Few-Shot Hybrid Incremental Learning: Continually Learning under Data Scarcity and Task Uncertainty

Supplementary Material

1. More Methodological Details

Following the methodology of [1, 3, 4], we implement an autoencoder as the distribution detection module, which comprises an encoder and a decoder. Each expert is paired with a corresponding distribution detection module, which learns the feature distribution corresponding to the expert's knowledge by minimizing the reconstruction loss, defined as:

$$\mathcal{L}_{\mathbf{D}} = \sum_{l \in \mathcal{S}^L} \sum_{k \in \mathcal{S}^K_{l,\text{new}}} \|\mathbf{x}^l - \mathbf{D}_{l,k}(\mathbf{x}^l)\|^2, \tag{1}$$

where $\mathcal{S}^L$ is the set of layers and $\mathcal{S}^K_{l,\text{new}}$ denotes the set of newly added $\mathbf{D}_{l,*}(\cdot)$ in layer l.

Upon receiving new data, the distribution detection module quantifies the deviation from the established feature distribution by computing the reconstruction loss. Inspired by established practices [4], we employ the z-score transformation of L rather than the raw loss, which provides a scale-invariant standardized anomaly score suitable for comparison and thresholding. The z-score is calculated as:

$$\text{z-score} = \frac{\mathcal{L}_{\mathrm{D}} - \mu}{\sigma}, \tag{2}$$

where μ and σ denote the mean and standard deviation of the reconstruction loss recorded from the historical data, respectively. The z-score is used to evaluate the deviation of the current data distribution from those learned by existing experts. If the z-score exceeds a predefined threshold τ_z, a new expert is expanded to learn the new data distribution; otherwise, existing experts are reused to handle the data.

2. More Experimental Details

Hyper-Parameters Configuration. Table 1 provides the configuration settings for various hyper-parameters across different datasets in FSHIL, including incremental learning rate is for expert expansion training, fine-tuning rate for expert reuse training, and distribution rate for distribution detection training.

3. More Experiments and Analysis

Comparison of Forgetting Rates with State-of-the-Art Methods. We present a comparison of forgetting rates between our method and the competing approaches in Figure 1. The results demonstrate that our method exhibits the lowest forgetting rate across nearly all datasets, highlighting its ability to not only achieve superior accuracy but also maintain minimal forgetting.

Table 1. Hyper-parameters configuration on five datasets in FSHIL. Incremental learning rate is for expert expansion training, fine-tuning rate for expert reuse training, and distribution rate for distribution detection training.

Dataset	Incremental learning rate	Fine-tuning learning rate	Distribution learning rate
Office-31	0.02	0.01	0.01
Office-Home	0.02	0.01	0.01
iDigits	0.1	0.1	0.01
CORe50	0.01	0.01	0.01
DomainNet	0.01	0.005	0.01

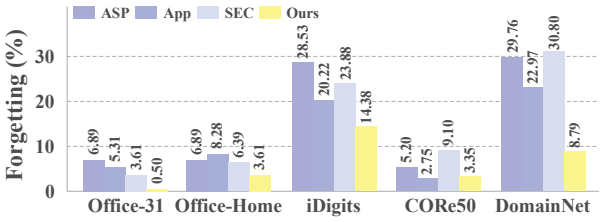


Figure 1. Forgetting comparison on five datasets in FSHIL.

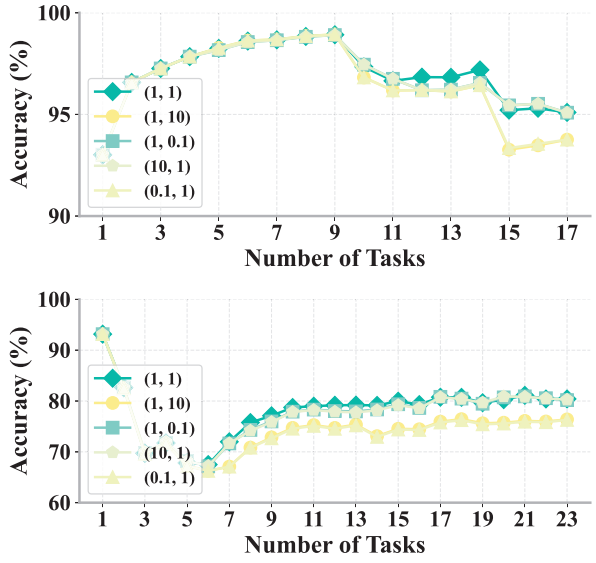


Figure 2. Ablation study of the loss coefficients λ_1 and λ_2 on Office-31 and CORe50 datasets.

Table 2. Ablation study of the proposed method in FSCIL and FSDIL on five datasets.

Component		Office-31		Office-Home		iDigits		CORe50		DomainNet	
CME-MoE	SEPC	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
FSCIL											
✗	✗	39.41	16.09	37.20	15.19	51.36	24.80	39.42	16.09	20.09	6.65
✓	✗	78.08	69.23	70.20	66.21	46.42	24.22	44.05	30.95	26.33	28.64
✗	✓	60.99	41.95	71.32	56.47	52.48	29.31	56.85	30.88	38.81	25.55
✓	✓	**91.66**	**85.35**	**83.45**	**79.10**	**63.39**	**43.21**	**82.23**	**73.79**	**45.89**	**41.15**
FSDIL											
✗	✗	74.61	76.58	58.81	64.45	51.99	48.05	48.98	49.26	31.89	25.38
✓	✗	82.73	90.31	67.07	79.40	64.07	61.40	65.82	75.59	32.56	32.27
✗	✓	78.85	89.25	71.15	73.59	74.42	65.85	79.71	77.03	51.83	43.17
✓	✓	**91.69**	**93.84**	**80.22**	**82.47**	**85.46**	**83.13**	**86.61**	**87.01**	**52.92**	**44.05**

Table 3. Ablation study of expert initialization on CORe50 dataset.

Expert Initialization	Avg	Last
Ours (w/ Random Init)	73.90	75.37
Ours (w/ Meta-learned Init)	**78.27**	**80.84**

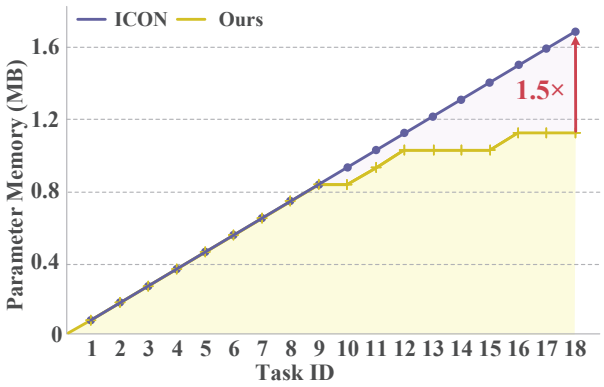
Analysis of Loss Coefficients λ_1 and λ_2 on More Datasets. Figure 2 shows the ablation study of the loss coefficients λ_1 and λ_2 on the Office-31 and CORe50 datasets. Experimental results on additional datasets further demonstrate the robustness of these two coefficients, and the optimal performance is achieved when $\lambda_1 = 1$ and $\lambda_2 = 1$.

Ablation study of the proposed method in FSCIL and FSDIL on five datasets. To further validate the effectiveness of our method in various scenarios, we conduct ablation studies in both FSCIL and FSDIL settings, as presented in Table 2.

Ablation study on expert initialization. Table 3 presents ablation studies on the initialization methods for expanded experts, demonstrating that our proposed meta-expansion strategy achieves superior performance compared to random initialization.

Analysis on Hyper-parameter τ_z in CME-MoE. We conduct an ablation study on the threshold τ_z using commonly adopted statistical thresholds [2], specifically 1.5, 2, and 3. The results are presented in Table 4.

Analysis on the Selection of Expert Insertion Layers. Table 5 presents the performance evaluation of selecting various expert insertion layers and allowable expansion layers. For the expert insertion layers, we investigate performance by choosing the first six layers, the intermediate six layers, and the final six layers of the model, respectively. Regarding the allowable expansion layers, we compare the results obtained by selecting the last layer, the last two layers, and the last three layers of each expert insertion layer choice. The results demonstrate that for the majority of datasets, optimal performance is achieved when the expert insertion layers are chosen as the first six layers and the allowable expansion layer is selected as the last layer.



(a) DomainNet

(b) Office-31

Figure 3. Memory overhead from expanded experts on DomainNet and Office-31 datasets.

Analysis of memory overhead from expanded experts. Figure 3 illustrates the cumulative memory overhead of the expanding experts in our proposed method during in-

Table 4. Ablation study on hyper-parameter τ_z in CME-MoE.

τ_z	Office-31		Office-Home		iDigits		CORe50		DomainNet	
	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
1.5	**96.97**	**95.10**	81.64	80.00	**67.42**	**60.77**	**78.27**	**80.84**	**47.20**	**40.97**
2.0	96.73	94.97	**81.78**	**80.33**	53.16	52.11	77.50	79.03	46.82	41.17
3.0	96.73	94.97	79.71	77.16	53.16	52.11	77.50	79.03	45.78	39.83

Table 5. Ablation study on the selection of expert insertion layers

Expert insertion layers	Allowable expansion layers	Office-31		Office-Home		iDigits		CORe50		DomainNet	
		Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
[0, 1, 2, 3, 4, 5]	[5]	**96.97**	**95.10**	**81.64**	**80.00**	**67.42**	**60.77**	78.27	80.84	**47.20**	**40.97**
[0, 1, 2, 3, 4, 5]	[4, 5]	96.07	94.44	81.29	79.80	63.50	54.27	77.88	80.39	46.73	40.96
[0, 1, 2, 3, 4, 5]	[3, 4, 5]	95.68	92.85	81.34	79.76	25.23	16.44	78.10	80.32	47.08	40.03
[3, 4, 5, 6, 7, 8]	[8]	76.55	71.97	78.73	77.80	50.08	43.51	**79.02**	**81.16**	41.22	36.99
[3, 4, 5, 6, 7, 8]	[7, 8]	47.28	29.16	78.64	77.21	50.69	38.81	70.55	76.84	41.12	36.29
[3, 4, 5, 6, 7, 8]	[6, 7, 8]	48.87	30.33	73.41	74.93	49.65	40.24	75.24	79.35	41.58	36.49
[6, 7, 8, 9, 10, 11]	[11]	59.70	76.49	79.17	78.01	62.44	52.11	78.00	79.72	43.81	38.43
[6, 7, 8, 9, 10, 11]	[10, 11]	96.09	93.73	79.80	78.55	60.17	48.63	77.76	79.69	44.52	39.15
[6, 7, 8, 9, 10, 11]	[9, 10, 11]	96.03	93.25	77.04	75.88	27.44	23.12	76.84	78.25	45.86	40.02

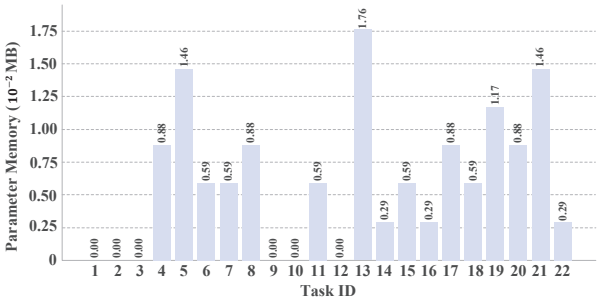


Figure 4. Memory overhead from expanded prototypes across tasks on CORe50 dataset.

cremental tasks, compared with ICON, which similarly employs an expansion strategy to address task uncertainty. We select both the distributionally complex DomainNet dataset and the simpler Office-31 dataset to fairly demonstrate the advantages of our method in terms of memory overhead.

Analysis of memory overhead from expanded prototypes. We illustrate the memory overhead of the expanded prototypes for each task on the CORe50 dataset in Fig. 4. It can be observed that the memory overhead from prototype expansion is negligible, with a maximum of merely 1.76×10^{-2} MB.

4. Training Pseudocode

The algorithmic procedure of the proposed method is detailed in Algorithm 1.

Algorithm 1: Few-Shot Hybrid Incremental Learning with **CME-MoE** and **SEPC**

Input: Sequential datasets $\{(\mathcal{X}^t, \mathcal{Y}^t)\}_{t=0}^{T}$ for tasks $\{t_1, \ldots, t_T\}$
Output: Continually learned model under data scarcity and task uncertainty

Step 1: Initialization
Initialize expert $\mathbf{E}_{l,0}(\cdot)$ and distribution detection module $\mathbf{D}_{l,0}(\cdot)$ in the MLP sub-layer of the Transformer layer $l \in \mathcal{S}^L$
Step 2: Pre-training
Meta-optimization for $\mathbf{E}_{l,0}(\cdot)$ and $\mathbf{D}_{l,0}(\cdot)$ with parameter set $\Theta_{l,0} = \{\Theta^{\mathrm{e}}_{l,0}, \Theta^{\mathrm{d}}_{l,0}\}$ using $(\mathcal{X}^0, \mathcal{Y}^0)$:
$\min_{\Theta_{l,0}} \mathbb{E}_{\mathcal{M}_i \sim p(\mathcal{M})} \left[\mathcal{L}^{\text{query}}_{\mathcal{M}_i}\left(\Theta_{l,0} - \alpha \nabla_{\Theta_{l,0}} \mathcal{L}^{\text{support}}_{\mathcal{M}_i}(\Theta_{l,0})\right)\right]$
// where $\mathcal{L}^{support}_{\mathcal{M}_i}$ and $\mathcal{L}^{query}_{\mathcal{M}_i}$ denote the losses on the support and query sets of task $\mathcal{M}_i$, respectively
# Step 3: Few-shot hybrid incremental learning
for *each task* t_T **do**
  while *not converged* **do**
    Sample mini-batch $\{(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i)\}_{i=1}^{B}$ from $(\mathcal{X}^t, \mathcal{Y}^t)$
    // CME-MoE: Conditional Meta-Expanding Mixture-of-Experts
    # Step 3.1: Expert reuse versus expansion
    Compute distribution error for each expert l and layer k: $\mathbb{E}^{\mathrm{D}}_{l,k} = \|\mathbf{x}^l_i - \mathbf{D}_{l,k}(\mathbf{x}^l_i)\|^2$
    Compute the z-score based on $\mathbb{E}^{\mathrm{D}}_{l,k}$
    if *z-score exceeds threshold* **then**
      # Step 3.1.1: Expert reuse
      Compute gating weights for existing experts: $\mathbf{m} = \mathrm{Softmax}(\mathbf{G}(\mathbf{x}^l_i))$, $\mathbf{m} = \{m_1, ..., m_{K_l}\}$
      // where $\mathbf{G}(\cdot)$ and $\mathrm{Softmax}$ denote a learned gating network and softmax function, respectively
      Reuse existing experts by aggregating the most relevant expertise: $\mathbf{y}^l_i = \sum_{i \in \{1,\ldots,K_l\}} m_i \cdot \mathbf{E}_{l,i}(\mathbf{x}^l_i)$
    end
    else
      # Step 3.1.2: Expert meta-expansion
      Expand experts via meta-expansion: $\Theta^{\mathrm{e}}_{l,K_l+1} \leftarrow \Theta^{\mathrm{e}}_{l,0}$, $\Theta^{\mathrm{d}}_{l,K_l+1} \leftarrow \Theta^{\mathrm{d}}_{l,0}$
    end
    // SEPC: Self-Expanding Prototype Classifier
    # Step 3.2: Performance monitoring
    Compute the classifier expansion threshold of each class c over its n incremental visits:
    $\mathcal{T}_c = \max\left(\overline{\mathrm{Acc}}(c, n) \cdot (1 - \beta),\ \tau_{\min}\right)$
    // where $\beta \in [0, 1)$ is the tolerance for performance decay and $\tau_{\min}$ is the minimum acceptable performance
    if *performance exceeds threshold* $\mathcal{T}_c$ **then**
      # Step 3.2.1: Classifier expansion
      Compute new prototype for the each underperforming class c: $\mathbf{p}_{c,j} = \frac{1}{|\mathcal{S}^c|} \sum_{(\hat{\mathbf{x}}_i, c) \in \mathcal{S}^c} \mathbf{F}(\hat{\mathbf{x}}_i)$
      // where $\mathbf{F}(\cdot)$ denotes the model backbone, $\hat{\mathbf{x}}_i$ is the original sample, and j indexes the prototype within the c-th class
      Add the expanded prototype $\mathbf{p}_{c,j}$ to the prototype set $\mathcal{S}^{P_c}$
    end
    else
      # Step 3.2.2: Classifier reuse
      Compute each logit Logit_c as the maximum similarity between $\mathbf{f}$ and prototypes $\mathbf{p}_{c,j} \in \mathcal{S}^{P_c}$ for class c:
      $\mathrm{Logit}_c = \max_j \mathrm{sim}\left(\mathbf{f}, \mathbf{p}_{c,j}\right), \quad \mathbf{p}_{c,j} \in \mathcal{S}^{P_c}$
      Compute the final prediction $\hat{c}$ as the class label corresponding to the maximum logit:
      $\hat{c} = \arg\max_{c \in \{1,\ldots,C\}} \mathrm{Logit}_c$
    end
    # Step 3.3: Loss composition and update
    Compute loss $\mathcal{L}_{\mathrm{D}}$ for distribution detection modules, and $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{PD}}$ for experts and classifier:
    $\mathcal{L}_{\text{total}} = \mathcal{L}_{\mathrm{D}} + \lambda_1 \mathcal{L}_{\mathrm{CE}} + \lambda_2 \mathcal{L}_{\mathrm{PD}}$
    Backpropagate and update experts, distribution detection modules and classifier parameters
  end
end

References

[1] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 1

[2] Robert V Hogg, Elliot A Tanis, and Dale L Zimmerman. *Probability and statistical inference*. Macmillan New York, 1977. 2

[3] Minqi Jiang, Songqiao Han, and Hailiang Huang. Anomaly detection with score distribution discrimination. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 984–996, 2023. 1

[4] Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10087–10098, 2025. 1