Boosting Domain Incremental Learning: Selecting the Optimal Parameters is All You Need Supplementary Materials

A. Appendix

A.1. Training and Inference Algorithms

Algorithm 1 illustrates the training and inference details of the SOYO framework, which are discussed in Sec. 3.2.

Algorithm 1 SOYO Algorithms for Domain Incremental Learning

Input: Training set $(\mathcal{X}_t, \mathcal{Y}_t)$, test set $(\mathcal{X}'_t)$, backbone network f

Output: The predicted results y' of the test set

```
# 1. Training stage
for t = 1 to T do
    for (x, y) in (X_t, Y_t) do
        Train the additional parameters φ_t
        Compress the domain features X^D as Gaussian distribution parameters θ = {λ_k, μ_k, Σ_k}_{k=1}^K using Eq. (4) and Algorithm 2
    end for
    if t > 1 then
        Resample the pseudo-domain features X̃^D with parameters θ using Eq. (5)
        for x̃^D in X̃^D do
            Train the MDFN parameters Δ = {δ_1, δ_2, δ_3} using Eq. (2)
        end for
    end if
end for
# 2. Inference stage
for t = 1 to T do
    for x' in {X'_t}_{t=1}^T do
        Predict the domain label of x' as t̂ using MDFN
        Output the results using the backbone network f and the additional parameters φ_t̂ as y'
    end for
end for
```

A.2. Expectation-Maximization Algorithm

Algorithm 2 presents the Expectation-Maximization (EM) algorithm used in the Gaussian mixture compressor. We abbreviate $\{\mathbf{x}_{\tau,i}^l\}_{i=1}^{N_\tau}$ as $\{x_1, x_2, \ldots, x_N\}$ for brevity.

A.3. Details of Evaluation Metrics

The average classification accuracy, A_T, represents the mean test accuracy across the first T domains after training on the T-th domain. It is calculated as:

$$A_T = \frac{1}{T}\sum_{i=1}^{T} B_{i,T}, \tag{7}$$

where $B \in \mathbb{R}^{T\times T}$ is an upper triangular matrix, and $B_{i,j}$ indicates the test accuracy of the i-th domain after training on the j-th domain.

The forgetting degree, F_T, measures the performance degradation on previous domains caused by learning new domains. It is calculated as:

$$F_T = \frac{1}{T-1}\sum_{i=1}^{T-1} BWT_i, \tag{8}$$

$$BWT_i = \frac{1}{T-i}\sum_{j=i+1}^{T} (B_{i,j} - B_{i,i}), \tag{9}$$

where BWT_i represents the mean backward transfer degradation for the i-th domain.

The parameter selection accuracy, S_T, reflects the accuracy of predicting domain-specific parameters, which is equivalent to the domain label prediction accuracy. S_T is defined as the average parameter selection accuracy across the first T domains, computed as:

$$S_T = \frac{1}{T}\sum_{i=1}^{T} S'_i, \tag{10}$$

where S'_i denotes the parameter selection accuracy for the i-th domain.

Algorithm 2 EM Algorithm for Gaussian Mixture Compressor

Require: Dataset $\{x_1, x_2, \ldots, x_N\}$, number of Gaussian components K, convergence threshold ϵ

Ensure: Estimated parameters $\theta = \{\lambda_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

Initialize parameters $\lambda_k^{(0)}$, $\mu_k^{(0)}$, $\Sigma_k^{(0)}$ for $k = 1, 2, \ldots, K$

Set iteration counter $t \leftarrow 0$

repeat

  E-step (Expectation step):

  for $i = 1$ to N **do**

    for $k = 1$ to K **do**

      Compute the responsibility

$$\gamma_{ik}^{(t)} = \frac{\lambda_k^{(t)} \cdot \mathcal{N}(x_i \mid \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{j=1}^{K} \lambda_j^{(t)} \cdot \mathcal{N}(x_i \mid \mu_j^{(t)}, \Sigma_j^{(t)})}$$

    end for

  end for

  M-step (Maximization step):

  for $k = 1$ to K **do**

    Compute the effective number of samples $N_k^{(t)} = \sum_{i=1}^{N} \gamma_{ik}^{(t)}$

    Update the mixing coefficient $\lambda_k^{(t+1)} = \dfrac{N_k^{(t)}}{N}$

    Update the mean $\mu_k^{(t+1)} = \dfrac{1}{N_k^{(t)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} x_i$

    Update the covariance matrix $\Sigma_k^{(t+1)} = \dfrac{1}{N_k^{(t)}} \sum_{i=1}^{N} \gamma_{ik}^{(t)} (x_i - \mu_k^{(t+1)})(x_i - \mu_k^{(t+1)})^\top$

  end for

  Compute the log-likelihood $\ln L^{(t+1)} = \sum_{i=1}^{N} \ln \left(\sum_{k=1}^{K} \lambda_k^{(t+1)} \cdot \mathcal{N}(x_i \mid \mu_k^{(t+1)}, \Sigma_k^{(t+1)}) \right)$

  Compute the log-likelihood increment $\Delta L = \left| \ln L^{(t+1)} - \ln L^{(t)} \right|$

  Update the iteration counter $t \leftarrow t + 1$

until $\Delta L < \epsilon$