PCDC: Prototype-assisted dual-contrastive learning with depthwise separable convolutional neural network for few-shot fault diagnosis of permanent magnet synchronous motors under new operating conditions

Minseok Chae\textsuperscript{a,1}, Hyeongmin Kim\textsuperscript{a}, Hye Jun Oh\textsuperscript{a}, Chan Hee Park\textsuperscript{a}, Chaehyun Suh\textsuperscript{a}, Heonjun Yoon\textsuperscript{b,*}, and Byeng D. Youn\textsuperscript{a,c,d,*}

\textsuperscript{a} Department of Mechanical Engineering, Seoul National University, Seoul 08826, Republic of Korea
\textsuperscript{b} School of Mechanical Engineering, Soongsil University, Seoul 06978, Republic of Korea
\textsuperscript{c} OnePredict Inc., Seoul 06160, Republic of Korea
\textsuperscript{d} Institute of Advanced Machines and Design, Seoul National University, Seoul 08826, Republic of Korea
\textsuperscript{*} Corresponding Author: heonjun@ssu.ac.kr and bdyoun@snu.ac.kr

Abstract

The fault diagnosis of permanent magnet synchronous motor is of vital importance in industrial fields to ensure user safety and minimize economic losses from accidents. However, recent fault diagnosis methods, particularly the methods using deep learning, require a massive amount of labeled data, which may not be available in industrial fields. Few-shot learning has been recently applied in fault diagnosis for rotary machineries, to alleviate the data deficiency and/or to enable unseen fault diagnosis. However, two major obstacles still remain, specifically: a) the limited ability of the models to be generalized for use under new operating conditions and b) insufficient discriminative features to precisely diagnose fault types. To address these limitations, this study proposes a Prototype-assisted dual-Contrastive learning with Depthwise separable Convolutional neural network (PCDC) for few-shot fault diagnosis for permanent magnet synchronous motors under new working conditions. Operation-robust fault features are extracted to reinforce generalization of PCDC under new operating conditions by extracting fault-induced amplitude and frequency modulation features and by eliminating the influence of operating conditions from the motor stator current signals. Prototype-assisted dual-contrastive learning is proposed to clearly distinguish the fault categories even when the fault features are similar to each other by learning both local- and global-similarity features, which increases the instance-discrimination ability while alleviating an overfitting issue. Experimental results show that the proposed PCDC outperforms the comparison models in few-shot fault diagnosis tasks under new operating conditions.

Keywords: Motor fault diagnosis, Motor stator current, Few-shot learning, Unseen faults, New operating conditions
List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D^\text{meta-min}$</td>
<td>Source-domain dataset consisting of a support set and a query set</td>
</tr>
<tr>
<td>$S_{\text{src}}, Q_{\text{src}}$</td>
<td>Support set and query set of source domain dataset</td>
</tr>
<tr>
<td>$x^S_i, x^Q_i$</td>
<td>Support and query sample of source domain dataset</td>
</tr>
<tr>
<td>$y^S_i, y^Q_i$</td>
<td>Support and query label of source domain dataset</td>
</tr>
<tr>
<td>$D^\text{meta-test}$</td>
<td>Target-domain dataset consisting of a support set and a query set</td>
</tr>
<tr>
<td>$S_{\text{tar}}, Q_{\text{tar}}$</td>
<td>Support set and query set of target domain dataset</td>
</tr>
<tr>
<td>$(x^<em>)^S_i, (x^</em>)^Q_i$</td>
<td>Support and query sample of target domain dataset</td>
</tr>
<tr>
<td>$(y^<em>)^S_i, (y^</em>)^Q_i$</td>
<td>Support and query label of target domain dataset</td>
</tr>
<tr>
<td>$P^c_\theta(x)$</td>
<td>Prototype embeddings of $c$-th class parameterized by encoder $\theta$</td>
</tr>
<tr>
<td>$F_\phi(x)$</td>
<td>Instance embeddings parameterized by encoder $\phi$</td>
</tr>
<tr>
<td>$J_{\text{tot}}$</td>
<td>Total loss function</td>
</tr>
<tr>
<td>$J_{IB}$</td>
<td>Instance block loss function</td>
</tr>
<tr>
<td>$J_{PB}$</td>
<td>Prototype block loss function</td>
</tr>
<tr>
<td>$d$</td>
<td>Euclidian distance</td>
</tr>
<tr>
<td>$N$</td>
<td>The number of classes</td>
</tr>
<tr>
<td>$K$</td>
<td>The number of data per each class</td>
</tr>
</tbody>
</table>

1. Introduction

Permanent magnet synchronous motors (PMSMs) have been widely used in mechanical drive systems, such as electric vehicles and industrial robots, due to their precision control and high efficiency (Gangsar & Tiwari, 2020; Jang et al., 2023). As PMSMs are used in various applications, reliable fault diagnosis is of vital significance to ensure user safety and minimize economic losses from accidents. In the literature, diverse methods have been proposed for motor fault diagnosis; existing methods can be categorized into data-driven methods (Feng, Ji, et al., 2023; K. Feng et al., 2022; Feng, Xu, et al., 2023; Kong, Cai, Liu, Zhu, Yang, et al., 2022; Oh et al., 2022; Raouf et al., 2022; Yang, Cai, Wu, et al., 2023; Yang, Cai, Zhang, et al., 2023) and model-based methods (Gao et al., 2015; Isermann, 2005; Kong, Cai, Liu, Zhu, Liu, et al., 2022). Because servo motor drive systems gradually accumulate condition monitoring data (i.e., motor stator current signals), many data-driven methods have been developed in recent years. In particular, deep learning is widely used for data-driven methods due to its accurate fault diagnosis capability of autonomous feature extraction in which it can directly learn optimal features from the input signal for fault diagnosis (Kim et al., 2023; Ko et al., 2023; Wang et al., 2023).
While deep learning-based models have led to a series of breakthroughs in a fault diagnosis field, these methods still have limitations. First, conventional deep learning-based models require a massive amount of labeled data to ensure high generalization performance and convergence (Ma et al., 2023). In industrial fields, however, obtaining sufficient labeled data is difficult, as motors usually operate under a normal status (Han et al., 2023; Jian & Ao, 2023), and labeling massive data for each fault type is an expensive and laborious task (Hu et al., 2020). Second, conventional deep learning-based models are hard to diagnose previously unseen motor faults, because the model assumes that the source and target domains consist of the same fault type (Y. Feng, J. Chen, J. Xie, et al., 2022; Han & Li, 2022). When a new motor is installed in an industrial site, unseen motor faults can occur due to the lack of prior failure history. Third, the fault diagnosis performance of these models could decrease under variable operating conditions and the resulting data distribution discrepancy (Chen et al., 2023; Y. Li et al., 2022; Park et al., 2023). Because motors in manufacturing equipment operate at various loads and speeds depending on a production process, building a generalized model that can diagnose faults in all operating conditions is challenging (X.-y. Zhang et al., 2022).

The current mainstream methods to deal with the challenges mentioned above can be categorized into transfer-learning (TL)-based and few-shot learning (FSL)-based approaches (T. Zhang et al., 2022). For TL-based approaches, parameter transfer learning and feature-representation transfer learning have been widely used (Hwang et al., 2023). Parameter transfer learning is a method that fine-tunes a pre-trained model using small target domain dataset to form an improved target learner. Feature-representation transfer learning models aim to minimize distribution discrepancies between domains in the feature space. Kim et al. (Kim & Youn, 2019) proposed a new parameter repurposing method to alleviate an overfitting problem in a small dataset. By choosing output-sensitive parameters from the source learner and selectively freezing or fine-tuning the parameters, the informative parameters are protected from the overfitting to a small dataset. Kim et al. (Kim et al., 2022) introduced a semantic-clustering-based domain adaptation technique for bearing fault diagnosis to extract features that are invariant between different working conditions. However, TL-based models have several drawbacks: First, TL-based models require features extracted from pre-trained models; this is a major obstacle to their adoption, because pre-trained models cannot be obtained in some cases, especially in real-world settings (Ruan et al., 2021). Second, an additional training process is required to extract domain-invariant features in the source and target domains, but it is hard to be performed in most real industrial environments (Zhang et al., 2020). For example, for a motor found in military robots, deep learning model parameters are difficult to be redesigned with new data in advance because the motor operates under arbitrary speed and load profiles. Finally, the fault diagnosis performance of the model may decrease due to the negative transfer that could arise due
to target domain interference in the additional training process (Ribani & Marengoni, 2019).

FSL-based approaches have been recently applied in environments where labeled data is scarce in both the source and target domains, since FSL algorithms can learn from small dataset with no need for the pre-trained models or the additional training by using meta-learning (Wu et al., 2020). Also, FSL models can diagnose unseen fault data which were not observed in the source domain. In the FSL perspective, unseen fault data in the target domain refers to both labeled- and unlabeled-data on which the model was not trained. FSL assumes that the labeled data in the target domain cannot be used in the training process as it is difficult to collect and learn the target domain data in advance, specially when the operating condition changes from time to time. Further, FSL algorithms can be used with different operating conditions without additional model training by using learned fault features from the source domain dataset (C. Li et al., 2022; Long et al., 2023). Following this idea, Zhang et al. developed an efficient FSL algorithm based on stacked sparse autoencoders and a Siamese network to detect interturn short-circuits in PMSMs under limited-data conditions (J. Zhang et al., 2021). Wang et al. proposed a hybrid approach, which is a mixture of a generalized supervised-learning model and a metric-based, meta-learning model to overcome the few-shot fault diagnosis problem under the scarce data environments (Wang et al., 2021). For an unseen fault diagnosis problem in a few-shot setting, Zhang et al. introduced a few-shot fault diagnosis method based on model-agnostic meta-learning (MAML) that showed the improved model generalization performance for diagnosing unseen bearing faults (S. Zhang et al., 2021). For the operating condition shift problem in a few-shot setting, Wang et al. introduced a metric learning module and a label-smoothing technique to predict the similarity of the samples, even under different operating conditions (Wang & Xu, 2021). Li et al. used time-frequency images as MAML model inputs to make the model robust for different operating conditions and verified the proposed method using bearing datasets (Li et al., 2021).

While these studies offered important contributions, existing few-shot fault diagnosis methods are not sufficiently discriminative under target domain conditions, especially when the fault features are similar. Also, as the existing methods show the insufficient performance under new operating conditions and are mainly evaluated under the small changes in speed or torque levels, it is needed to develop a method which can generalize well on the large changes in operating conditions. Furthermore, to the best of our knowledge, there is no FSL study focusing on evaluating its generalized performance for unseen fault diagnosis of motors under new operating conditions.

This paper thus proposes a novel FSL-based PMSM fault diagnosis method, named Prototype-assisted dual-Contrastive learning with Depthwise separable Convolutional neural network (PCDC). PCDC consists of an operation-robust fault feature extraction module (OFM) and a few-shot diagnosis module (FDM). OFM,
which consists of preprocessing and depthwise separable convolution, helps to reinforce generalization of PCDC for use with new operating conditions and make PCDC learn operation-robust fault features from the motor stator current signals. The preprocessing obtains pairwise scaled inputs from the motor stator current signals and eliminates the effect of operating conditions (i.e., the torque and speed levels). The depthwise separable convolution, which is suitable for the FSL algorithm as it uses fewer model parameters in a limited-data condition, extracts fault features from the pairwise scaled inputs. In FDM, prototype-assisted dual-contrastive learning proceeds with the extracted fault-induced features to learn both the global- and local similarity features from the prototypes and instance embeddings, respectively. By extracting prototype-instance contrastive features where instance embeddings are transferred from another contrastive learning space, the global- and local-similarity features are learned. Therefore, the proposed model becomes more discriminative on the small dataset while alleviating the overfitting, and thereby improving the fault diagnosis performance. The main contributions of the paper are summarized as follows. The main contributions of the paper are summarized as follows.

1) Operation-robust fault features are extracted to reinforce generalization of PCDC for fault diagnosis under new operating conditions by extracting fault-induced amplitude and frequency modulation features and by eliminating the influence of operating conditions from the motor stator current signals.

2) Prototype-assisted dual-contrastive learning is proposed to clearly distinguish the fault categories even when the fault features are similar to each other by learning both local- and global similarity features, which increases instance-discrimination ability while alleviating the overfitting problem.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts of supervised FSL. Section 3 describes the few-shot problem and the proposed PCDC for PMSMs fault diagnosis. Section 4 illustrates the experiments carried out to verify the effectiveness of the proposed fault-diagnostic method. The conclusions are summarized in Section 5.

2. Background

2.1. Supervised few-shot learning

The objective of supervised FSL is to generalize unseen tasks \( \mathcal{I} \) using the knowledge obtained from a training dataset where the amount of labeled samples is scarce (Wang et al., 2020). While traditional supervised-learning methods require a massive amount of data to learn the features of each fault category, supervised FSL methods can learn with small amount of data as they only need to learn whether a set of sample pairs belong to the same category or not. The terminology used in FSL is
summarized in Table 1 for easier understanding. The concept of few-shot fault diagnosis is described in Figure 1. For data acquisition, motor stator current signals are obtained by fault categories. Each fault category is acquired in a different operating condition (i.e., the torque and speed levels). Data is divided into seen and unseen fault data, and only seen fault data is used in meta-train stage. When a fault that was not present in the source domain emerges in the target domain, it is referred to as an unseen fault in the few-shot learning (FSL) perspective. Both seen and unseen fault data consist of labeled data (support set) to make model predictions and unlabeled data (query set) for comparison with the model predictions. Even though the data has labels, it can be unseen data if the FSL model did not learn the data during the training process. In meta-train stage, the FSL model learns similarity features between seen fault data in a supervised manner. In meta-test stage, the learned FSL model diagnoses faults by comparing the similarity score for each class even though the fault data were not observed in the meta-train stage.

In supervised FSL, a model trained on multiple tasks \( \mathcal{I} \) which consists of the meta-train dataset \( D_{\text{meta-train}} \), the meta-test dataset \( D_{\text{meta-test}} \) and the loss function \( L_3 \), which are expressed as:

\[
\mathcal{I} = \{D_{\text{meta-train}}, D_{\text{meta-test}}, L_3\}
\]

\[
D_{\text{meta-train}} = \{(x_i^S, y_i^S, x_i^Q, y_i^Q)\}_{i=1}^{N+uK}
\]

\[
D_{\text{meta-test}} = \{(x_i^Q, (x_i^*)^Q, (y_i^*)^S)\}_{i=1}^{N+uK}
\]

where \( x_i^S \) and \( x_i^Q \) indicate the support and query sets which are randomly sampled from meta-train dataset, respectively. \( (x^*_i)^S \) and \( (x^*_i)^Q \) indicate the support and query sets randomly sampled from meta-test dataset, respectively. The FSL model maps \( D_{\text{meta-train}} \) and \( D_{\text{meta-test}} \) to the query sets label \( (y^*_i)^Q \) of meta test dataset. In contrast, traditional supervised-learning tasks can be expressed as \( \mathcal{I} = \{D_{\text{train}}, D_{\text{test}}, L\} \) where train dataset \( D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n} \) and test dataset \( D_{\text{test}} = \{(x_i)\}_{i=1}^{m} \), where \( n \) and \( m \) denote the total number of samples in each dataset. Traditional supervised-learning models understand single task with a specifically designed neural network. The model maps input data \( x_i \) to label \( y_i \). The goal of supervised FSL and traditional supervised learning is to find optimal deep learning model parameter \( \Phi^* \) using training dataset to generalize for test dataset, as described in equations (4) and (5), respectively.
Supervised FSL: \[ \phi^* = \arg \max_{\phi} P\left( \phi \mid D_{\text{meta-train}}, D_{\text{meta-test}} \right) \] (4)

Traditional supervised learning: \[ \phi^* = \arg \max_{\phi} P\left( \phi \mid D_{\text{train}} \right) \] (5)

Figure 1. Concept of few-shot fault diagnosis
Table 1. Terminology used in few-shot learning

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-way K-shot</td>
<td>The number of classes ($N$) and the number of data ($K$) per each class for a single batch</td>
</tr>
<tr>
<td>Support set (N-way K-shot)</td>
<td>Support dataset with data and labels used to make the model prediction</td>
</tr>
<tr>
<td>Query set (N-way K-shot)</td>
<td>Query dataset with data and labels used for comparisons with the model prediction</td>
</tr>
<tr>
<td>Unseen fault</td>
<td>A new fault emerging in the target domain (meta-test stage) that was not present in the source domain (meta-train stage)</td>
</tr>
</tbody>
</table>

2.2. Metric-based meta-learning algorithms

FSL is a subset of meta-learning, which is largely divided into two frameworks: metric-based and optimization-based meta-learning approaches. Metric-based meta-learning algorithms are frequently used in FSL classification tasks due to their expressive and easy optimization (Wang et al., 2020). These models aim to learn how to effectively measure the similarity between support and query sets; then, prediction is achieved based on learned similarity (metric) features. Representative models include the Prototypical network (Proto Net) (Snell et al., 2017), Siamese network (Siamese Net) (Zhou et al., 2020), and Relation network (Relation Net) (Sung et al., 2018) approaches. These models consist of a feature extractor and a metric embedding function. However, the methods for calculating the similarity are different from each other. Schematics of each model are described in Figure 2. A support set $S$ of $n$ number of samples and a query set $Q$ of $m$ number of samples can be expressed as:

$$S = \{(x_1^S, y_1^S), \ldots, (x_n^S, y_n^S)\} \quad (6)$$

$$Q = \{(x_1^Q, y_1^Q), \ldots, (x_m^Q, y_m^Q)\} \quad (7)$$

where $x_i^S, y_i^S \in \mathbb{R}^M$ is an $M$-dimensional vector of an input sample and $y_i^S, y_i^Q = \{1, 2, \ldots, C\}$ are the corresponding labels where $C$ is the total number of classes.

Proto Net computes an $E$-dimensional feature vector of the $c$-th class ($p_c \in \mathbb{R}^E$), which is called a “prototype”, using a feature extractor and an embedding function $h_\phi$:
The prototype is the mean feature vector of the embedded support samples belonging to each class, which is expressed as:

\[ p^c = \frac{1}{|S_i|} \sum_{(x_i, y_i) \in S_i} h_\phi(x_i) \] (8)

With a query point \( x_i^Q \) and L2 distance function \( d \), the probability distribution over classes for a query point \( x_i^Q \) based on the distances to the prototypes is obtained with a softmax activation function as:

\[ P_\phi(y = c|x_i^Q) = \frac{\exp\left(-d\left(h_\phi(x_i^Q), p_c\right)\right)}{\sum_{j=1}^{c} \exp\left(-d\left(h_\phi(x_j^Q), p_j\right)\right)} \] (9)

Then, the negative log-likelihood loss function for query point \( x_i^Q \) of the \( c \)-th true class is denoted as:

\[ L_{\text{Proto}}(\phi) = -\log P_\phi\left(y = c|x_i^Q\right) \] (10)

The data is gradually clustered into each category as the optimization progresses in the training stage. In the test stage, the trained model infers the predicted categories based on the closest prototype to which the query sample belongs.

A Siamese Net consists of twin networks \( h_\phi: \mathbb{R}^M \rightarrow \mathbb{R}^E \) that accept distinct pairwise inputs \((x^S, x^Q)\). Training datasets are divided into positive and negative pairs that are randomly chosen from the same categories and different categories, respectively. The embedding vectors \( h_\phi(x^S) \) and \( h_\phi(x^Q) \) are joined in a differencing layer to measure the similarity between the embedding vectors. A similarity score is calculated using the L1 distance function, and can be expressed as \( \|h_\phi(x^S) - h_\phi(x^Q)\|_1 \). A contrastive loss function is used for metric learning where the function decreases the distance loss of positive pairs and increases the distance loss of negative pairs as described in equation (11):

\[ L_{\text{contrastive}}(x_i^S, x_j^Q, \phi) = \mathbb{I}[y_i = y_j] \|h_\phi(x_i^S) - h_\phi(x_j^Q)\|_1 \\
+ \mathbb{I}[y_i \neq y_j] \max\left(0, m - \|h_\phi(x_i^S) - h_\phi(x_j^Q)\|_1\right) \] (11)

where \( m \) is the margin value in the negative pairs, which is used for making the distance between the samples of a different class to be no less than \( m \). The Siamese Net approach
learns the similarity and dissimilarity metric features from pairwise inputs during model training. In the test stage, the predicted class is selected as the class that is most similar to the query sample by ranking the measured distance.

Relation Net uses a deep metric embedding function to directly measure the distance between the support and query samples, in contrast to the other models. The network consists of a feature extractor $g_{\phi}(x)$ and a deep metric embedding function $m_{\phi}(x)$. Relation Net outputs a relation score that represents the degree of the similarity between two samples. The relation score $r_{i,j}\in(0,1)$ of pairwise inputs is calculated as shown in equation (12):

$$r_{i,j} = m_{\phi}\left(\text{cat}\left(g_{\phi}(x^S_i), g_{\phi}(x^Q_j)\right)\right)$$  \hspace{1cm} (12)

where $\text{cat}$ refers to the concatenation operation of the feature maps. The learning goal is a regression of the relation score to the label $y_i\in(0,1)$, with mean square error (MSE), where 0 is for negative pairs and 1 is for positive pairs. The loss function with MSE is described in equation (13). The Relation Net infers the predicted label by the highest relation score: $\hat{y}_j = \arg\max_c r_{c,j}$.

$$L_{\text{Relation}} = \sum_{i=1}^{n}\sum_{j=1}^{n}(r_{i,j} - 1(y_i = y_j))^2$$  \hspace{1cm} (13)
3. Proposed Method: Prototype-assisted dual-Contrastive Learning with Depthwise separable Convolutional Neural Network (PCDC)

3.1. Problem description

In actual industrial fields, gathering motor fault data is not easy, and even previously unseen faults can occur. In the FSL perspectives, unseen fault data in the target domain refer to both labeled- and unlabeled-data which the FSL model did not learn during the training process. Also, a variation in working conditions causes discrepancies in the data distribution and thereby leads to performance drops in deep learning-based fault diagnosis. The problem of few-shot fault diagnosis including unseen fault categories under new working conditions is investigated in this paper. In order to reflect the conditions found in actual industrial fields, two different datasets are considered: a source-domain dataset $D_{src}$ and a target-domain dataset $D_{tar}$. These datasets are built on the basis of an $N$-way $K$-shot protocol from the few-shot learning paradigm (Finn et al., 2017). The source-domain dataset $D_{src}$ consists of a labeled single-phase stator current signal acquired from working condition (A). The source-domain dataset $D_{src}$ consists of support set $S^{src} = \{(x^s_i, y^s_i)\}_{i=1}^{N*K}$ and query set $Q^{src} = \{(x^q_i, y^q_i)\}_{i=1}^{N*K}$, where $N$ is the number of known fault classes and $K$ is the number of samples per class. The target-domain dataset $D_{tar}$ consists of an unlabeled single-phase stator current signal acquired from new working condition (B).
dataset consists of support set \( S_{\text{tar}} = \{(x^*_u, y^*_u)\}_{u=1}^{(N+u)K} \) and query set \( Q_{\text{tar}} = \{(x^*_u, y^*_u)\}_{u=1}^{(N+u)K} \), where \( u \) is the number of unseen fault classes. In this case, the target domain data \( (x^*_u)_{\text{tar}} \) and \( (x^*_u)_{\text{iq}} \) are unseen data. Then, the goal of the proposed method is to find query label \( (y^*_u)_{\text{iq}} \) in the target-domain. While \( D_{\text{src}} \) is used in model training, \( D_{\text{tar}} \) is used to test the fault diagnosis performance.

### 3.2 Proposed Framework

The framework for PCDC is described in Figure 3. PCDC consists of the operation-robust fault feature extraction module (OFM) and the few-shot diagnosis module (FDM). OFM is designated to enhance the generalization performance of PCDC for new operating conditions. FDM is designated to make PCDC more discriminative on small dataset and improve its fault diagnosis performance.

As the preprocessing in OFM, Hilbert transform is performed to obtain the instantaneous amplitude and frequency of motor stator current signals in all observed fault categories (Fault A, Fault B, Fault C). Then, the instantaneous amplitude and frequency are normalized, respectively, ranging from zero to one. This normalization corresponds to the scaling of the operating condition (i.e., the torque and speed levels) and thereby leads to removing the influence of operating conditions; thus, the few-shot learning model can diagnose motor faults even under new operating conditions. In addition, depthwise separable convolution is performed to extract fault-induced modulation features from the pairwise scaled instantaneous amplitude and frequency of the motor stator current signal.

In FDM, on the other hand, prototype-assisted dual-contrastive learning minimizes similarity distance loss of both the instance-wise and prototype-instance pairs composed of the extracted fault-induced modulation features. In the meta-train stage, instance block (IB) learns low-level, instance-wise similarity features between support and query sets in the observed categories (Fault A, Fault B, and Fault C). The instance features being learned from the IB are transferred to prototype block (PB). The PB learns both local- and global-similarity features by optimizing the distance between the prototype embeddings made by the PB and the instance embeddings transferred from the IB. In the meta-test stage, by using the extracted similarity features, the trained feature extractor \( \tilde{F}_\theta \) from the PB measures the similarity distance between support set (labeled data) and query set (unlabeled data) including the unseen fault category (Fault D), and finally the category of the prototype which is the most similar to the query sample is identified.

The architecture of PCDC is described in Figure 4 and Table 2. The following subsections will provide the computation procedure of OFM and FDM in PCDC.
(a) Operation-robust fault feature extraction module (OFM)

* IA: Instantaneous amplitude, IP: Instantaneous phase
Figure 3. The concept of the prototype-assisted dual-contrastive learning with depthwise separable convolutional neural network (PCDC): (a) Operation-robust fault feature extraction module (OFM) and (b) Few-shot diagnosis module (FDM)
3.2.1. Operation-robust fault feature extraction module

3.2.1.1. Preprocessing

Motor faults generally induce amplitude modulation (AM) and frequency modulation (FM) in the motor stator current signal (Chen & Feng, 2020). Under a motor fault (i.e., dynamic eccentricity) in steady-state conditions, the amplitude modulated stator current signal $x_{AM}(t)$ or phase modulated stator current signal $x_{PM}(t)$ can be expressed with the fault characteristic frequency $f_c$ (Blodt et al., 2006; Blodt et al., 2008).

$$x_{AM}(t) = I[1 + \alpha \cos(2\pi f_s t)] \cos(2\pi f_c t)$$

$$x_{PM}(t) = x_{st}(t) + x_{rt}(t) = I_{st} \sin(2\pi f_s t) + I_{rt} \sin(2\pi f_s t + \beta \cos(2\pi f_c t))$$

where $I$ is the amplitude of the stator current fundamental components; $x_{st}(t)$ and $x_{rt}(t)$ are the stator current components from the stator and rotor magnetic field with amplitude $I_{st}$ and $I_{rt}$, $\alpha$ and $\beta$ are modulation indices which are proportional to a degree of the faults, $f_s$ and $f_r$ are the stator supply frequency, the rotational frequency. To extract the fault-induced modulation terms, Hilbert transform $H$ is used to convert the single-phase stator current signal $x(t)$ into the analytic signal $z(t)$. The discrete form of the analytic signal is as follows:

$$z[t] = x[t] + jH\{x[t]\} = a[t]e^{j\varphi[t]}$$

$$f_{IF}[t] = \frac{1}{2\pi} \frac{d}{dt} \varphi[t] = f_s - \beta f_c \sin(2\pi f_c t)$$

where $a[t]$ and $\varphi[t]$ are instantaneous amplitude (IA) and phase (IP) of the motor stator current, respectively, and $f_{IF}$ is instantaneous frequency (IF). Therefore, IA and IF of the stator current signals are influenced by fault-induced modulation terms in equations (14) and (15) which indicates IA and IF contain fault-related information. For simple notation, we denote IF as $b[t]$ from now on.

In addition, since IA and IF provide information on load torque and rotating speed levels, respectively, changes in operating conditions (i.e., the torque and speed levels) greatly impact the motor stator current signal as well. Therefore, it can be inferred that using IA and IF as deep learning model inputs makes the model directly learn fault-related features so that the model performs well in fault diagnosis (Park et al., ...
In order to help the model converge faster for a given learning rate and generalize well for use with new load torque and rotating speed conditions, min-max normalization technique is used for pairwise input \(a[t]\) and \(b[t]\) as a scaling method.

\[
a'[t] = \frac{a[t] - \min \{a[t]\}}{\max \{a[t]\} - \min \{a[t]\}} \tag{18}
\]

\[
b'[t] = \frac{b[t] - \min \{b[t]\}}{\max \{b[t]\} - \min \{b[t]\}} \tag{19}
\]

It is worth pointing out that even if operating conditions of motors change, the few-shot learning model can diagnose motor current signals by scaling load torque and rotating speed, respectively. It indicates that the influence of operating conditions is removed, and thereby the similarity features learned from specific operating conditions can be also used in other operating conditions. For simple notations, we denote pairwise scaled input \(a'(t)\) and \(b'(t)\) as \(a(t)\) and \(b(t)\) from now on. To construct the deep learning model inputs, pairwise scaled inputs \(a[t]\) and \(b[t]\) are converted into a temporary support set and a query set with \(N\) classes:

\[
S_{\text{src}} = \left\{ \left( a_{it}^s, b_{it}^s, y_{it}^s \right) \right\}_{i=1}^{N \times K}
\]

and query set

\[
Q_{\text{src}} = \left\{ \left( a_{iq}^o, b_{iq}^o, y_{iq}^o \right) \right\}_{i=1}^{N \times K}.
\]

### 3.2.1.2. Depthwise separable convolution for feature extraction

The feature extractor is developed to learn unique fault-induced modulation features in the IA and IF of the motor stator current signal and to map the two unique features into a metric space. The feature extractor consists of two blocks: an instance block (IB) and a prototype block (PB) with different parameter \(\phi\) and \(\theta\). Both IB and PB contain depthwise separable feature extractor \(h(x)\) and metric embedding function \(m(x)\).

The depthwise separable feature extractor performs the convolution operation independently in the spatial and channel directions; thus, it can reduce model parameters, while forming the feature map with a shape equal to the standard convolution operation result (Chollet, 2017; Guo et al., 2019). Therefore, it can be inferred that the depthwise separable feature extractor is suitable for the FSL algorithm, as it is advantageous to use fewer model parameters in a limited-data condition. The metric embedding function is used to map the extracted features into the metric space and forms the prototype- and instance-embedding vectors.

First, depthwise separable convolution is performed on the pairwise input data \(a[t]\) and \(b[t]\). For spatial direction convolution, a series of convolution blocks \(G\) is used to extract unique fault-induced modulation features in the instantaneous amplitude and
frequency of the motor stator current signal, by using the convolution block parameters, \( W_1 \) and \( W_2 \), respectively. The convolution blocks contain a one-dimensional convolution layer \( f_{\text{conv}} \), a batch normalization layer \( f_{\text{bn}} \), and an activation function \( f_{\text{relu}} \). The output of the \( n \)th convolution blocks \( a_n^a \) and \( b_n^b \) are expressed, respectively, as:

\[
a_n^a = G_{W_1} (a_{n-1}^a) = f_{\text{relu}} [ f_{\text{bn}} ( f_{\text{conv}} (a_{n-1}^a) ) ] \quad (a_n^a, \in \mathbb{R}^{C^d})
\]

\[
b_n^b = G_{W_2} (b_{n-1}^b) = f_{\text{relu}} [ f_{\text{bn}} ( f_{\text{conv}} (b_{n-1}^b) ) ] \quad (b_n^b, \in \mathbb{R}^{C^d})
\]

where \( n \in [1, 2], a_0^a = a, b_0^b = b \). Let \( C \) and \( I \) denote the channel dimension and spatial length of the input data, respectively. For the channel direction convolution, a \( 1 \times 1 \) convolution layer \( f_{1 \times 1\text{conv}} \) is used to extract the cross-channel correlation features using the output of the spatial direction convolution.

As the final output of the depthwise separable convolution, which contains two convolution blocks as described in Figure 4, is expressed as:

\[
h(a_n^a, b_n^b) = f_{1 \times \text{conv}} (\text{cat}(a_n^{(2)}(2), b_n^{(2)}(2)))
\]

where \( \text{cat} \) denotes concatenation in the channel direction.

The metric embedding function aims to map equation (22) into the metric space. The metric embedding function contains a global average pooling layer (GAP) \( f_{\text{GAP}} \) and a linear layer \( f_{\text{linear}} \). The GAP layer applies average pooling over all spatial dimensions, and the linear layer maps the global-averaged feature into the metric space. For PB, the metric embedding function includes an additional prototype calculation process identical to equation (8). Finally, IB outputs instance support set \( S_\phi \) and instance query set \( Q_\phi \), as shown in equations (23) and (24). PB outputs the prototype support set of \( c \)-th class, \( P^c_\phi \), as in equation (25).

\[
S_\phi = F_\phi (x_\phi^a) = m_{IB} (h_{IB}(a_n^a, b_n^b))
\]

\[
Q_\phi = F_\phi (x_\phi^q) = m_{IB} (h_{IB}(a_n^a, b_n^b))
\]

\[
P^c_\phi = \frac{1}{|S^c_\phi|} \sum_{(x_\phi^a, y_\phi^a) \in S^c_\phi} F_\phi (x_\phi^a) = \frac{1}{|S^c_\phi|} \sum_{(x_\phi^a, y_\phi^a) \in S^c_\phi} m_{PB} (h_{PB}(a_n^a, b_n^b))
\]
Figure 4. The architecture of prototype-assisted dual-contrastive learning with depthwise separable convolutional neural network (PCDC).
**Table 2. The parameter settings of the prototype-assisted dual-contrastive learning with depthwise separable convolutional neural network (PCDC)**

<table>
<thead>
<tr>
<th>Block</th>
<th>Sub-Module</th>
<th>Layer</th>
<th>Parameters [kernel size, channel, stride, padding]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype block</td>
<td>Depthwise separable feature extractor</td>
<td>1-D depthwise conv/BN/ReLU</td>
<td>2*[9, 8, 2, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>concatenate</td>
<td>2*[9, 8, 2, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1x1 channel-wise conv/BN/ReLU</td>
<td>[1, 32, 1, 0]</td>
</tr>
<tr>
<td></td>
<td>Metric embedding</td>
<td>Global average pooling</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fully connected layer</td>
<td>[-, 50, -, -]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prototype calculation</td>
<td>-</td>
</tr>
<tr>
<td>Instance block</td>
<td>Depthwise separable feature extractor</td>
<td>1-D depthwise conv/BN/ReLU</td>
<td>2*[9, 8, 2, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>concatenate</td>
<td>2*[9, 8, 2, 4]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1x1 channel-wise conv/BN/ReLU</td>
<td>[1, 32, 1, 0]</td>
</tr>
<tr>
<td></td>
<td>Metric embedding</td>
<td>Global average pooling</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fully connected layer</td>
<td>[-, 50, -, -]</td>
</tr>
</tbody>
</table>

### 3.2.2. Few-shot diagnosis module

#### 3.2.2.1. Prototype-assisted dual-contrastive learning

The prototype-assisted dual-contrastive learning is proposed in this study; it is implemented through the integration of two distinct contrastive learning approaches: the instance block (IB) proceeds instance-wise contrastive learning to learn the local similarity features, and the prototype block (PB) proceeds prototypes-instance contrastive learning to learn both local- and global-similarity features where the instance embeddings used in PB are transferred from IB. Conducting two different contrastive learning in separate spaces enhances the stability of the training process, compared to conduct them jointly in the same space which may cause conflicting objectives due to mismatch in distance scale between instance-wise and prototype-instance distances. First, the instance embeddings being learned from IB are transferred to PB. By using the transferred instance embeddings which contain instance-wise contrastive features, PB can learn local similarity features without additional learning parameter, which can alleviate the overfitting problem in few-shot learning. While optimizing the distance loss between prototypes and transferred instance embeddings in PB learning space, PB can learn prototype-instance contrastive features which contain both local- and global similarity features, and thereby the PB can increase instance-discrimination ability while alleviating the overfitting problem.

In the source-domain dataset $S_{src}$, any two different samples ($x_i$, $x_{iq}$) are positive sample pair ($x$, $x^+$) if $y_i = y_{iq}$. Otherwise, the two different samples are negative sample pair. For the prototype-assisted dual-contrastive feature learning, the positive sample
pairs $X^+ = (x, x')$ and the negative sample pairs $X^- = (x, x')$ are formulated. The positive and negative sample pairs are randomly sampled from the support dataset $S$ and query dataset $Q$ as denoted in (6) and (7).

For the instance block (IB), after obtaining the instance embeddings $S_\phi$ and $Q_\phi$ from the feature extractor $F_\phi$, Euclidean distance is used to measure the similarity between the instance embeddings as follows:

$$d(S_\phi - Q_\phi) = \|S_\phi - Q_\phi\|_2 = \|F_\phi(x) - F_\phi(x')\|_2$$  \hspace{1cm} (26)

The positive and negative contrastive loss $L_{IB}$ can be calculated, respectively, by squaring Euclidean distance between instance sample embeddings made by feature extractor $F_\phi$. For the positive sample pairs, the Euclidean distance between the sample embeddings is minimized to increase the similarity between the embeddings as in equation (27). In the contrary, the Euclidean distance between the negative sample embeddings is maximized with the margin distance $m$ to decrease the similarity between the features to separate them at a distance of $m$ as in equation (28).

$$L_{IB}^{(+)}(x, x') = \left(\|F_\phi(x) - F_\phi(x')\|_2\right)^2$$  \hspace{1cm} (27)

$$L_{IB}^{(-)}(x, x') = \left\{\max\left(0, m - \|F_\phi(x) - F_\phi(x')\|_2\right)\right\}^2$$  \hspace{1cm} (28)

Therefore, the total loss function $J_{IB}$ with respect to the network parameter $\phi$ for the single task ($N$-way $K$-shot) is expressed as:

$$J_{IB}(\phi) = \frac{1}{\frac{1}{2}N \times K} \sum_{i=1}^{\frac{1}{2}N \times K} \left( ||y_i^s = y_i^q||\|F_\phi(x_i) - F_\phi(x_i')\|_2^2 + ||y_i^s \neq y_i^q||\max\left(0, m - \|F_\phi(x_i) - F_\phi(x_i')\|_2\right)\right)$$  \hspace{1cm} (29)

where $N$ is the number of categories and $K$ is the number of samples per each category. $N \times K$ is batch size. Once the support label $y_i^s$ and corresponding support sample $x_i^s$ are randomly sampled from support dataset $S$, the query label $y_i^q$ and corresponding query sample $x_i^q$ from query dataset $Q$ are also selected depending on whether they are the positive or negative pair. In the proposed method, half number of batch size is allocated to the positive and negative pairs, respectively. Therefore, the positive instance-wise pairs can be written as $X^{(+)} = \left\{(x_i^s, x_i^q, y_i^s, y_i^q)\right\}_{i=1}^{\frac{1}{2}N \times K}$ where $y_i^s = y_i^q$ and the negative instance-wise pairs are $X^{(-)} = \left\{(x_i^s, x_i^q, y_i^s, y_i^q)\right\}_{i=1}^{\frac{1}{2}N \times K}$ where $y_i^s \neq y_i^q$. The instance-wise contrastive loss helps bring closer instances from the same category while separate
instances from different category by minimizing intra-class variation $|F_{\phi}(x_n) - F_{\phi}(x_q)|$ and increasing inter-class difference $|F_{\phi}(x_n) - F_{\phi}(x_q)|$, simultaneously. For the network optimization process, the derivative of the IB loss function when the distances between negative sample pairs are smaller than margin distance is derived by using chain rule in equation (30). The parameter $\phi$ is updated as in equation (31) where $\alpha$ is learning rate and a stochastic gradient descent method based on adaptive estimation (Adam) is used to train the model more efficiently (Kingma & Ba, 2014).

$\frac{\partial J_{\text{m}}(\phi)}{\partial \phi} = \frac{1}{N \times K} \sum_{n=1}^{N} \sum_{q=1}^{K} \left( F_{\phi}(x_n) - F_{\phi}(x_q) \right) \left( \frac{\partial F_{\phi}(x_n)}{\partial \phi} - \frac{\partial F_{\phi}(x_q)}{\partial \phi} \right) + \| y_n^k = y_q^k \| \times (-2) \left( F_{\phi}(x_n) - F_{\phi}(x_q) \right) \left( \frac{\partial F_{\phi}(x_n)}{\partial \phi} - \frac{\partial F_{\phi}(x_q)}{\partial \phi} \right)$ if $m > \| F_{\phi}(x_n) - F_{\phi}(x_q) \|$

(30)

$\phi \leftarrow \phi - \alpha \frac{\partial J_{\text{IB}}(\phi)}{\partial \phi}$

(31)

From the equations (30) and (31), the total loss $J_{\text{IB}}$ consists of instance embeddings, and the network parameter $\phi$ is updated by the multiplication of Euclidean distance between instance embeddings $F_{\phi}(x)$ and between their first derivative term $\frac{\partial F_{\phi}(x)}{\partial \phi}$. Therefore, the network only learns correlations between instance embeddings. Though the instance-wise contrastive learning performs well on the instance discrimination tasks by identifying even a subtle difference between samples, it only conserves the instance-wise correlations but neglects the global similarity structures in each fault class, which makes it vulnerable to noisy samples and may cause the overfitting; this in turn limits the model performance as further training gives limited information (Tschannen et al., 2019). Therefore, the PB is developed to overcome these disadvantages while maintaining the advantages of the instance-discrimination ability.

The prototype block (PB) formulates prototypes $P(x)$ from support dataset $S$, which represent global embeddings of the dataset. After transferring instance embeddings from the IB to PB, contrastive learning proceeds between the transferred instance embeddings and prototype embeddings formulated by the PB. The positive and negative contrastive loss $L_{\text{PB}}$ with respect to the network parameters $\theta$ and $\phi$ are obtained by squaring Euclidean distance between prototype embeddings and instance embeddings as follows:
\[
L_{PB}^{(i)}(P_a(x), x_{iq}^*, y_{iq}^*, y_{iq}^*) = \left( \frac{1}{\|S_{y} \|} \sum_{(x_a, y_a) \in S_{y}} F_\theta(x_a) - F_\phi(x_{iq}^*) \right)_2^2 \]  
(32)

\[
L_{PB}^{(i)}(P_a(x), x_{iq}, y_{iq}, y_{iq}) = \left( \max \left(0, m - \frac{1}{\|S_{y} \|} \sum_{(x_a, y_a) \in S_{y}} F_\theta(x_a) - F_\phi(x_{iq}) \right)_2 \right)^2 \]  
(33)

\[
S_y \text{ indicates support dataset corresponding to the label } y \text{ as denoted in (6). Here, the instance embedding } F_\phi(x) \text{ from IB are used as negative or positive feature pairs with respect to the prototype embedding } (1/|S_y|) \sum F_\theta(x) \text{ made by PB. From the perspective of feature extractor } F_\theta \text{ of PB, the transferred instance embedding } F_\phi(x) \text{ act as a prior knowledge of local similarity as the transferred instance embeddings contain instance-wise contrastive features. Therefore, by transferring the instance embedding } F_\phi(x) \text{ being learn from IB to the PB contrastive learning space, PB can learn local similarity features without using additional network parameters which can alleviate the overfitting, the problem of instance-wise contrastive learning. Though } F_\theta \text{ and } F_\phi \text{ contain different model parameters, as they have the same neural network structure as described in Figure 4, the input data are located in the same dimensional embedding space which makes distance calculation between } (1/|S_y|) \sum F_\theta(x) \text{ and } F_\phi(x) \text{ possible. The total loss function made by PB with respect to the network parameter } \theta \text{ and } \phi \text{ for the single episode (N-way K-shot) is expressed as:}
\]

\[
J_{PB}(\theta, \phi) = \frac{1}{N \times K} \sum_{i=1}^{N \times K} \left[ \sum_{y_{iq}^* = y_{iq}} \frac{1}{\|S_{y} \|} \sum_{(x_a, y_a) \in S_{y}} F_\theta(x_a) - F_\phi(x_{iq}^*) \right)_2^2
\]
\[
+ \sum_{y_{iq}^* \neq y_{iq}} \max \left(0, m - \frac{1}{\|S_{y} \|} \sum_{(x_a, y_a) \in S_{y}} F_\theta(x_a) - F_\phi(x_{iq}^*) \right)_2 \right)^2 \]  
(34)

where \( N \) is the number of categories and \( K \) is the number of samples per each category. \( N \times K \) is batch size. According to the labels of the transferred query embeddings \( F_\phi(x_{iq}) \), the support labels \( y_{is} \) and corresponding prototype \( (1/|S_y|) \sum F_\theta(x) \) are determined from support dataset \( S \), depending on whether they are the positive or negative pair. Half number of batch size is allocated to the positive and negative pairs, respectively. Therefore, the positive prototype-instance pairs can be written as

\[
XP^{(+)} = \left\{ \left( F_\phi(x), F_\phi(x_{iq}), y_{is}, y_{iq} \right) \right\}_{i=1}^{N \times K} \]  
where \( y_{is} = y_{iq} \) and the negative prototype-instance
pairs are $X P^{(-)} = \{(P^y_\theta(x), F_\phi(x_i), y_{i_0}, y_{i_0})\}_{i_0,i_0}^{N \times K}$ where $y_{i_0} \neq y_{i_0}$. $P^y_\theta(x)$ refers to the prototype of $y$-th class calculated from the support set $S$. For the network optimization process, the derivative of the PB loss function when the distance between negative sample pairs are smaller than margin distance is derived by using chain rule in equation (35). The parameter $\theta$ and $\phi$ are updated as in equation (36).

$$\frac{\partial J_{m_\theta}(\theta, \phi)}{\partial \theta} = \frac{1}{N \times K} \sum_{n=1}^{N \times K} \left[ (y_{i_0}^{(1)} = y_{i_0}^{(2)}) \times 2 \left( \frac{1}{|S_m|} \sum_{(x_{i_0}, y_{i_0}) \in S_m} F_\phi(x_{i_0}) - F_\phi(x_{i_0}^+) \right) \frac{\partial}{\partial \theta} \left( \frac{1}{|S_m|} \sum_{(x_{i_0}, y_{i_0}) \in S_m} F_\phi(x_{i_0}) \right) \right] + \left[ (y_{i_0}^{(1)} \neq y_{i_0}^{(2)}) \times (-2) \left( \frac{1}{|S_m|} \sum_{(x_{i_0}, y_{i_0}) \in S_m} F_\phi(x_{i_0}) - F_\phi(x_{i_0}^+) \right) \frac{\partial}{\partial \theta} \left( \frac{1}{|S_m|} \sum_{(x_{i_0}, y_{i_0}) \in S_m} F_\phi(x_{i_0}) \right) \right],$$

(35)

if $m > \left\| \frac{1}{|S_m|} \sum_{(x_{i_0}, y_{i_0}) \in S_m} F_\phi(x_{i_0}) - F_\phi(x_{i_0}^+) \right\|_2^2$

$$\theta \leftarrow \theta - \alpha \frac{\partial J_{m_\theta}(\theta, \phi)}{\partial \theta}$$

(36)

The network parameter $\theta$ is updated by the multiplication of Euclidean distance between the prototypes and instance embeddings, and the first derivative of the prototypes. As the embedding position of the prototypes $(1/|S_m|) \sum F_\theta(x)$ of PB are updated depending on the distance from the transferred instance embedding $F_\phi(x)$ from IB, feature extractor $F_\theta$ of the PB learn prototype-instance contrastive features which contain both local- and global similarity features while optimizing the distance loss between the prototypes and transferred instance embeddings. Local similarity features are extracted from the instance embeddings transferred from the IB and global similarity features are extracted from the prototypes while contrastive learning proceeds between prototypes and instance embeddings. Also, when minimizing the loss function by applying the gradient descent method, the parameter $\theta$ is updated by averaged embeddings (prototypes) rather than single embeddings, which mitigates the gradient going in the wrong direction due to the noisy characteristics of individual sample, unlike instance-wise contrastive loss (30) where the network parameter $\phi$ is updated only by instance-wise embeddings. The slight distance difference between instance embeddings may affects the minimization of the loss function. To summarize, the PB network parameter $\theta$ is updated by the combination of global behavior of the prototypes and local behavior of transferred instances embeddings, which makes the PB to discriminate well the data by each category while alleviate the overfitting problem. On the other hand, for the comparison with loss function of Proto Net as described in (10), Proto Net mainly focuses on increasing the distance for different fault clusters and does not directly consider the distance between individual samples. Therefore, Proto Net may
have a large variation between samples in the same category, which may decrease fault diagnosis performance. Though Proto Net captures global similarity features of each category well, it has difficulties of clustering samples within the same category with low variation. In case of the proposed method, it can separate the data by each category with low intra-class variation by minimizing intra-class variation while maximizing inter-class difference through contrastive learning of prototypes and instance embeddings.

Finally, the total loss function of the prototype-assisted dual-contrastive learning with respect to the feature extractors, and objective function are expressed as follows:

$$ J_{total}(F_{\phi}(x^S),F_{\phi}(x^O),P_{\phi}(x^S)) = J_{in}(F_{\phi}(x^S),F_{\phi}(x^O)) + J_{pn}(P_{\phi}(x^S),F_{\phi}(x^O)) + \lambda \sum \beta $$

$$ \theta^*,\phi^* = \arg\min_{\theta,\phi} J_{total}(F_{\phi}(x^S),F_{\phi}(x^O),P_{\phi}(x^S)) \quad (38) $$

where $P(x^S)$ indicates prototypes which calculated by support dataset and $\lambda \sum \beta$ is L2 regularization term, and the $\beta$ is any weight in IB and PB. Lambda is the L2 regularization rates for $\beta$. The IB parameter $\phi$ is optimized to transfer optimal instance embeddings to the PB. The final goal of objective function is to find optimal parameter $\theta^*$ which contains both local- and global similarity features based on the prototypes and transferred instance embeddings. The pseudo code for optimization process are provided in Algorithm 1.

Algorithm 1. The optimization process of the proposed method for one epoch

**Input**: The source-domain dataset $S_{src}$, the feature extractor $F_{\phi}$ and $F_{\theta}$, learning rate $\alpha$, batch size with $N$-way $K$-shot, episode number $e$

**Initialize**: The weight parameters $\phi$ and $\theta$ of the feature extractors

**Output**: The trained model parameter $\phi$ and $\theta$

Randomly initialize the network parameters $\phi$ and $\theta$

for $i=1$ to episode number do

############### local similarity feature learning process ###############

Randomly sample the positive sample pairs and negative sample pairs from support dataset $S$ with $N$-way $K$-shot


Calculate the instance block (IB) instance embeddings $F_{\phi}$ by (23) and (24)

Calculate the Euclidean distance $d(F_{\phi}(x), F_{\phi}(x_{iq}))$ for positive and negative sample pairs by (27) and (28)

Calculate the instance-wise loss function $J_{ib}$ by (29)

 local and global similarity feature learning process 

Calculate the prototype block (PB) prototype $P_{\theta}(x)$ by (25)

Transfer the instance embeddings $F_{\phi}$ to the PB feature spaces

Make prototype-instance pairs from support dataset $S$ and transferred instance embeddings with $N$-way $K$-shot

$$XP^{(+)} = \left\{ (P_{\theta}^{+}(x), F_{\phi}(x_{iq}), y_{ia}, y_{iq}) \right\}_{i_{ia}, i_{iq}=1}^{N \times K} , y_{ia} = y_{iq} \text{ and }$$

$$XP^{(-)} = \left\{ (P_{\theta}^{-}(x), F_{\phi}(x_{iq}), y_{ia}, y_{iq}) \right\}_{i_{ia}, i_{iq}=1}^{N \times K} \text{ where } y_{ia} \neq y_{iq}$$

Calculate the Euclidean distance $d(P_{\theta}(x), F_{\phi}(x_{iq}))$ for positive and negative sample pairs by (32) and (33)

Calculate the prototype-instance loss function $J_{pb}$ by (29)

Update IB and PB parameter:

$$\phi \leftarrow \phi - \alpha \frac{\partial J_{ib}(\phi)}{\partial \phi}, \theta \leftarrow \theta - \alpha \frac{\partial J_{pb}(\theta, \phi)}{\partial \theta}$$

end

### 3.2.2.2. Fault diagnosis including unseen fault categories

The target-domain dataset is used for fault diagnosis including unseen fault categories. The target-domain dataset is obtained from a new operating condition and contains unseen fault categories. In the proposed method, the target-domain dataset consists of support set $S_{tar} = \left\{ (x_{i}^{S}, y_{i}^{S}) \right\}_{i=1}^{(N+u) \times K}$ and query set $Q_{tar} = \left\{ (x_{i}^{Q}, y_{i}^{Q}) \right\}_{i=1}^{(N+u) \times K}$ where the number of the observed categories is $N$ and the number of unseen categories is $u$. $K$ is the number of samples per categories. Then, the goal of few-shot fault
diagnosis is to predict the query label $y^Q$ in the query set $Q_{\text{tar}}$ even though certain fault categories were not observed in the model training process. In order to evaluate the classification performance of the PCDC, trained feature extractor $F_{\theta}$ of the PB measures the similarity distance between the prototype $P_{\theta}(x^*)$ and instance query set $F_{\theta}(x^*)$ and selects the prototype most close to query sample. The class of query sample $\hat{y}_{Q}^{\text{Q}}$ can be predicted as outlined in equation (39), where $P_{\theta}^{*}(x^*)$ is $c$-th prototype made by feature extractor $F_{\theta}$ of the PB and $d$ is the Euclidian distance.

$$\hat{y}_{Q}^{\text{Q}} = \arg \max_c \frac{\exp \left( -d \left( F_{\theta}(x_{iq}^*), P_{\theta}^{*}(x^*) \right) \right)}{\sum_{c=1}^{N+U} \exp \left( -d \left( F_{\theta}(x_{iq}^*) P_{\theta}^{*}(x^*) \right) \right)}$$ (39)

4. Experimental Validation

The proposed method is validated by applying it to two different case studies. It is assumed that both the source and target domains contain a limited number of labeled samples, which recreates the challenges of a real-world industrial environment. The first case study focuses on evaluating the influence of diverse working condition, while the second case study focuses on evaluating the influence of the number of unseen categories.

4.1. Few-shot settings for motor fault diagnosis

Considering the experimental settings commonly used in few-shot fault diagnosis problems and samples of the meta-datasets used in experiments, this study constructed 7-way 5-shot tasks and 5-way 5-shot tasks with five query samples per categories for two different experiments. For the meta-train dataset, the total number of normal labeled samples was set as 100 and each fault labeled sample as 20 samples to reflect a real industrial environment, where the normal condition data was relatively larger in quantity than the fault condition data. For the model test, the meta-test dataset obtained from different operating conditions is used to evaluate the model performance. The total number of the normal and each fault samples was set as 100 for evaluation. In the meta-test stage, query samples per categories were set as 95 to investigate the classification performance more precisely. The window length for per current signal sample is 2048 current data points. We used a grid search algorithm to select hyperparameters related to model training, such as a learning rate and learning rate decay. The values of the learning rate and learning rate decay were chosen from \{0.04, 0.004, 0.0004, 0.00004\} and \{0.005, 0.0005, 0.00005, 0.000005\}, respectively. Based on the highest average diagnosis performance, the optimum values of learning rate and learning rate decay were set as 0.0004 and 0.00005. An early stopping technique was
used to mitigate the risk of overfitting by stopping training when the validation loss increased during a certain epoch. The early stopping criterion was defined with a threshold of 10 epochs, considering that the validation loss failed to converge when there was no improvement for more than 10 consecutive epochs compared to the best validation loss. The maximum number of training epochs as 100, and the learning tasks per epoch (episodes) as 30. The margin value $\varepsilon$ was set as two empirically. The number of unseen fault categories $u$ was set as one for the case study 1 and both one and two for the case study 2. The hyper parameters and few-shot settings are summarized in Table 3.

<table>
<thead>
<tr>
<th>Hyper parameter</th>
<th>Values</th>
<th>Few-shot settings</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance metric</td>
<td>Euclidian distance</td>
<td>Labeled samples used in meta-train stage</td>
<td>100 (Normal) 20 (Each fault class)</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.0004</td>
<td>Labeled samples used in meta-test stage</td>
<td>100 (Normal) 100 (Each fault class)</td>
</tr>
<tr>
<td>Learning rate decay</td>
<td>0.00005</td>
<td>Support samples per class ($K$-shot)</td>
<td>5 (Meta-train/test stage)</td>
</tr>
<tr>
<td>Maximum epoch</td>
<td>100</td>
<td>Query samples per class ($K$-shot)</td>
<td>5 (Meta-train stage) 95 (Meta-test stage)</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>Number of unseen fault categories</td>
<td>1 (Case study 1) 1 or 2 (Case study 2)</td>
</tr>
<tr>
<td>Early stop duration epoch</td>
<td>10</td>
<td>Tasks per epoch</td>
<td>30</td>
</tr>
</tbody>
</table>

The comparison models include representative meta-learning models and a traditional parameter-transfer learning model which are frequently used in other few-shot fault diagnosis papers. Prototypical network (Proto Net) (Snell et al., 2017), Siamese network (Siamese Net) (Zhou et al., 2020), and Relation network (Relation Net) (Sung et al., 2018) are used for the meta-learning comparison models. CNN-based fine-tuned network (Fine-tuned Net) is used as the traditional parameter-transfer learning model. The same number of convolution blocks as PCDC were used in comparison models and the experiments were performed under same environments for the fair comparison. Proto Net with OFM (Proto Net w/ OFM) is used as the comparison model to investigate the effect of OFM and prototypical-assisted dual-contrastive learning on the fault diagnosis performance, respectively. Since prototype-assisted dual-contrastive learning cannot be used alone without OFM, only Proto Net w/ OFM is used as an ablation model. Table 4 summarizes the comparative setting for the ablation study.
Table 4. The comparative setting for ablation study

<table>
<thead>
<tr>
<th>Model</th>
<th>Operation-robust fault feature extraction module</th>
<th>Prototype-assisted dual-contrastive learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCDC</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Proto Net w/ OFM</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Proto Net</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

4.2. Case study 1: Interior permanent magnet synchronous motor (IPMSM) dataset

4.2.1. Experimental setup and dataset description

Figure 5 shows the IPMSM testbed used in the first case study. The testbed consists of a target motor, a load motor, a torque meter, an encoder, and a data acquisition (DAQ) system. The FMAIN22-IWAL IPMSM made by Higenmotor was used for the target motor, controlled by the E40H12-2048 encoder and the A1000 Inverter from Yaskawa. The SIMOTICS M-1PH8 induction motor by Siemens was used for the load motor. The specifications of the IPMSM are shown in Table 5. The single-phase current data was obtained by the DAQ system with a sampling rate of 12,800 Hz. The DAQ system contains current probes from Tektroniks A622, Siemens SCADAS Mobile, and Siemens Testlab.

Figure 5. Experimental setup for the interior permanent magnet synchronous motor testbed.

Six different types of faulty motor specimens and one normal motor specimen are used in the experiments. Seven different IPMSM health conditions are considered, including normal (Nor), static eccentricity (SE), bearing inner race fault (BIF), bearing
outer race fault (BOF), inter-turn short (SIS), dynamic eccentricity (DE), and demagnetization (DG); their descriptions are provided in Table 6. All current data by category were collected under four different operating conditions. The experimental operating conditions are expressed by speed and load torque, as illustrated in Table 7. The torque and speed values were selected as 100 and 50 % of the rated torque and speed of the test motor, respectively. Representative normal and fault current signals under operating conditions at 2250 RPM and 2.54 Nm are visualized in Figure 6. From the instantaneous amplitude (IA) of each current signal, distinct amplitude modulations can be observed across different fault categories. As described in Section 3.2.1.1, motor faults induce modulations in stator current signals based on the motor fault characteristics, and thereby distinct IA profiles are observed.

Figure 6. Normal and fault current signals under operating conditions at 2250 RPM and 2.54 Nm: (a) Normal, (b) Static eccentricity, (c) Bearing inner race fault, (d) Bearing outer race fault, (e) Inter-turn short, (f) Dynamic eccentricity, (g) Demagnetization.

Table 5. Specifications of the interior permanent magnet synchronous motor

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated speed</td>
<td>4500 RPM</td>
</tr>
<tr>
<td>Rated torque</td>
<td>4.7 Nm</td>
</tr>
<tr>
<td>Rated output</td>
<td>2.2 kW</td>
</tr>
<tr>
<td>Pole pairs</td>
<td>3</td>
</tr>
<tr>
<td>Number of slots</td>
<td>36</td>
</tr>
<tr>
<td>Moment of inertia</td>
<td>$17.18 \times 10^{-4}$ kg·m$^2$</td>
</tr>
</tbody>
</table>
Table 6. Description of the health conditions of the interior permanent magnet synchronous motor

<table>
<thead>
<tr>
<th>Class</th>
<th>Fault mode</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal</td>
<td>Nor</td>
</tr>
<tr>
<td>2</td>
<td>Static eccentricity (horizontal axis 4mm)</td>
<td>SE</td>
</tr>
<tr>
<td>3</td>
<td>Bearing inner race fault (1 spalls)</td>
<td>BIF</td>
</tr>
<tr>
<td>4</td>
<td>Bearing outer race fault (1 spalls)</td>
<td>BOF</td>
</tr>
<tr>
<td>5</td>
<td>Inter-turn short (variable resistance 5Ω)</td>
<td>SIS</td>
</tr>
<tr>
<td>6</td>
<td>Dynamic Eccentricity (2mm)</td>
<td>DE</td>
</tr>
<tr>
<td>7</td>
<td>Demagnetization</td>
<td>DG</td>
</tr>
</tbody>
</table>

Table 7. Experimental settings of the domain shift cases (unseen fault: demagnetization)

<table>
<thead>
<tr>
<th>Representation of the experimental operating conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
</tr>
<tr>
<td>2,250 RPM, 2.54 Nm</td>
</tr>
</tbody>
</table>

Incremental domain shift       Decremental domain shift

<table>
<thead>
<tr>
<th>No</th>
<th>Source domain</th>
<th>Target domain</th>
<th>No</th>
<th>Source domain</th>
<th>Target domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>a</td>
<td>b</td>
<td>V</td>
<td>d</td>
<td>c</td>
</tr>
<tr>
<td>II</td>
<td>a</td>
<td>c</td>
<td>VI</td>
<td>d</td>
<td>b</td>
</tr>
<tr>
<td>III</td>
<td>b</td>
<td>d</td>
<td>VII</td>
<td>c</td>
<td>a</td>
</tr>
<tr>
<td>IV</td>
<td>c</td>
<td>d</td>
<td>VIII</td>
<td>b</td>
<td>a</td>
</tr>
</tbody>
</table>

Fault diagnosis scenarios are divided into two types: incremental domain shift and decremental domain shift; these are illustrated, respectively, in Table 7. The incremental domain shift indicates the case where either the speed or load torque values increase as the source domain changes to the target domain. The decremental domain shift represents the opposite situation. Each domain shift scenario has four cases, as shown in Table 7. In this case study, the number of unseen faults is fixed as one when evaluating the target domain performance; DG is selected for the unseen fault kind in the experiments. Therefore, the source domain has six categories, which consist of one normal mode and five fault modes. The target domain contains seven categories, which consist of the same categories in the source domain and one unseen fault category.

4.2.2. Results and analysis

Figure 7 and Table 8 show the 10-iteration averaged accuracy of fault diagnosis for the target domain in both the incremental and decremental domain shift cases. It is worth pointing out that PCDC outperforms the other comparison models in all cases.
addition, the overall accuracy of PCDC is high consistently, regardless of the domain shift cases, even though the overall accuracy of most other comparison models decreases in the decremental domain shift cases compared to the incremental domain shift cases.

For the meta-learning comparison models, it can be seen that the models adapt to a new operating condition to some amount by using learned similarity features from the source domain dataset as shown in other papers (Y. Feng, J. Chen, T. Zhang, et al., 2022; Wang et al., 2021; Zhang et al., 2019). Within the meta-learning comparison models (Proto Net, Siamese Net, and Relation Net), Proto Net achieves the best performance in all cases. As compared with Proto Net, the results of Proto Net w/ OFM show average diagnosis performance improvements of 2 % and 9 % in the incremental and decremental domain shift cases, respectively, owing to the effect of OFM which removes the influence of the operating conditions in the motor current signals. Further, PCDC results show that prototype-assisted dual-contrastive learning can increase the additional performance by 2 % and 5 % compared to Proto Net w/ OFM by extracting more discriminative similarity features. Also, the standard deviation of the accuracy of PCDC is smaller than that of the other models. These results indicate that the proposed PCDC not only has superior generalization ability for the unseen target dataset but also retains great reliability in terms of performance.

It is also noticeable that all meta-learning models excluding Relation Net perform better than Fine-tuned Net, even though these models did not train on the target domain data. The accuracy of Relation Net is significantly decreased when the speed condition changes. It can be inferred that Relation Net overfits to the metric space of the source domain as Relation Net adopts a deep learning-based relation module for computing the similarities contrary to the other models.

To compare the time efficiency, the processing time of each model is measured. The processing time can be divided into data pre-processing time and mean training time, as described in Table 9. Data pre-processing time refers to the duration required for transforming raw data into suitable input shapes for a few-shot learning model. As PCDC and Proto Net w/ OFM contain additional pre-processing steps such as Hilbert transform and normalization, the data pre-processing time slightly increases from 1.2 seconds to 4.1 seconds. The mean training time represents the average duration for the loss function to converge during the model training. As a limited amount of data is used in model training, the mean training times of the models are much shorter than those observed in other conventional fault diagnosis research. The mean training time for PCDC is 23.6 seconds, which is 6 seconds longer than the mean training time of Fine-tune Net, which showed the fastest convergence speed. The deviation in total processing times among the models is approximately 10 seconds, which is not a significant time difference.
Table 8. Average fault diagnosis accuracy including unseen fault category in the target domain under 7-way 5-shot: (a) Incremental domain shift cases, (b) Decremental domain shift cases.

<table>
<thead>
<tr>
<th>Model</th>
<th>Average test accuracy ± STD (%) for 7-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a → b</td>
</tr>
<tr>
<td>PCDC (proposed)</td>
<td>98.6±1.5</td>
</tr>
<tr>
<td>Proto Net w/ OFM</td>
<td>95.9±0.55</td>
</tr>
<tr>
<td>Proto Net</td>
<td>93.6±1.52</td>
</tr>
<tr>
<td>Siamese Net</td>
<td>81.2±9.52</td>
</tr>
<tr>
<td>Relation Net</td>
<td>65.0±15.02</td>
</tr>
<tr>
<td>Fine-tuned Net</td>
<td>67.7±10.56</td>
</tr>
</tbody>
</table>

(a) and (b)
In order to investigate the distribution of the support and query sets embeddings in the source and target domains, principal component analysis (PCA) was used for both PCDC and Proto Net approaches. As PCA projects each embedding onto the first few principal components while retaining the data’s variation, the PCA plot is used to visualize the variation of the dimension-reduced embeddings. Figure 8 shows PCA results of prototype $P=[P_1,\ldots,P_N]$ and corresponding query set $q=[q_1,\ldots,q_{N\times K}]$ where $N$ is the number of categories and $K$ is the number of samples in each category in a given source or target domain. The star marks indicate prototype embeddings and the circle marks indicate single-query sample embeddings in five-shot settings. The source domain shows the feature space of the model when the model is trained on the train dataset which contains only the observed categories, while target domain shows the feature space of the model when the model is tested on the test dataset which contains...
both the observed- and unseen-category. The results are described for incremental domain shift case (Case III) and decremental domain shift case (Case VI). The range of the principal axis means the range of the source or target data distribution.

In case of Proto Net, the range of the first or second principal axis changes as the rotating speed changes, which indicates the feature space is influenced by the changes of the operating conditions as described in Figure 8a and Figure 8b. Also, the embeddings are aligned along the first principal axis by category and the variance of the embedding along the second principal axis is much smaller than that of the first principal axis. It can be inferred that the diversity of the embeddings found by Proto Net can be almost explained by the variance along the first principal axis. It can be seen that the embeddings in the Proto Net results significantly overlap with other categories in the target domain, especially in the decremental domain shift cases, as illustrated in Figure 8b. Since there is not sufficient feature to clearly classify the embeddings by categories in the target domain, the embeddings are aligned and classified only through the first principal axis.

However, in case of PCDC, as the influence of the operating conditions have been removed from the motor current signals by OFM, the range of the first and second principal axis remains similar even if the domain shifts. It indicates that the operation-robust fault features are extracted and the features learned from source domain dataset can be well used in target domain dataset. It reinforces generalization of PCDC under new operating conditions as described in Figure 8c and Figure 8d. Further, it can be observed that the embeddings in both the source and target domains are well-clustered by each category, with two independent principal axes, even when the rotating speed decreases. It can be inferred that PCDC extract sufficient discriminative features without causing the overfitting problem by learning local- and global similarity features from the small dataset. Therefore, PCDC can generalize well for use with different operating conditions, regardless of the incremental or decremental domain shift cases.
(a) Proto Net in Case III. (2,250 RPM, 4.5 Nm) to (4,500 RPM, 4.5 Nm)

(b) Proto Net in Case VI. (4,500 RPM, 4.5 Nm) to (2,250 RPM, 4.5 Nm)
To further observe the target domain test performance of the models with a large number of test samples, the t-distributed stochastic neighbor embedding (t-SNE) approach was also used for analysis. t-SNE is a non-linear dimensionality reduction algorithm that can be used to visualize the high-dimensional data into a low-dimensional plane, and thereby examines whether the data are well clustered (Van der Maaten & Hinton, 2008). Figure 9 shows the t-SNE results of the decremental domain shift case (Case VI). For Proto Net, Siamese Net, and Fine-tuned Net, the extracted features commonly mixed among Nor, SIS, and DE and among SE, BIF, and BOF. For Relation Net, all embeddings overlap each other and are not properly classified by
category. Proto Net w/ OFM extracts better features than the meta-learning comparison models, which indicates that by removing the influence of the operating conditions, similarity features learned from the source domain are well used to diagnose the target domain dataset. However, the distributions of features corresponding to Nor and DE and corresponding to SE and BIF are slightly mixed. On the other hand, it is clearly observed that PCDC can identify seven different categories precisely and the features are well-separated by each category at equal distances. In addition to the effect of OFM, PCDC can extract more discriminative features by learning local- and global-similarity features from the dataset. The features are well-clustered in a circle with low variance, which shows that PCDC maximizes inter-class variation while minimizes intra-class differences by prototype-assisted dual-contrastive learning. Therefore, it can be explained that PCDC extracts the best generalized features for target domain using OFM and prototype-assisted dual-contrastive learning.

Figure 9. t-SNE visualization of features extracted by the various models for Case VI under 7-way 5-shot settings: (a) PCDC (proposed), (b) Proto Net w/ OFM, (c) Proto Net, (d) Siamese Net, (e) Fine-tuned Net, (f) Relation Net.
4.3. Case study 2: Surface-mounted permanent magnet synchronous motor (SPMSM) datasets

4.3.1. Experimental setup and dataset description

The SPMSM testbed was set up for the second case study as shown in Figure 10. The target SPMSM (200 W, 3000 rpm, 5.5 Nm, 10 pole pairs) manufactured by TM TECH was position-controlled with a Welcon servo driver system. A BHB-3BA hysteresis brake manufactured by Magtrol was connected to the target motor shaft with couplings to exert torque. The single-phase motor current signal was obtained with a sampling rate of 12,800 Hz by a data acquisition (DAQ) system that contains an NI-9870 measurement module by National Instruments and a current probe (Tektroniks, A622).

![Figure 10. Experimental setup for the surface-mounted permanent magnet synchronous motor testbed.](image)

In contrast to the IPMSM dataset, which consists of different fault categories only, the SPMSM dataset contains both different severity categories as well as fault categories. Two different fault modes were tested: stator inter-turn short (SIS) and misalignment (MIS). The SIS specimens were made by coiling the uncovered windings in the motor stator. A weak-severity specimen (SIS1) and a severe-severity specimen (SIS2) were made by decreasing the resistance by 2 % and 6 %, respectively. The MIS specimens were made by rearranging the vertical height of the motor. The weak-severity specimen (MIS1) and severe-severity specimen (MIS2) were set by the amount of height variation, specifically, 2 mm and 4 mm, respectively. Therefore, the SPMSM dataset has a total of five categories, as described in Table 10. Representative normal and fault current signals under operating conditions at 3000 RPM and 3.8 Nm are visualized in Figure 11. From the instantaneous amplitude (IA) of each current signal, distinct amplitude modulations can be observed across different fault categories. As
motor faults induce modulations in stator current signals based on the motor fault characteristics, distinct IA profiles are observed.

Figure 11. Normal and fault current signals under operating conditions at 3000 RPM and 3.8 Nm: (a) Normal, (b) SIS1, (c) SIS2, (d) MIS1, (e) MIS2.

In terms of unseen faults, the fault diagnosis scenarios are divided into two types: cases with one unseen fault (1UF) and cases with two unseen faults (2UF). Table 10 also shows the class distribution for the source and target domains. Cases I to IV represent the cases where one unseen fault category appears in the target domain, while Cases V to VII refer to the cases where two unseen fault categories appear in the target domain. The unseen fault data was not participated in the train stage, but only appear in the test stage and the similarity with other data is measured by the trained model. The operating condition was set to be the same for all experimental cases for the source domain (3000 rpm, 3.8 Nm) and the target domain (3000 rpm, 5.5 Nm), respectively.
Table 10. Experimental settings of fault diagnosis scenario

<table>
<thead>
<tr>
<th>Class No</th>
<th>Fault mode</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal</td>
<td>Nor</td>
</tr>
<tr>
<td>2</td>
<td>Stator inter-turn short (weak-severity: resistance reduction of 2 %)</td>
<td>SIS1</td>
</tr>
<tr>
<td>3</td>
<td>Stator inter-turn short (severe-severity: resistance reduction of 6 %)</td>
<td>SIS2</td>
</tr>
<tr>
<td>4</td>
<td>Misalignment (weak-severity: height variation of 2 mm)</td>
<td>MIS1</td>
</tr>
<tr>
<td>5</td>
<td>Misalignment (severe-severity: height variation of 4 mm)</td>
<td>MIS2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>One unseen fault case (1UF)</th>
<th>Two unseen faults case (2UF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Source domain</td>
</tr>
<tr>
<td>Ⅰ</td>
<td>1, 3, 4, 5</td>
</tr>
<tr>
<td>Ⅱ</td>
<td>1, 2, 4, 5</td>
</tr>
<tr>
<td>Ⅲ</td>
<td>1, 2, 3, 5</td>
</tr>
<tr>
<td>Ⅳ</td>
<td>1, 2, 3, 4</td>
</tr>
</tbody>
</table>

4.3.2. Results and analysis

Figure 12 and Table 11 show the 10-iteration average few-shot fault diagnosis performance for both 1UF cases and 2UF cases in the target domain. PCDC outperforms the other models regardless of the unseen fault scenario. In addition, the fault diagnosis performance of PCDC is more reliable as its accuracy variation is smaller than that of the other models. For the comparison models (Proto Net, Siamese Net, Relation Net, and Fine-tuned Net), the average accuracy is limited to under 90 % in all fault scenarios. We found that the average accuracy in 2UF is lower than 1UF for the comparison models. In the case of Proto Net, the accuracy decreases about 10 % in Cases V and VI, compared to the 1UF cases. It can be inferred that Cases V and VI are relatively challenging environments, as they require the model to classify both an unseen fault and unseen severity simultaneously. Proto Net achieves the best performance with a lower standard deviation in most cases excluding Case V and VI among the comparison models. Compared to Proto Net, the results of Proto Net w/ OFM show that the operation-robust fault feature extraction module can improve the average diagnosis performance by 8 % and 14 % in 1UF and 2UF, respectively. Compared to Proto Net w/ OFM, PCDC results show the additional performance improvements of 6 % and 11 % owing to the prototype-assisted dual-contrastive learning. For the other comparison models, Fine-tuned Net shows similar performances for different unseen-fault scenarios, as the model is fine-tuned on target domain data.
Relation Net shows similar diagnosis trends as Proto Net or Siamese Net; however, its performance decreases more in Cases V and VI.

To compare the time efficiency, the processing time of each model is described in Table 12. It can be seen that the processing times for SPMSM dataset are shorter than those for IPMSM dataset, as SPMSM dataset contains fewer motor fault data. Similar to Case study 1, the processing time was slightly increased from 0.7 seconds to 3.3 seconds due to additional pre-processing steps in PCDC. The mean training time for PCDC is 13.5 seconds, which is 3 seconds longer than the mean training time of Fine-tune Net, which shows the fastest convergence speed. The deviation in total processing times among the models is approximately 8 seconds, indicating relatively small variability.

Table 11. Average fault diagnosis accuracy with standard deviation (%) in the target domain including unseen fault categories under 5-way 5-shot settings;

<table>
<thead>
<tr>
<th>Model</th>
<th>Average test accuracy ± STD (%) for 5-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1UF cases</td>
</tr>
<tr>
<td></td>
<td>SIS1</td>
</tr>
<tr>
<td>PCDC (proposed)</td>
<td>98.2±2.4</td>
</tr>
<tr>
<td>Proto Net w/ OFM</td>
<td>96.3±5.8</td>
</tr>
<tr>
<td>Proto Net</td>
<td>86.4±2.9</td>
</tr>
<tr>
<td>Siamese Net</td>
<td>81.3±12.8</td>
</tr>
<tr>
<td>Relation Net</td>
<td>78.2±14.7</td>
</tr>
<tr>
<td>Fine-tuned Net</td>
<td>79.6±4.4</td>
</tr>
</tbody>
</table>
Figure 12. Chart for fault diagnosis accuracy in the target domain including unseen fault categories under 5-way 5-shot settings.

Table 12. The comparison of processing time of the SPMSM dataset by each model

<table>
<thead>
<tr>
<th>Model Name</th>
<th>PCDC</th>
<th>Proto Net w/ OFM</th>
<th>Proto Net</th>
<th>Siamese Net</th>
<th>Fine-tuning Net</th>
<th>Relation Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data pre-processing time (s)</td>
<td>3.3</td>
<td>3.3</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>Mean training time (s)</td>
<td>13.5</td>
<td>13.0</td>
<td>11.2</td>
<td>14.6</td>
<td>10.3</td>
<td>18.6</td>
</tr>
<tr>
<td>Total processing time (s)</td>
<td>16.8</td>
<td>16.3</td>
<td>11.9</td>
<td>15.3</td>
<td>11.0</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Figure 13 shows the confusion matrix for PCDC and Proto Net in 1UF-Case III and 2UF-Case VI. The vertical axis denotes the true label assigned to the test samples in the target domain and the horizontal axis represents the predicted label inferred from the model. In the case of Proto Net, the overall fault diagnosis accuracy for the examples is 84.4 % in Case III and 65.7 % in Case VI, respectively. Even though Proto Net classifies well different fault modes (Nor, SIS, and MIS) in both experimental scenarios, it misclassifies severity classes under both cases, especially severe in Case VI. It can be seen that as the number of unseen categories increases, severity diagnosis performance deteriorates due to insufficient extracted features. Also, the results also show that classifying MIS1 and MIS2 is more difficult than classifying SIS1 and SIS2. However,
in the case of PCDC, the severity categories are successfully classified owing to OFM and prototype-assisted dual-contrastive learning, as shown in Figures 13a and Figures 13b. By removing the influence of the operating conditions through OFM and by learning global- and local-similarity features through prototype-assisted dual-contrastive learning, PCDC becomes discriminative for small datasets enough to classify severity classes even when the torque condition changes. Especially, prototype-assisted dual-contrastive features push all different severity categories and brings samples closer in the same category; this indicates the PCDC becomes discriminative for small datasets enough to classify severity classes. To sum up, the confusion matrix of PCDC shows improvements in severity diagnosis performance compared to Proto Net, which means that PCDC can precisely diagnose both fault and severity categories in both cases.

Figure 14 shows the t-distributed stochastic neighbor embedding (t-SNE) visualization for features extracted by the models under 2UF Case V. For Proto Net, Siamese Net, and Fine-tune Net, though the features are well separated by different fault modes (Nor, SIS, and MIS), the distribution of the features is mixed by severity class, SIS1 and SIS2 and MIS1 and MIS2, due to insufficient discriminative features. For Relation Net, the distribution of the features is so mixed that it cannot even distinguish normal category from the other fault modes. Although the features extracted by Proto Net w/ OFM are better than those by the comparison models, the distribution of the features is still mixed by severity class of MIS. Even though OFM increases diagnosis performances by removing the influence of the operating conditions, the severity class of MIS cannot be classified well because the features of MIS1 and MIS2 are so similar, and thereby it is difficult to learn the discriminative features for them. Further, it can be inferred that Proto Net w/ OFM overfitted the similarity features of the source domain dataset (Nor, SIS1, and SIS2); this makes it difficult to be used in classifying the target domain dataset. For PDCD, however, even the distribution of the features of the severity class of MIS is clearly separated. By using prototypes in dual-contrastive learning, which maintains the instance-discrimination ability while alleviating the overfitting, the features extracted by PCDC are well-separated by each category, at equal distances with low intra-class variance. It can be summarized that PCDC can precisely classify both motor fault and severity categories simultaneously even in the two unseen fault case.
Figure 13. Confusion matrix under different unseen fault scenarios under 5-way 5-shot settings: (a) PCDC for Case III, (b) PCDC for Case VI, (c) Proto Net for Case III, and (d) Proto Net for Case VI.
Figure 14. t-SNE visualization for extracted features by the models for Case V under 5-way 5-shot settings: (a) PCDC, (b) Proto Net w/ OFM, (b) Proto Net, (d) Siamese Net, (e) Fine-tuned Net, and (f) Relation Net.

5. Limitations and future works

Despite the overall success of PCDC, some limitations still remain, specifically: diagnosis under a) transient operating conditions and b) noisy conditions. While PCDC can adapt well to diverse steady-state operating conditions, it cannot adapt well to transient operating conditions where both load and/or speed change continuously. In steady-state operating conditions, preprocessing methods used in PCDC eliminate the influence of operating conditions on the motor current signals by scaling diverse torque and speed levels. In transient operating conditions, however, the preprocessing methods cannot eliminate the influence of operating conditions as they cannot reflect changes in acceleration over time. Therefore, PCDC might underperform in transient operating conditions. For future work, domain-invariant feature learning will be incorporated into a few-shot learning model to reduce the effect of transient operating conditions. A domain-invariant feature will be defined as a shared feature among datasets obtained from a transient operating condition. The feature plays a crucial role in enabling the model to diagnose unseen faults even under transient operating conditions.

Also, PCDC is relatively vulnerable to a noise compared to conventional deep learning models that are trained on a large amount of data. The difficulty arises from the
limited data usage of few-shot learning, making it hard for the model to generalize to noisy data distributions. To evaluate robustness to noise, experimental evaluations were conducted. Diverse levels of Gaussian noise were introduced to the target domain datasets to generate a set of noisy datasets. Signal-to-noise ratio (SNR) were used to determine the noise level. SNR in decibel scale is defined as follows:

\[
\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right)
\]

where \( P_{\text{signal}} \) and \( P_{\text{noise}} \) represent the power of the original signal and Gaussian noise, respectively. The evaluation was conducted based on SNR corresponding to the set \{-8, -4, 0, 4, 8\}. The few-shot fault diagnosis performance largely dropped when SNR values fell below four. For future work, a de-noising autoencoder will be incorporated into a few-shot learning model to enhance robustness to noise. The de-noising autoencoder will be trained with a virtual noisy signal as input and the original signal as output, aiming to help the few-shot learning model generalize to noisy environments.

### 6. Conclusions

In this paper, a novel prototype-assisted dual-contrastive learning with depth-wise convolutional neural network (PCDC) approach was proposed for few-shot fault diagnosis for motors using motor stator current signals. Using the proposed method, unseen fault categories under new operating conditions can be precisely classified owing to the discriminative similarity features and its generalization ability for diverse operating conditions. The proposed method consists of the operation-robust fault feature extraction module (OFM) and the few-shot diagnosis module (FDM). In OFM, operation-robust fault features are learned to reinforce generalization of PCDC for fault diagnosis under new operating conditions by extracting fault-induced amplitude and frequency modulation features and by eliminating the influence of operating conditions from the motor stator current signals. In FDM, prototype-assisted dual-contrastive learning was proposed, which minimizes the similarity distance loss of the prototypes-instance pairs where instance embeddings are transferred from the IB, to learn both the global similarity features and the low-level, instance-wise similarity features; this helps the proposed model to be more discriminative and generalize on the dataset.

The proposed method was validated through two experimental studies, including an interior permanent magnet synchronous motor dataset and surface-permanent magnet synchronous motor dataset. The results of the proposed PCDC showed average diagnosis performance improvement of 1) 4 % and 14 % in the incremental and decremental working condition shift scenarios, respectively, in Case 1; and 2) 14 % and 25 % in one unseen fault scenarios and two unseen faults scenarios, respectively, in
Case 2, as compared with Proto Net which achieves best performance among the comparison models. It can be confirmed from the results that OFM which makes the pairwise scaled inputs has an impact in adjusting well to different operating condition, while the prototype-assisted dual-contrastive learning in the few-shot diagnosis module helps learns sufficient task-specific sample-relation features. However, the proposed method has a limitation that it cannot adapt well in environments where both load and/or speed change continuously. However, the proposed method has the limitations that it cannot adapt well to transient operating conditions or noisy conditions.

In future work, domain shift problem due to transient operating condition will be considered. For example, sharp speed and/or load variation shifts data distribution largely, which poses a severe domain adaptation challenges for the existing few-shot fault diagnosis methods. Therefore, domain-invariant feature learning will be incorporated into the few-shot algorithm to reduce the effect of transient operating conditions. Also, methods for enhancing noise robustness of few-shot learning models will be studied. As few-shot learning models use limited amount of data, they are relatively vulnerable to noise compared to conventional deep learning models. Therefore, advanced deep learning algorithms, such as de-noising autoencoder, will be explored to help the models generalize on noisy environments.

Acknowledgment

This work was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008691, HRD Program for Industrial Innovation), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2021R1F1A1064082, 2022K1A4A7A04096329), and the Korea government (MSIT) (IITP-2024-RS-2022-0015630, Innovative HRD for Local Intellectualization) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References


Chen, Y., Rao, M., Feng, K., & Niu, G. (2023). Modified Varying Index Coefficient Autoregression Model for Representation of the Nonstationary Vibration From a


