2	1➞3	2➞0	2➞1	2➞3	3➞0	3➞1	3➞2

我们直接采用原始时域信号作为网络输入。考虑到 4 种运行工况中的最小转频为 28.83Hz（对应转轴旋转一周的采样点数约为 417 个），为确保每个样本在不同工况下至少包含 2 个转子旋转周期的信息，同时保证充足的训练样本数量，我们设置滑动窗长为 1024，滑动步长为 256 进行滑动采样。随后，为避免样本不平衡对不同 DTL 基础范式性能对比的干扰，我们从每个类别的所有样本中随机抽取 256 个样本，确保各子类别样本数量均衡。接着，我们对时域输入样本进行 Z 分数归一化预处理，并将总样本的 70%划分为训练集，30%划分为测试集，以防止测试数据泄露。上述数据处理流程均应用于源域数据与目标域数据。

为系统评估不同 DTL 基础范式在跨域维护任务中的泛化能力，并确保实验结果稳定性、鲁棒性及模型间对比的公平性，我们实施了以下三重保障措施以消除随机因素干扰。首先，为确保实验可重复性，我们在代码框架中通过自定义的“setup_seed(seed)”函数固定所有可能的随机源（包括 PyTorch、NumPy、Python 原生随机数及 CUDA 环境等）。每次实验均使用唯一且固定的随机种子（seed=43,44,45,46,47），确保数据划分（如训练/测试集分割）、模型参数初始化、数据增强方式等所有含随机性的环节完全一致。其次，模型每次训练 300 个 epoch，并记录每个 epoch 的准确率以观察长期收敛性。鉴于训练初期诊断准确率可能因优化器震荡或过拟合/欠拟合而波动较大，而后期（接近收敛时）的结果更能反映模型的真实性能。因此，我们选取最后 10 个 epoch 的平均准确率作为单次实验的性能指标，以此获得该随机种子下模型的“稳定性能估计”。最后，重复进行 5 次独立实验（每次使用不同的固定种子），并对单次实验的稳定性能估计值进行二次平均，得到最终的平均准确率。此方法有效评估了算法在不同初始化条件下的整体表现，降低了随机性干扰，确保了结果的普遍性和可靠性。值得注意的是，所有实验均在 Windows 10 系统及 PyTorch 2.3.1 环境下执行，运行平台为配备英特尔酷睿 i7-12700KF 处理器、英伟达 GeForce RTX 4060Ti 显卡及 16GB 内存的计算机。

1）基线模型的领域偏移脆弱性

如图 6 所示，基线模型（即未引入深度迁移学习的模型）在不同跨域维护任务的源域测试集上均达到 99%以上的诊断准确率，表明其具备优异的域内诊断能力。然而，当直接将源域预训练模型应用于目标域测试集时，其性能在全部 12 项跨域维护任务中均出现了不同程度的退化。在轻度偏移任务（如 T0➞T1：源域和目标域载荷相差 1HP）中，准确率下降约 3%（99.94%➞97.03%）。而在重度偏移任务（如 T0➞T3：源域和目标域载荷相差 3HP）准确率骤降至 76.41%（T10）甚至 73.13%（T11），最大降幅达 27%。此外，图 7 展示了基线模型在 12 项跨域任务目标域测试集上多次测试的诊断准确率箱型图。图中显示基线模型性能波动剧烈，在部分跨域任务中（如 T6）中，Min-Max 差值高达 18.47%（73.44%-91.91%），表明模型对目标域数据分布极度敏感。**综上，上述结果证实领域分布差异导致基线模型泛化能**

力严重劣化。

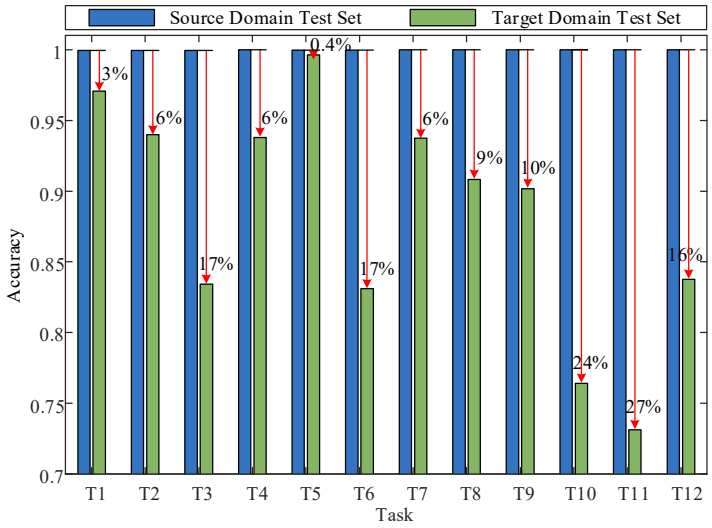


Fig 6. Mean diagnostic accuracy of the baseline model on source domain and target domain test sets in different cross-domain tasks.

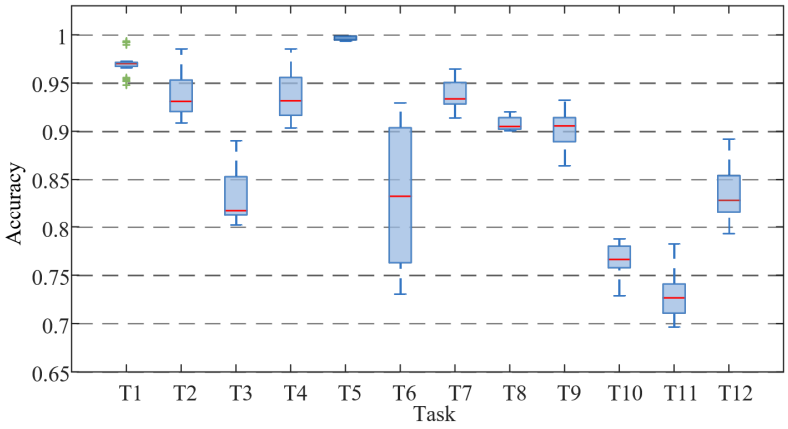


Fig 7. Box plot of diagnostic accuracy of the baseline model on target domain test sets of 12 cross-domain tasks.

2）DTL 范式的必要性验证

表 5 汇总了基线模型与三种 DTL 基础范式（PTFT、SMM、DAT）在所有跨域任务中 5 次随机测试中的平均、最大及最小诊断准确率。数据显示，三种 DTL 基础范式在所有跨域任务中的性能均显著超越基线模型。在重度偏移任务（如 T11）中，DTL 方法（PTFT:98.77%, SMM:97.06%, DAT:98.49%）的平均诊断准确率较基线模型（73.13%）提升超 23%。即使在轻度偏移任务（如 T1）中，DTL 仍将平均诊断准确率从 97.07%提升至 99.56%以上。我们通过三维混淆矩阵对跨域任务 T10 中基线模型与三种 DTL 基础范式在 10 种类别上的分类结果进行可视化（图 8）。图中显示，基线模型在 Class_label_3、Class_label_5、Class_label_9 三个类别上的诊断准确率极差。其中，Class_label_3 中有 41.56%的样本被误诊断为 Class_label_7，Class_label_5 中有 93.51%的样本被误诊断为 Class_label_7，而 Class_label_9 的所有样本均被误诊断。相比之下，DTL 基础范式（无论是通过参数迁移还是特征适配）均显著提升了这三类在基线模型中被严重误诊的子类别的准确率。此外，三种 DTL 基础范式的 Min-Max 差值普遍≤2%（如 PTFT 在 T12 的 100%-100%），远低于基线模型的最大差值（18.47%）。**因此，上述结果充分表明，DTL 通过领域自适应机制有效抑制了分布偏移的影**

响，验证了其在跨域诊断中的必要性。

Tab 5. Diagnostic accuracy of three basic paradigms of DTL in different cross-domain tasks.

Method	T1-Accuracy(%)			T2-Accuracy(%)			T3-Accuracy(%)			T4-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	97.07	99.21	95.23	93.99	98.40	91.16	83.41	88.66	80.29	93.78	97.79	90.63
PTFT	**99.92**	100	99.87	99.87	100	99.79	**99.97**	100	99.87	**99.22**	99.74	98.76
SMM	99.56	99.74	99.19	99.79	100	99.41	99.86	100	99.71	98.99	99.51	98.66
DAT	99.84	100	99.65	**99.91**	100	99.75	99.74	99.97	99.48	98.94	99.61	98.61
Method	T5-Accuracy(%)			T6-Accuracy(%)			T7-Accuracy(%)			T8-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	99.63	99.87	99.44	83.10	91.91	73.44	93.74	95.49	91.97	90.83	91.85	90.23
PTFT	99.95	100	99.87	**99.99**	100	99.96	97.92	99.23	96.29	**99.09**	99.30	98.80
SMM	99.95	100	99.87	99.76	99.90	99.58	97.85	98.92	96.55	98.95	99.47	98.22
DAT	**99.97**	100	99.87	99.67	99.87	99.27	**97.94**	98.66	97.12	98.48	98.72	98.19
Method	T9-Accuracy(%)			T10-Accuracy(%)			T11-Accuracy(%)			T12-Accuracy(%)		
	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max	Min
Base	90.17	92.68	87.98	76.41	78.19	73.39	73.13	78.15	69.78	83.77	88.75	80.04
PTFT	**100**	100	100	96.24	97.04	95.47	**98.77**	99.23	97.89	**100**	100	100
SMM	99.98	100	99.9	97.15	98.48	95.57	97.06	98.54	96.12	99.93	100	99.82
DAT	99.95	100	99.87	**97.91**	98.19	97.27	98.49	98.83	98.26	99.78	100	99.38

Remarks: T1-Accuracy (%) represents the accuracy in the Task 1. Mean is the mean accuracy of the 5 repeated experiments; Max is the maximum accuracy and Min is the minimum accuracy.

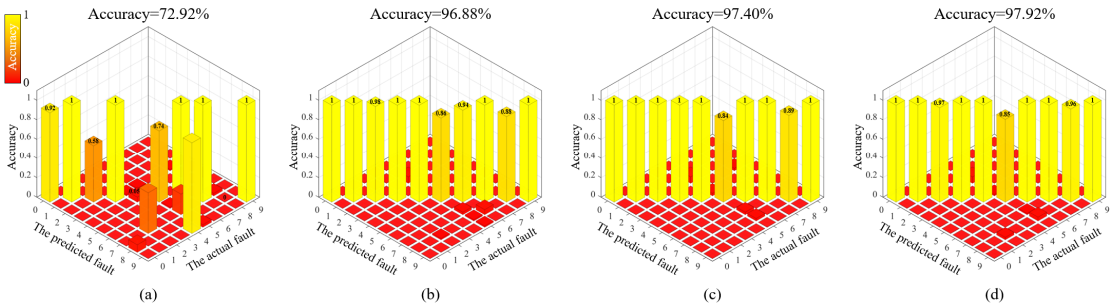


Fig 8. Three-dimensional confusion matrix in the T10. (a)Base; (b) PTFT; (c) SMM; (d) DAT.

3）范式性能的任务依赖性分析

图 9 展示了基线模型与三种 DTL 基础范式（PTFT、SMM、DAT）在 12 项跨域任务中 5 次随机测试的平均诊断准确率。观察可知，三种 DTL 基础范式在不同跨域任务中表现各异，不存在绝对最优的范式。这种差异可能与任务特性差异、数据内在关联及领域偏移的方向性（如载荷增加方向 0→3 HP 与减少方向 3→0 HP）有关，导致最优范式随迁移方向变化。因此，DTL 基础范式的有效性高度依赖于具体跨域场景的特性（如偏移类型、故障与工况耦合关系），需要结合任务特点选择适配的范式。

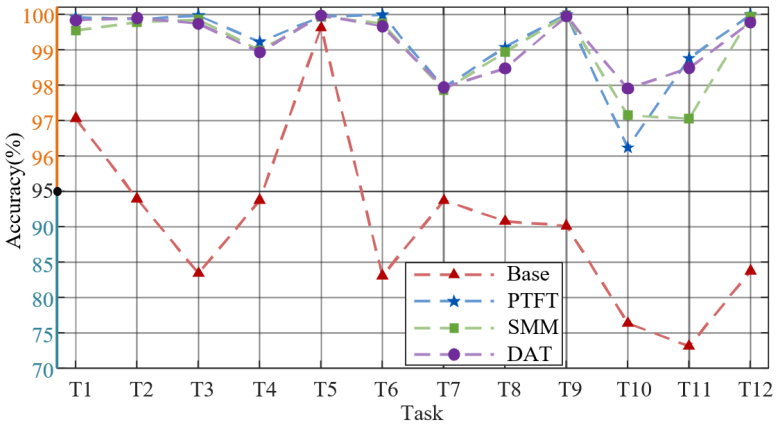


Fig 9. Mean diagnostic accuracy of three basic paradigms of DTL in different cross-domain tasks.

综上所述，实验结果证明：

- DTL 相比基线模型具有显著必要性；
- 范式选择需结合任务内在特性。

更重要的是，**上述分析的核心目标在于验证 DTL 基础范式在跨域任务中的必要性与普适有效性，而非寻求在所有跨域任务中的单一最优解**。此结论也更符合工程实际—实际电动汽车三电系统的跨域场景具有高度异构性，需灵活适配不同的 DTL 范式。此外，尽管当前对比实验基于 CWRU 轴承数据，但其验证的深度迁移学习机制（针对数据分布偏移/特征空间差异）与电动汽车系统维护的跨域本质一致。迁移学习的验证核心在于算法对分布偏移的鲁棒性，而非特定场景的复现，且所采用的代码框架可无缝扩展至其他系统。因此，本研究为深度迁移学习在电动汽车三电系统维护中的落地应用提供了方法论支撑与性能基准。