CoUDA: Continual Unsupervised Domain Adaptation for Industrial Fault Diagnosis under Dynamic Working Conditions (Supplementary Material)

A. Algorithm

The algorithm of the CoUDA framework is summarized in Algorithm 1.

Algorithm 1 : CoUDA Framework

Stage 1: Source Domain Pre-training
Input: Source data $\mathcal{D}_{\mathcal{S}}$, feature extractor f_θ, classifier g_θ.
Hyperparameters: Total epochs E, learning rate η, batch size b, momentum m, temperature τ, balance parameter α.

```
for e ← 0 to E do
    for every mini-batch (x^S, y^S) ∈ D_S do
        Infer. L^S(x^S, y^S; θ_S) = L_IPC + α · L_IIC;
        SGD. θ_S ← θ_S − η · ∇L^S(x^S, y^S; θ_S).
    end for
end for
```

$\mathcal{L}^{\mathcal{S}}\left(x^{\mathcal{S}}, y^{\mathcal{S}}; \theta_{\mathcal{S}}\right) = \mathcal{L}_{\text{IPC}} + \alpha \cdot \mathcal{L}_{\text{IIC}}$; $\theta_{\mathcal{S}} \leftarrow \theta_{\mathcal{S}} - \eta \cdot \nabla \mathcal{L}^{\mathcal{S}}\left(x^{\mathcal{S}}, y^{\mathcal{S}}; \theta_{\mathcal{S}}\right)$.

Return: Pre-trained feature extractor $f_{\theta_{\mathcal{S}}}$, classifier $g_{\theta_{\mathcal{S}}}$.

Stage 2: Sequential Target Domain Adaptation
Input: Source data $\mathcal{D}_{\mathcal{S}}$, pre-trained feature extractor $f_{\theta_{\mathcal{S}}}$, classifier $g_{\theta_{\mathcal{S}}}$, unlabeled sequential target data $\{\mathcal{D}_{\mathcal{T}_t}\}_{t=1}^{T}$.
Hyperparameters: Total epochs E, learning rate η, batch size b, momentum m, temperature τ, balance parameter α.

```
for t ← 1 to T do
    Initialize f_θt ← f_θt−1, g_θt ← g_θS;
    for e ← 0 to E do
        for every mini-batch (x^S, y^S) ∈ D_S, (x^Tt) ∈ D_Tt do
            Infer. L^Tt(x^S, y^S, x^Tt; θ_t, θ_t−1) = L_CE + L_FKD
                       + β(e) · L_IEM + (1 − β(e)) · L_LDA;
            SGD. θ_t ← θ_t − η · ∇L^Tt(x^S, y^S, x^Tt; θ_t).
        end for
        if t > 1 then
            evaluation on previous target domains {D_T1, ..., D_Tt};
        end if
end for
```

$\mathcal{L}^{\mathcal{T}_t}\left(x^{\mathcal{S}}, y^{\mathcal{S}}, x^{\mathcal{T}_t}; \theta_t, \theta_{t-1}\right) = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{FKD}} + \beta(e) \cdot \mathcal{L}_{\text{IEM}} + (1-\beta(e)) \cdot \mathcal{L}_{\text{LDA}}$; $\theta_t \leftarrow \theta_t - \eta \cdot \nabla \mathcal{L}^{\mathcal{T}_t}\left(x^{\mathcal{S}}, y^{\mathcal{S}}, x^{\mathcal{T}_t}; \theta_t\right)$; evaluation on previous target domains $\{\mathcal{D}_{\mathcal{T}_1}, \cdots, \mathcal{D}_{\mathcal{T}_t}\}$.

Return: New feature extractor f_{θ_t}, global classifier $g_{\theta_{\mathcal{S}}}$.

B. Prototype Contrastive Learning (PCL)

As shown in Fig. 1, we compare the performance of the source-only model with and without PCL on the target domains. The source-only model with PCL achieves better performance in all unseen target domains, which demonstrates the effectiveness of PCL in improving the generalization ability of the source model. The removal of PCL reduces the generalization ability of the source model, leading to a reduction in its ability of both adaptation and anti-forgetting.

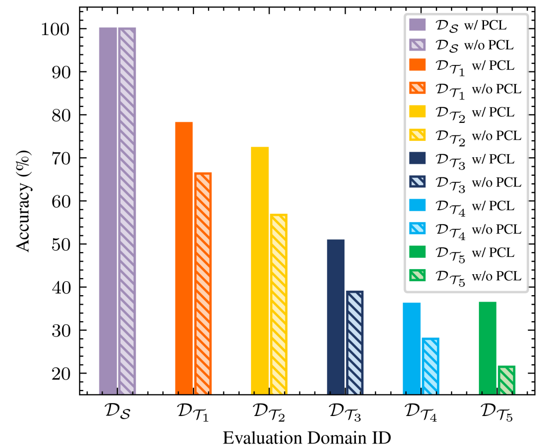

Fig. 1. The performance comparison of source-only model w/ and w/o PCL.

C. Hyperparameter Sensitivity Analysis

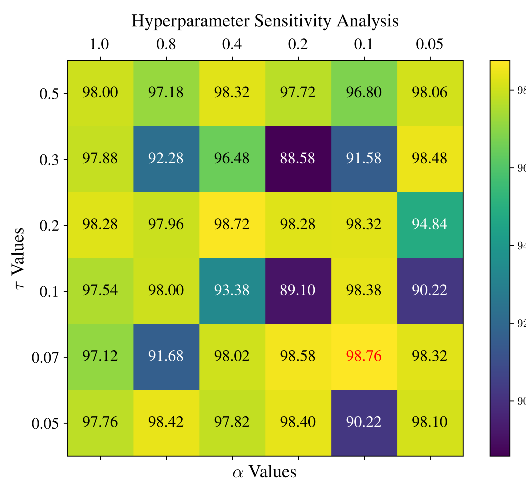

Fig. 2. Hyperparameter sensitivity analysis. The ACC of CoUDA with different hyper-parameters $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$.

We conducted a grid search of the hyper-parameters τ and α, which are the temperature parameter in the IIC loss and the balance parameter between the IPC and IIC losses, respectively. To evaluate the sensitivity of the hyper-parameters, we varied the values of τ and α from 0.05 to 0.5 and from 1.0 to 0.05, respectively. We plotted the heatmap of the ACC with different hyper-parameters using the domain scenario S1 for the analysis, as shown in Fig. 2. The best performance is achieved when $\tau = 0.07$ and $\alpha = 0.1$.