# Cost-Effective Incremental Deep Model: Matching Model Capacity With the Least Sampling

Yang Yang, Da-Wei Zhou, De-Chuan Zhan, Hui Xiong, *Fellow, IEEE*, Yuan Jiang, and Jian Yang, *Member, IEEE*

**Abstract**—Most existing approaches often utilize the pre-fixed structure and large number of labeled data for training complex deep models, which are difficult to implement on incremental scenarios. As a matter of fact, real-world data is always in stream form. Thereby, there exits two challenges for building incremental deep models: a) *Capacity Scalability*. The entire training data is not available before learning the task. It is a challenge to make the deep model structure scale with streaming data for flexible model evolution and faster convergence. b) *Capacity Sustainability*. The distribution streaming data usually changes in nature (concept drift), thus it is necessary to update the model while preserving previous knowledge for overcoming the catastrophic forgetting. To this end, we develop an incremental deep model (IDM), which expands the network structure according to streaming data and slows down forgetting with the adaptive fisher regularization. However, IDM ignores another significant challenge with streaming data: c) *Capacity Demand*. Training a deep model always needs a large amount of labeled data, whereas it is almost impossible to label all unlabeled instances in real time. The core problem is to select a small number of the most discriminative instances to label while keeping the predictive accuracy of the model. Thereby, we focus on the online semi-supervised learning scenario with abrupt changes in data distribution, and further improve IDM to a cost-effective incremental deep model (CE-IDM), which can adaptively select the most discriminative newly coming instances for query to reduce the manual labeling costs. Specifically, CE-IDM adopts a novel extensible deep network structure by using an extra attention model for hidden layers. Based on the adaptive attention weights, CE-IDM develops a novel instance selection criterion by jointly estimating unlabeled instances' representative and informative degree to satisfy the capacity demand. With the newly labeled instances, CE-IDM can quickly update the model with adaptive depth from streaming data and enable capacity scalability. Also, we address capacity sustainability by exploiting the attention based fisher information matrix, which can slow down the forgetting in consequence. Finally, CE-IDM can deal with the three capacity challenges methioned above in a unified framework. We conduct extensive experiments on real-world data and show that CE-IDM outperforms the state-of-the-art methods with a substantial margin.

**Index Terms**—Incremental deep learning, active sampling, capacity scalability, capacity sustainability, capacity demand

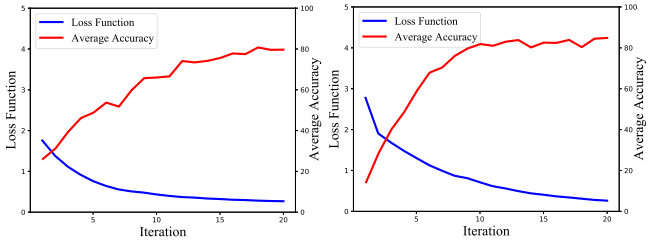

- *Yang Yang is with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Jiangsu Key Lab of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China, and with the Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing, Jiangsu 211189, China, and also with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210094, China. E-mail: yyang@njust.edu.cn.*
- *Da-Wei Zhou, De-Chuan Zhan, and Yuan Jiang are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, China. E-mail: zhoudw@lamda.nju.edu.cn, {zhandc, jiangy}@nju.edu.cn.*
- *Hui Xiong is with Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 07102 USA. E-mail: hxiong@rutgers.edu.*
- *Jian Yang is with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, Jiangsu Key Lab of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China. E-mail: csjyang@njust.edu.cn.*


*Manuscript received 20 Mar. 2020; revised 30 Oct. 2021; accepted 30 Nov. 2021. Date of publication 7 Dec. 2021; date of current version 7 Mar. 2023.*
*This work was supported in part by NSFC under Grants 62006118, 61906092, 61773198, and 91746301. The Natural Science Foundation of Jiangsu Province of China under Grants BK20200460 and BK20190441, and in part by Jiangsu Shuangchuang (Mass Innovation and Entrepreneurship) Talent Program, CAAI-Huawei MindSpore Open Fund under Grant CAAIXSJLJJ-2021-014B.*
*(Corresponding author: Yang Yang.)*
*Recommended for acceptance by B. Moseley.*
*Digital Object Identifier no. 10.1109/TKDE.2021.3132622*

## 1 Introduction

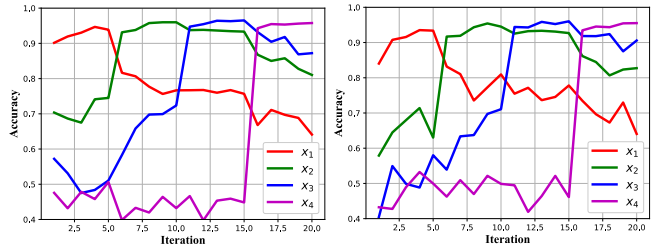
Nowadays, a large amount of streaming data, such as traffic flows, sensor data, and query logs, has been accumulated in many application scenarios. As a result, there is a critical need for developing incremental learning methods [1]. Indeed, tremendous efforts have been made in different application domains, such as incremental recommendation [2], demand prediction [3], and graph matching [4]. However, most existing incremental learning methods are in shallow structures (e.g., linear or kernel) [5], [6], which are not designed to learn complex nonlinear functions. As we know, deep learning techniques have achieved a wide range of successes with powerful nonlinear models, such as recommendation [7], [8], article analysis [9], and semantic representation [10]. However, existing deep models always require entire training data and are not designed for incremental learning tasks. Therefore, there is a need to perform *Incremental Deep Learning* (IDL).

A direct way to do IDL is applying the standard backpropagation training for the pre-fixed model. Such an approach is simple but has several limitations, particularly for solving the model capacity issue. Different from off-line learning that requires the entire training data available in prior, incremental learning manages to optimize classifiers over the streaming data. Thereby IDL requires the models to




(*a*) Shallow Model (*b*) Deep Model

Fig. 1. Performance measure (Loss versus Accuracy) with different network structures on CIFAR-10. (a) Network with 18 layers (ResNet18); (b) network with 34 layers (ResNet34).




(*a*) Shallow Model (*b*) Deep Model

Fig. 2. Catastrophic forgetting phenomenon. In detail, we construct 4 stages from MNIST dataset, and remove 1/4 part from the images ($X_1$ with top left part removed, $X_2$ with top right part removed, $X_3$ with bottom left part removed, $X_4$ with bottom right removed), which is submitted to the concept drift scenario [24]. The results reveal that with the streaming data, the previous stages will appear forgetting, as the accuracy is decreasing for both shallow and deep models.

have flexible structures, which can scale with the streaming data for convergence and performance improvement. As shown in Fig. 1, the learning process will converge slowly if the model is too complex, while the capacity is restricted if the model is simple. We define this challenge as "*Capacity Scalability*". Furthermore, it is notable that streaming data is always evolving in nature, i.e., the joint distribution between the input feature and the ground truth will change as the concept drift [11]. If we ignore the distribution change, the performance of previous distribution will dramatically drop down with the catastrophic forgetting phenomenon [12]. E.g., Fig. 2 indicates that the knowledge learned from the previous distribution ($X_1$) will be lost when information relevant to the current distribution ($X_2$) is incorporated. We define this challenge as "*Capacity Sustainability*". However, previous IDL methods rarely consider this crucial problem. Recently, fisher information matrix is introduced to prevent this problem [13], [14], [15], whereas these methods concentrate on the life-long multi-task learning with the obvious task conversion, i.e., significant category changes occur in different tasks. Moreover, these methods ignore that the importance of different parts in the fisher information matrix is also adapting with the evolution of model structure.

Another nonnegligible challenge is that training IDL models first needs to label all the streaming unlabeled data, whereas it is almost impossible to label them all in real time. A direct approach is to store this data in an off-line form, manually label them, and then update the model. But this is costly when fast model update is required to cope with stream mining demands, and contrary to the basic assumption of incremental learning. Therefore, an effective method is to select a small number of the most discriminative instances for labeling while keeping the predictive accuracy. This challenge can be defined as the "*Capacity Demand*", and can be solved by active sampling. Traditional active learning is designed for reducing the labeling cost [16], [17], which tries to train an effective model with less queries to reduce the overall cost. There are two mainly criteria for active selecting in previous methods [18], [19], [20]: 1) Informativeness, which measures the ability of an instance in reducing the uncertainty [21], [22]; and 2) Representativeness, which measures the representation of an instance to the overall input patterns [23]. However, previous active sampling strategies are usually based on static environment with all the previous data, which is difficult to apply in stream selection directly.

These three problems are always co-existing, and impose challenges to develop IDL models. In this paper, we focus on the online semi-supervised learning scenario with abrupt changes in data distribution, i.e., we receive the semi-supervised streaming data, and drift may happen suddenly/abruptly by switching from one concept to another. To this end, we design the "Cost-Effective Incremental Deep Model" (CE-IDM), a novel adaptable deep model that can handle three capacity challenges in one unified framework naturally. In detail, first, CE-IDM can evolve from a shallow network (fast convergence) to deep model (large capacity) with the streaming data by combing a novel adaptive attention network, which highlights the importance of each hidden layer gradually, and CE-IDM is knowledgeable about the past and present data distribution with the adaptive fisher regularization. Second, CE-IDM designs an adaptive instance selection criterion by jointly considering the informativeness and representativeness, which can satisfy the capacity demand of streaming data and reduce the labeling cost.

## 2 Related Work

Incremental modeling aims to update the models from data stream sequentially, and has achieved many successes in both application and theory [25], [26]. As a matter of fact, incremental deep learning can directly adopt online back-propagation, yet with many drawbacks, e.g., convergence limitation (gradient vanishing and diminishing feature reuse) [27]. Thus, Lee *et al.* proposed a dual memory architecture to process slow-changing global patterns [28]; Zhou *et al.* proposed an incremental feature learning algorithm to determine the optimal model complexity based on the auto-encoder [29]. The most relevant work to our approach is [27], which proposes a novel framework for deep model in the incremental setting, and adapts the model capacity from simple to complex incrementally. However, the crucial parameters of the proposed Hedge Backpropagation (HBP), i.e., the weights for adapting the depth, are defined separately and tuned difficultly.

Concept drift caused by the distribution evolution is a well-recognized research direction in incremental learning [11], [30]. Previous methods can be divided into three categories: sliding window based approach [31], evolving

based approach [32], and ensemble based approach [33]. However, these methods ignore an important phenomenon in incremental learning, catastrophic forgetting, which is the tendency of losing the learned knowledge from previous distribution. To mitigate the catastrophic forgetting, there are many attempts, including ensemble methods combining multiple classifiers for final prediction [34], rehearsal methods mixing data from earlier sessions [35], dual-memory models storing memories in two distinct neural networks [35], and sparse-coding methods reducing the forgetting by learning sparse representation [36]. Readers can refer to the introduction for further information [37]. Recently, many researches concentrate on utilizing the fisher information matrix as regularization and have achieved excellent performance. E.g., Kirkpatrick *et al.* proposed the elastic weight consolidation to reduce catastrophic forgetting in artificial neural networks [13]. Lee *et al.* proposed to incrementally match the moment of the posterior distribution of the neural network [15]. Lee *et al.* dynamically decided the network capacity for lifelong learning [38]. Nevertheless, these methods are designed for multi-task learning, they require clear task segmentation, and can not be directly applied to the concept drift setting.

Therefore, in the conference version, we developed an incremental deep model (IDM), which adopts an extra attention model to learn deep models with adaptive depth, and further exploits the attention based fisher information matrix to mitigate the forgetting. *It is notable that IDM belongs to the supervised learning, since it requires all streaming data to be labeled.* However, it is almost impossible to label all unlabeled streaming instances in real time. Thereby, in this manuscript, *we combine a novel incremental active sampling method and IDM into a unified DIL framework, and proposed the cost-effective incremental deep model (CE-IDM).* The most popular approach for reducing the labeling cost is probably the active learning. Classical approaches include random sampling, uncertainty sampling [39], [40], query-by-committee [41], hierarchical sampling [42] and so on. The main weaknesses of these approaches are the low quality of data structure. To solve this problem, methods combing informativeness and representativeness measures are designed for finding the optimal query instances. E.g., Donmez *et al.* proposed a dual strategy for active learning that exploits both measures [43], Huang *et al.* utilized model capacity degree to actively adapt a pre-trained model with less labeled examples [44]. However, considering the data storage and concept drift, these off-line algorithms are unsuitable for incremental learning scenario.

## 3 Proposed Method

### 3.1 Notations

In this paper, we solve the problems of incremental deep model training considering concept drift with the semi-supervised streaming data. Specifically, our goal is to learn an adaptive model $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^K$ for the specific task with sequence instances. $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_t, \ldots, \mathcal{D}_T\}$ denotes streaming data with unbounded $T$, where the $t-$th stage comes with training data $\mathcal{D}_t = \{X_t^{t_l}, X_t^{t_u}\}$, in which $X_t^{t_l} = \{\mathbf{x}_{t,i}, \mathbf{y}_{t,i}\}_{i=1}^{N_{t_l}}$ and $X_t^{t_u} = \{\mathbf{x}_{t,j}\}_{j=N_{t_l}+1}^{N_{t_l}+N_{t_u}}$, $N_{t_l}/N_{t_u} (N_{t_l} \ll N_{t_u})$ is the number of labeled/unlabeled data, $N_t = N_{t_l} + N_{t_u}$ is the number of examples in stage $t$. Without any loss of generality, the size of $\mathcal{D}_t$ is set manually, e.g., a fixed time window data, a fixed number of data. $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional instance representation, $\mathbf{y} \in \{0, 1\}^K$, and $K$ is the number of classes. Without any loss of generality, we focus on abrupt drift in this paper.

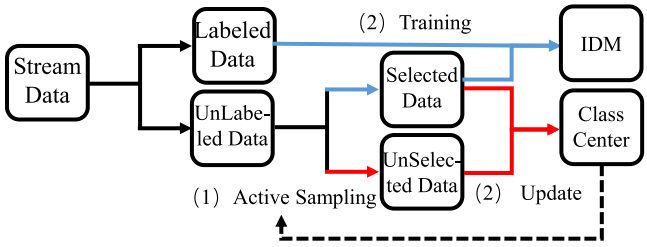



Fig. 3. Illustration of the proposed CE-IDM framework. The framework can be divided into two steps: (1) Active Sampling. The streaming data include labeled and unlabeled data, and the unlabeled data can be further divided into selected and unselected parts by the designed sampling technique. (2) Training and Update. Newly labeled examples and existing labeled examples are used to update the IDM model, and all the unlabeled data is applied to adaptively update the class center representation for using in the next stage.

### 3.2 The Framework and Definition

As shown in Fig. 3, the raw streaming data includes limited amount of labeled data and large number of unlabeled data. The limited amount of labeled data is not enough to train the IDM, so we need to additionally label the unlabeled data. Therefore, CE-IDM is specifically divided into two steps: first, we set a fixed data size to collect streaming data, and adopt incremental active sampling to sample the most discriminative unlabeled instances for labeling. Then, with the newly labeled examples and existing coming labeled data, we incrementally train the deep model (e.g., the IDM approach [45] in the conference version) considering both the scalability and sustainability. Additionally, we define several important definitions used in this paper:

**Definition 1 Incremental Deep Model.** *In the incremental scenarios, the new data arrive sequentially in a stream form, which is not scalable for traditional deep neural networks training that requires the entire training data in prior [27]. Therefore, incremental deep model learns deep networks on the fly in an incremental setting with the streaming data. The main challenge of stream model is the capacity scalability, i.e., how to make the deep model structure scaling with streaming data for flexible model evolution and faster convergence.*

**Definition 2 Classification Task on Streams.** *In the incremental scenarios, the model is dynamically trained, Therefore, classification task on streams aims to train an effective predictor without storing previous data for retraining. The main challenge is the capacity sustainability, i.e., we update the model while preserving previous knowledge for mitigating the catastrophic forgetting caused by the concept drift.*

**Definition 3 Labeling Task on Streams.** *Unlabeled streaming data always require huge manual annotations to train a deep model, which is costly. Thereby, the labeling task on streams adaptively selects the most discriminative newly coming instances to query ground-truths for follow-up training,*

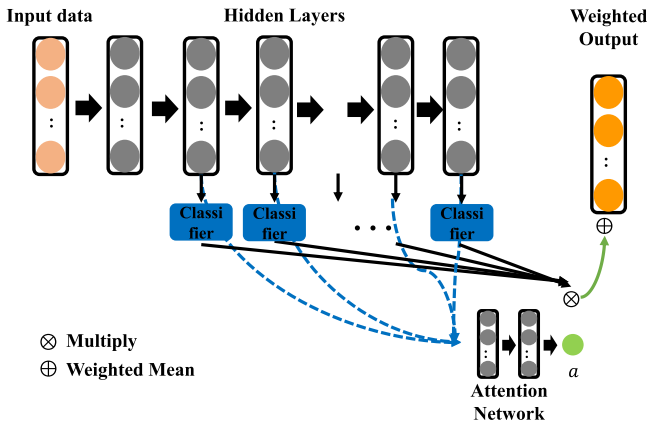



Fig. 4. Illustration of the IDM. Specifically, with the streaming data, we build independent classifiers for the hidden layers, and utilize an extra attention model to calculate the attention weights for final ensemble. Meanwhile, we also integrate the weights with fisher information matrix as the adaptive regularization for relieving forgetting.

*the challenge of incremental active sampling is to design an effective criterion considering the data storage and concept drift.*

These three co-existing concepts also correspond to those proposed challenges: " Capacity Scalability", "Capacity Sustainability", and "Capacity Demand".

### 3.3 Hierarchical Attention Mechanism

Inspired by [27], our method fixes the overall structure of the deep model $f$ in prior, but we adopt an additional attention network to incrementally use the network structure (i.e., the depth $L$) from shallow to deep over time. Specifically, previous deep networks are always designed to optimize the loss function based on the output obtained from the deepest layer. However, in the incremental setting, different depths are suitable for different numbers of instances [27], i.e., shallow network convergences fast, while with restricted learning capacity. Correspondingly, deep network is with larger capacity, yet the learning process converges slowly. Therefore, we fix the overall structure of the network in advance (i.e., the number $L$ and the number of nodes in each layer in Fig. 4 do not change over time), but use the additional attention network to adaptively learn the importance of each hidden layer in the deep network at different time periods, thereby gradually change the network structure from shallow to deep over time.

Without any loss of generality, the deep neural network is with $L$ hidden layers, i.e., fully connected network is with $L$ fully connected layers, CNN is with $L$ hidden blocks, and $L$ is set manually. The attention weight for each hidden layer can be represented as:

$$\alpha_l = g(h_l), \tag{1}$$

where $h_l$ denotes the $l-$th hidden layer feature representations, and $g(\cdot)$ is a shallow neural network (i.e., fully connected network) to calculate the weights for each output of hidden layers, which aims to discover the relationships among hierarchical classifiers. At the end of every round, the weights $\alpha_l$ are normalized as $\sum_l \alpha_l = 1$.

### 3.4 Cost-Effective Active Sampling

In this section, we focus on querying labels for a small batch of instances selected from the incremental unlabeled set, i.e., the unlabeled instances of stage $t$, $X_t^{t_u}$. Informativeness measures the ability of an instance in reducing the uncertainty of a statistical model, and representativeness measures whether an instance well represents the overall input patterns of unlabeled data [16], [23]. After querying the labels, selected instances are combined to update the neural network.

*Representativeness Degree*. Representativeness aims to exploit the evolving of feature transformation pattern, thus instances capturing the property of current distribution should be used. Inspired from [44], we can define the representative degree as the describing capacity of the model learning an instance with the class landmark. In detail, we utilize the embedding $\mathbf{x}_{t,i}^p = \sum_{l=1}^{L} \alpha_l h_l$ as the final representation of an instance $\mathbf{x}_{t,i}$ from $t-$th time. Then we define:

$$D(\mathbf{x}_{t,i}) = \begin{bmatrix} \|\mathbf{x}_{t,i}^p - \bar{\mathbf{x}}_1^t\|^2 \\ \|\mathbf{x}_{t,i}^p - \bar{\mathbf{x}}_2^t\|^2 \\ \cdots \\ \|\mathbf{x}_{t,i}^p - \bar{\mathbf{x}}_K^t\|^2 \end{bmatrix}, \tag{2}$$

where $\bar{\mathbf{x}}_k^t$ represents the $k-$th class center on $t$th stage, $D(\mathbf{x}_{t,i})$ can be regarded as the transformation pattern of $\mathbf{x}_{t,i}$. Then we can also approximate the transformation pattern by a weighted linear combination of class center. Formally, the transformation pattern can be defined as:

$$\hat{D}(\mathbf{x}_{t,i}) = \sum_{k=1}^{K} \beta_k(\mathbf{x}_{t,i}) D(\bar{\mathbf{x}}_k), \tag{3}$$

where $\beta_k(\mathbf{x}_{t,i})$ is the weight corresponding to the $k-$th class. Here we can define it as the prediction of current model, i.e., $\beta_k(\mathbf{x}_{t,i}) = p(k|\mathbf{x}_{t,i})$. $D(\mathbf{x}_{t,i})$ in Eq. (2) indicates the distance between the instance $\mathbf{x}_{t,i}$ and different class centers, and $\hat{D}(\mathbf{x}_{t,i})$ denotes the weighted linear combination of the structure pattern of each representative center by taking all the other centers as landmarks. Therefore, $D(\mathbf{x}_{t,i})$ denotes the observed structure pattern and $\hat{D}(\mathbf{x}_{t,i})$ represents the approximated structure pattern by employing the prediction probability. The difference between these two patterns reflects the potential contribution of the instance with regard to the representation learning, which can be used to estimate the representativeness of instance $\mathbf{x}_{t,i}$. There are various ways to estimate the difference between $D(\mathbf{x}_{t,i})$ and $\hat{D}(\mathbf{x}_{t,i})$. Inspired from [44], the relative rank correlation is more important than the exact value comparison, thereby we employ the Kendall's tau coefficient [46] to estimate the difference. Finally, the definition of representativeness is:

$$R(\mathbf{x}_{t,i}) = \frac{1 - \tau(D(\mathbf{x}_{t,i}), \hat{D}(\mathbf{x}_{t,i}))}{2}. \tag{4}$$

On the other hand, with the streaming data, the class center should be adaptively updated with respect to the coming instances:

$$\bar{\mathbf{x}}_k^{t+1} = \frac{M_{t-1,k}\bar{\mathbf{x}}_k^t + \sum_{j=1}^{|X_t^{t_l}|} \mathbf{x}_{t,j}^p z_{k_j} + \sum_{i=1}^{|X_t^{t_u}|} \mathbf{x}_{t,i}^p \hat{z}_{k_i}}{M_{t-1,k} + |X_{t,k}|} \quad (5)$$
$$\hat{z}_{k_i} = \arg\max_k (p(k|\mathbf{x}_{t,i}^p)),$$

where $M_{t-1,k}$ is the accumulated instance number of $k-$th class. $|\cdot|$ denotes the set size, $z_{k_j}$ is 1 if $j-$th instance belongs to $k-$th class, otherwise is 0, and $\hat{z}_{k_i}$ is the weight of $i-$th unlabeled instance for the class with highest probability.

*Informativeness Degree*. Informativeness always estimates the uncertainty of prediction with the current model. We directly utilize the entropy here, the uncertainty of $\mathbf{x}$ is defined as:

$$U(\mathbf{x}_{t,i}) = -\sum_{k=1}^{K} f_k(\mathbf{x}_{t,i}) log f_k(\mathbf{x}_{t,i}), \quad (6)$$

where $f_k$ is the current model and $f_k(\mathbf{x})$ is the probability of $\mathbf{x}$ belongs to class $k$.

*Trade-Off*. As discussed above, the representativeness measures contribution of an instance to improve representation learning, while the informativeness measures contribution to improve the classifier. To select the most discriminative instances for better adaptation of the network, we should consider these two criteria simultaneously:

$$score(\mathbf{x}_i) = \gamma U(\mathbf{x}_i) + (1-\gamma) R(\mathbf{x}_i), \quad (7)$$

where we should select the instance with higher $score(\mathbf{x})$. Different from traditional active sampling methods that require all the data to calculate the representativeness, CEIDM first represents the instance with weighted hidden layer output considering the representation capabilities of different hidden layers in various periods. And it adaptively updates the class center according to the streaming data without storing the previous data. The details are shown in Algorithm 1.

**Algorithm 1.** The Pseudo Code of Sampling

- **Input**:
- $\mathcal{D}_t = \{X_t^{t_l}, X_t^{t_u}\}$, $X_t^{t_l} = \{(\mathbf{x}_{t,i}, \mathbf{y}_{t,i})_{i=1}^{N_{t_l}}\}$ is the labeled data, $X_t^{t_u} = \{(\mathbf{x}_{t,j})_{j=N_{t_l}+1}^{N_{t_l}+N_{t_u}}\}$ is the unlabeled data.
- Parameter: $\gamma$
- **Output**:
- Queried data from unlabeled data: $X_t^{t_s}$

Receive $\mathcal{D}_t$
**for** $i = N_{t_l}+1 \rightarrow N_{t_l}+N_{t_u}$ **do**
  Calculate the Representativeness degree $R(\mathbf{x}_i) \leftarrow$ Eq. (4);
  Calculate the Informativeness degree $U(\mathbf{x}_i) \leftarrow$ Eq. (6);
  Calculate the $score(\mathbf{x}_i) \leftarrow$ Eq. (7);
**end for**
Select the instances $X_t^{t_s}$ with high score for querying;
Update class center according to Eq. (5).

### 3.5 Evolutive Deep Network

With the newly labeled data $X_t^{t_s}$ and existing labeled data $X_t^{t_l}$, we aim to update the model $f$. Following [27], our method uses an additional attention network to adopt the fixed network structure incrementally from shallow to deep over time. In detail, different from the original network using the final feature representation $h_L$ for prediction, in IDM, as shown in Fig. 4, the final prediction is a weighted combination of outputs learned using middle hidden layer feature representations from $\{h_1, h_2, \ldots, h_L\}$. Following is the prediction function using attention based pooling:

$$f(\mathbf{x}) = \sum_{l=1}^{L} \alpha_l f_l$$
$$f_l = softmax(h_l \Theta_l) \qquad \forall l = 1, 2, \ldots, L \quad (8)$$

where $f_l$ is the classifier using $l$th hidden layer feature representations $h_l$, and $\Theta_l$ is the parameters for $f_l$. At the end of every round, the weights $\alpha = \{\alpha_1, \ldots, \alpha_L\}$ are normalized as $\sum \alpha_l = 1$. Therefore, the loss is:

$$Loss_t(f(\mathbf{x}), \mathbf{y}) = \ell_t\left(\sum_{l=1}^{L} \alpha_l f_l(\mathbf{x}), \mathbf{y}\right), \quad (9)$$

the loss function can be any convex function here, and we utilize the cross-entropy loss for simplicity. During the incremental learning procedure, we need to learn the $g(\cdot), \Theta_l$, and $W_l$, in which $W_l$ is the parameters for learning $h_l$. Different from the original backpropagation, the error derivatives are backpropagated from the last output layer. In Eq. (9), the error derivatives are backpropagated from each classifier $f_l$, i.e., $W_l^{t+1} \leftarrow W_l^t - \eta \nabla_{W_l} \ell_t(\sum_{l=j}^{L} \alpha_l f_l(\mathbf{x}), \mathbf{y})$. We compute the gradient of the final prediction with respect to the parameters of each layer. Note that the summation can be started at $l = j$ in deep network, because the shallower blocks can be regarded as the basic feature extraction. Consequently, with the intuition that shallow models converge faster than deep models [47], the attention mechanism will concentrate on the shallower layers with larger $\alpha_l$ at the initial stage, while with the increase of data, larger $\alpha_l$ is learned for deeper layers, which conforms to the capacity scalability. Consequently, attention mechanism provides an effective approach to learn the optimal network depth automatically in sequence.

### 3.6 Weighted Fisher Regularization

In the model update process, we also aim to mitigate the problem of forgetting the knowledge from previous distributions. Without any loss of generality, as introduced in [13], $F_\theta$ is the "Empirical Fisher Information Matrix" [48], [49] at $\theta$, which can be defined as: $F_\theta = E_{\mathbf{x}\sim\mathcal{D}}\left[\left(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta} \frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta}^\top\right)\right]$, where $\mathcal{D}$ is the instance domain, and $p_\theta(\mathbf{y}|\mathbf{x})$ denotes the prediction. It is notable that log-likelihood $\log p_\theta(\mathbf{y}|\mathbf{x})$ is the same as negative of cross-entropy loss function in deep model for simplicity. Therefore, $F_\theta$ can be seen as the expected loss of gradient covariance matrix.

$F_\theta$ has three key properties [49]: 1) equivalent to the second derivative of the loss near a minimum; 2) can be computed from first-order derivatives alone, and thus is easy for large models; and 3) guarantee to be positive semi-definite. On the other hand, let $D_{KL}(p_\theta \| p_{\theta+\triangle\theta})$ be the KL-divergence [50] between the conditional likelihood of the model at $\theta$ and $\theta + \triangle\theta$, when $\triangle\theta \rightarrow 0$, it can be found that the second-order Taylor approximation of KL-divergence can be written as $D_{KL}(p_\theta \| p_{\theta+\triangle\theta}) \approx \frac{1}{2} \triangle\theta^\top F_\theta \triangle\theta$, which is also equivalent to the computing distance in a Riemannian manifold [51]. Since $F_\theta \in \mathbb{R}^{d_\theta \times d_\theta}$ and $d_\theta$ are usually with millions

for neural networks, it is practically infeasible to store $F_\theta$. To handle this problem, according to [13], we assume parameters to be independent from each other (only using the diagonal parameters of $F_\theta$), which results in the following approximation:

$$Loss = Loss_t + \frac{\lambda}{2}\sum_i F_{\theta_{t-1_i}}(\theta_{t_i} - \theta^*_{t-1_i})^2, \tag{10}$$

where $Loss_t$ is the loss for $t-$th stage only, $\theta_{t_i}$ is the $i-$th entry of $\theta$ at stage $t$, $\theta^*_{t-1_i}$ is the optimal parameters on last stage, and $\lambda$ denotes the trade-off parameter. It is notable that the fisher regularization will keep the important parameters (with large value in $F_\theta$) close to the learned parameters of previous stage. We can optimize the loss to update the model for new distribution data.

We study the setting of abrupt concept drift, in which the distribution of instances will not change drastically in a transitory stage $X_t$, but changes drastically between stages. E.g., users' interest will not change in a short time when following an online news stream, but changes drastically when sensational news appeared. Furthermore, even in more complicated situations, we can adopt the drift detection algorithm to split the data stream into epochs ensuring the smooth of underlying distribution. Thus, we regularize over the conditional likelihood distribution $p_\theta(\mathbf{y}|\mathbf{x})$ of every stage using the fisher information matrix, as Eq. (10), for forgetting measure. Intuitively, $F_\theta$ facilitates the network to learn parameters such that considering both new and previous distributions.

However, Eq. (10) only considers the fisher information matrix of the last stage, and neglects all previous stages. Thus there still exists interval forgetting, the problem can be solved either with multiple separate penalties, or with the sum of quadratic penalties over different stages. While in incremental setting, the network structure evolves with the attention mechanism, in other words, different layers of network weight differently. Similarly, different parts of the fisher information matrix have various importance in sequential stages. Therefore, to incrementally match the posterior distribution of the neural network trained on all stages, we embed the attention weights to the corresponding parameters of fisher regularization, and the adaptive regularization can be represented as following:

$$R = \frac{1}{T}\sum_{t=2}^{T}\sum_i \boldsymbol{\alpha}_{t-1} \odot F_{\theta_{t-1_i}}(\theta_{t_i} - \theta^*_{t-1_i})^2, \tag{11}$$

where $\boldsymbol{\alpha}_t = [\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,L}]^\top$, $\odot$ means multiplying the $\alpha_{t,l}$ to parameters of corresponding layer in the fisher information matrix. This continuous averaging leads to less influence from previous stages.

### 3.7 The Overall Objective

Therefore, we can address the incremental deep model (IDM) in a novel unified framework. IDM ingeniously illustrates both the capacity scalability and sustainability problems in design, i.e., attention based model expansion for capacity scalability, and weighted fisher regularization for capacity sustainability. Combing both Eqs. (9) and (11) comprehensively, the whole loss function can be represented as:

$$Loss = \ell_t\left(\sum_{l=1}^{L}\alpha_l f_l(\mathbf{x}), \mathbf{y}\right) + \frac{\lambda}{T}\sum_{t=1}^{T}\boldsymbol{\alpha}_{t-1} \odot F_{\theta_{t-1}}(\theta_t - \theta^*_{t-1})^2. \tag{12}$$

In summary, IDM adopts an additional shallow network to learn the attention weights for classifiers built by middle hidden layers, then fuses the multiple weighted hidden classifiers for the final prediction. Besides, we embed the learned attention weights to the corresponding elements in the fisher information matrix, which matches the moments of overall posterior distributions in an incremental way. Basically, IDM consists of two modules: 1) Capacity scalability by evolutive deep network: IDM builds the adaptive model with extra attention weights for the hidden layers. Thus, we can exploit the shallow networks at the initial stage, and the deep representations at later stages. 2) Capacity sustainability by weighted fisher regularization: IDM embeds hierarchical attention weights into fisher information matrix of different stages, which aims to match the the posterior distribution on all stages incrementally. The details are presented in Algorithm 2.

**Algorithm 2.** The Pseudo Code of CE-IDM

- **Input**:
- Data Stream: $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_t, \ldots, \mathcal{D}_T\}$, where $\mathcal{D}_t = \{X_t^{t_l}, X_t^{t_u}\}$, $X_t^{t_l} = \{(\mathbf{x}_{t,i}, \mathbf{y}_{t,i})_{i=1}^{N_{t_l}}\}$ is the labeled data, $X_t^{t_u} = \{(\mathbf{x}_{t,j})_{j=N_{t_l}+1}^{N_{t_l}+N_{t_u}}\}$ is the unlabeled data.
- Parameter: $\lambda$, Learning rate parameter: $\eta$
- **Output**:
- Deep network without forgetting: $f$

**for** $t = 1 \rightarrow T$ **do**
  Receive $\mathcal{D}_t$
  Sampling $X_t^{t_s}$ for query according to Algorithm 1;
  **while** stop condition is not triggered **do**
    Calculate the $\boldsymbol{\alpha}_t \leftarrow$ Eq. (8);
    Calculate the $F_{t-1}$;
    Calculate the loss $Loss \leftarrow$ Eq. (12);
    Obtain the derivative $\frac{\partial L}{\partial \boldsymbol{\alpha}_t}, \frac{\partial L}{\partial \boldsymbol{\Theta}_l^t}, \frac{\partial L}{\partial W_l^t}$;
    Update parameters $\boldsymbol{\alpha}, \boldsymbol{\Theta}_l, W_l$;
  **end while**
**end for**

## 4 Experiments and Discussion

In this section, we validate the effectiveness of the proposed CE-IDM approach. We first compare CE-IDM on synthetic and public image datasets as benchmarks, then present the assessment on real-world incremental datasets.

### 4.1 Datasets and Configurations

We first experiment on one synthetic (*Hyperplane* [52]) and two constructed image incremental datasets (*Incremental MNIST* [53], *Incremental CIFAR10* [54]), then give the analysis on three real-world datasets, i.e., action recognition (*Incremental UCF101*) [55], Weather (Wea) [56] and Electricity (Elec) [57]. All the datasets are streaming data with concept drift as [11]. More details of dataset descriptions refer to the supplementary, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/

10.1109/TKDE.2021.3132622. For synthetic and UCF101 datasets, we randomly sample $30\%$ of the examples at each stage for testing, and the remaining part for training. MNIST, CIFAR10, Weather, and Electricity datasets have standard testing sets. Furthermore, for all datasets, 30% of the training data in each stage is used as labeled data and the rest as unlabeled data. Finally, 5 criteria, i.e., average Accuracy, average Precision, average Recall, average F1, and average AUC are taken to measure the performance. E.g., let $acc_{k,j}$ be the accuracy evaluated on the hold-out set of the $j-$th stage ($j \leq k$), after training the network incrementally from stage 1 to $k$, the average accuracy at stage $k$ is defined as: $A_k = \frac{1}{k}\sum_{j=1}^{k} acc_{k,j}$ [14], higher $A_k$ represents for better classifier. Similar method is used to calculate other criteria. We set the number of instances actively selected by each stage as $\{1000, 2000, 3000, 4000, 5000, 6000\}$ and record the corresponding performance. To validate the capacity scalability, we calculate the evolution of parameter $\alpha$. Moreover, to validate the capacity sustainability, we calculate the performance about forgetting profile of different learning algorithms as [14] $\frac{A^* - mean(A)}{A^*}$, $A^*$ is the optimal accuracy with the entire data. $\uparrow$ / $\downarrow$ indicates the larger/smaller the better.

### 4.2 Compared Methods

Considering CE-IDM is related to the deep incremental learning with concept drift and active sampling, we first compare it with several active learning methods to validate the effectiveness: 1) Baseline, 2) Cluster, 3) Random, 4) Margin, and 5) ADMA. Furthermore, with the newly labeled data, we compare our model with several state-of-the-art incremental methods, i.e., DNN-SGD, Adwin-b [58], EFDT [59], and ODLD [27], note that EFDT and ODLD are both incremental ensemble methods. Besides, in our experiments, IDM can be degenerated into catastrophic forgetting setting, therefore, we also compare our method with several modified forgetting methods, i.e., DNN-Base, DNN-L2, DNN-EWC [13], IMM [15], and DEN [38], in which each stage is regarded as a task. More details of comparison methods refer to the supplementary, available online.

In conclusion, we first verify the effectiveness of proposed active sample selection criterion. In this step, we consider IDM as a black box model, operate Eq. (7) and comparison methods to select instances. On the other hand, since previous incremental methods leave the unlabeled data without consideration, we experiment on two settings: 1) using the newly labeled data selected by the best method from three active learning comparison methods (the results can refer to the supplementary, available online); and 2) using the criteria presented in this paper. Consequently, these two settings can validate the effectiveness of CE-IDM by comparing incremental methods with different sampling techniques.

### 4.3 Implementations

The target of CE-IDM is to train the deep model $f$. In detail, we employ CNN architecture, i.e., Resnet [60], for CIFAR10 and UCF101 datasets, and adopt five fully connected network structure as base model for remaining datasets. The images are randomly flipped before passing into the network and no other data augmentation method is utilized. For active sampling, similar to the traditional off-line active learning methods [39], [40], [44], the sample size is a hyper-parameter that needs to be manually set in prior. The parameter $\gamma = 0.8$, and $\lambda$ tuned in $\{1, 10, 100, 1000\}$. The base learning rate is set to 0.001 and optimized with Adam. When the variation between the objective values of Eq. (12) is less than $10^{-5}$ in iterations, we consider CE-IDM converges. We run the following experiments with the implementation of an environment on NVIDIA K80 GPUs server.

### 4.4 Capacity Demand

In this section, we will verify the effectiveness of the proposed cost-effective active sampling. In detail, we conduct two settings of experiments: 1) The number of selected data is fixed as 1000 per stage; and 2) The number of selected data changes in $\{1000, 2000, 3000, 4000, 5000, 6000\}$ per stage. In these two settings, we utilize IDM as black box model, and record the prediction performance after training the model combing queried and labeled data

Table 1 records the performance after selecting 1000 samples per stage. The results reveal that our selected criteria are much better than comparison methods on almost all performance measures, except Average Recall on CIFAR10. This shows that our method can select the most discriminative instance of each stage by considering both representativeness and informativeness.

Moreover, Fig. 5 plots accuracy curves of each stage with the increasing of queried number. Considering the page limitation, we only show the results of CIFAR10 and MNIST datasets. The blue dotted line is the prediction result by labeling full amount of unlabeled instances (i.e., the best performance with all unlabeled data labeled). The results show that the performance of our method increases faster than comparison methods on different stages, this reveals the effectiveness of our proposed selection method. It is notable that active sampling methods (i.e., Margin and ADMA) are better than Random and Cluster methods on most performance measures, and this is because comprehensive consideration of prediction and structure.

### 4.5 Sampling Case Study

We further examine whether the query results that well match discriminative instances should be preferred: 1) Active sampling visualization of current stage. We visualize output feature representation of selected instance in each stage, i.e., $\mathbf{x}^p$, via t-SNE [61]. 2) Active sampling examples of accumulative stages. We visualize output feature representation of the instances sampled so far via t-SNE. Due to page limitation, we only give the case study on CIFAR10 dataset as a representation. Fig. 6 records the t-SNE queried result of CE-IDM and random sampling, in which $(Satge - k)$ denotes the $k-$th stage. Red points represents the queried instances. Obviously, the queried instances of CE-IDM have biased distributions, which are more discriminative than random sampling with a uniform distribution.

We conduct more experiments to analyze the sample bias: we give the visualization of queried instances and previous data. Two diagrams in Fig. 7 record the results of different stages, and validate that the queried instances of CE-IDM have a biased distribution, which are away from the

TABLE 1
Comparison Results of CE-IDM With Other Sampling Methods

| Methods | Average Accuracy ↑ | | | | | | Average Precision ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Cluster | .617 | .810 | .629 | .828 | .719 | .774 | .617 | .810 | .629 | .829 | .712 | .768 |
| Random | .611 | .814 | .632 | .839 | .713 | .762 | .611 | .813 | .631 | .837 | .708 | .755 |
| Margin | .607 | .818 | .632 | .846 | .705 | .769 | .607 | .816 | .633 | .839 | .704 | .762 |
| ADMA | .633 | .815 | .641 | .844 | .726 | .782 | .633 | .813 | .641 | .838 | .719 | .779 |
| CE-IDM | **.666** | **.834** | **.645** | **.882** | **.747** | **.795** | **.666** | **.833** | **.644** | **.877** | **.739** | **.790** |

| Methods | Average Recall ↑ | | | | | | Average F1 ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Cluster | .620 | .838 | .659 | .828 | .720 | .776 | .615 | .801 | .632 | .788 | .719 | .773 |
| Random | .612 | .845 | .657 | .832 | .708 | .765 | .611 | .799 | .633 | .809 | .714 | .765 |
| Margin | .608 | .850 | .653 | .839 | .706 | .770 | .606 | .808 | .636 | .800 | .707 | .769 |
| ADMA | .654 | .844 | **.665** | .813 | .729 | .788 | .631 | .809 | .641 | .820 | .728 | .783 |
| CE-IDM | **.669** | **.854** | .663 | **.883** | **.760** | **.808** | **.665** | **.827** | **.643** | **.862** | **.749** | **.794** |

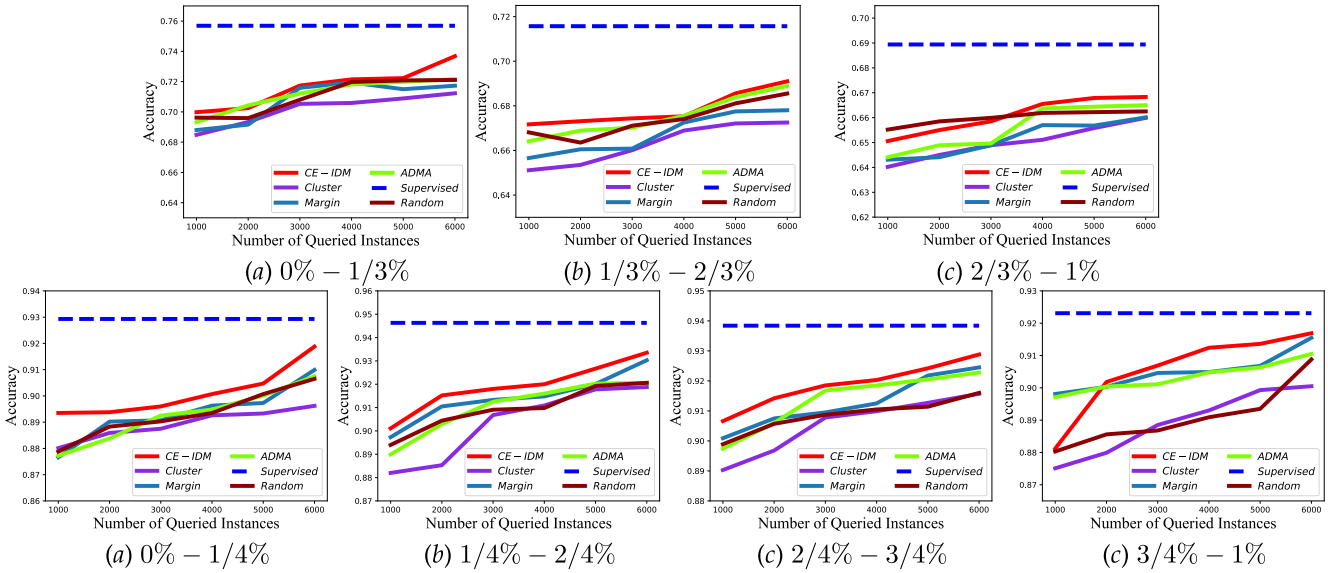
| Methods | Average AUC ↑ | | | | | | Forgetting↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Cluster | .617 | .894 | .794 | .892 | .718 | .776 | .014 | .062 | .068 | .153 | .052 | .064 |
| Random | .611 | .892 | .796 | .899 | .714 | .764 | .010 | .045 | .061 | .124 | .058 | .065 |
| Margin | .607 | .898 | .795 | .904 | .699 | .771 | .011 | .051 | .055 | .117 | .051 | .062 |
| ADMA | .633 | .896 | .801 | .903 | .723 | .781 | .008 | .036 | .060 | .132 | .049 | .057 |
| CE-IDM | **.661** | **.907** | **.802** | **.926** | **.746** | **.803** | **-.002** | **.031** | **.054** | **.095** | **0.44** | **.049** |

previous distribution, so that the incremental model update can more effectively learn the current distribution instances. E.g., in Fig. 7a, the blue points represent the sampled instances in last stage, while the red points denote the sampled instances in current stage, we can find that the blue points are distinct from the red points. On the other hand, sample bias will lead to the catastrophic forgetting problem (i.e., forget the knowledge of previous distribution). In this case, CE-IDM embeds hierarchical attention weights into fisher information matrix of different stages as weighted fisher regularization, which aims to match the the posterior distribution on all stages incrementally.

### 4.6 Performance Measure

In this section, we report the classification performance of all datasets on 5 criteria and forgetting profile in Table 2, which uses CE-IDM sampling. We also add the "oracle", which trains the model with the entire dataset.

From the results, we have the following findings: 1) ODLN achieves better or competitive results considering




(a) $0\% - 1/3\%$ (b) $1/3\% - 2/3\%$ (c) $2/3\% - 1\%$

(a) $0\% - 1/4\%$ (b) $1/4\% - 2/4\%$ (c) $2/4\% - 3/4\%$ (c) $3/4\% - 1\%$

Fig. 5. Accuracy performance of different stages with increasing queried instances. The abscissa denotes query number of instances and the ordinate denotes performance indicator. Each row represents the result of a dataset, from top to bottom: CIFAR10, MNIST.

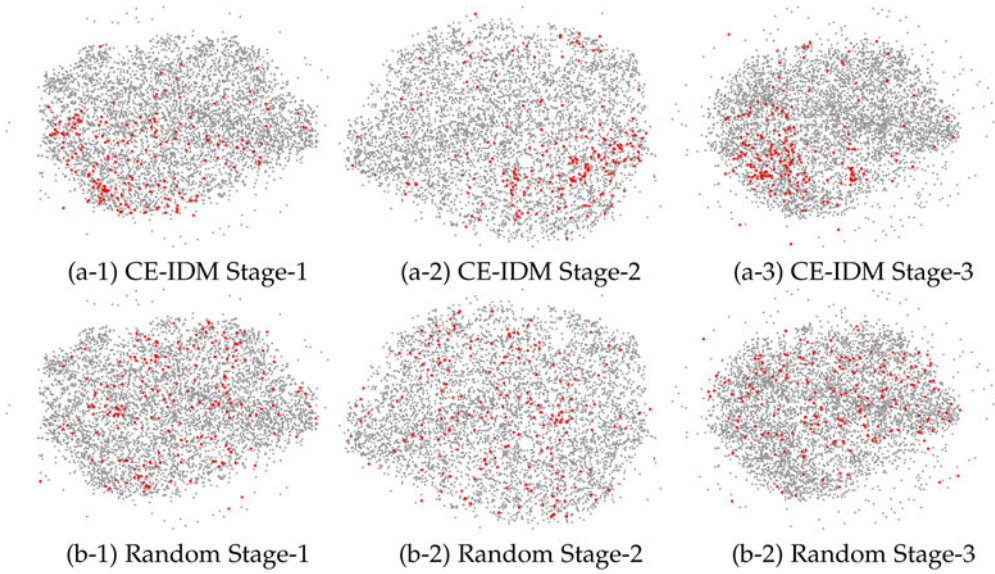



Fig. 6. Visualization of the queried instances of each stage on CIFAR10 dataset.

various criteria on complex datasets, comparing with traditional incremental methods, which validates the effectiveness of evolving structure; 2) The forgetting profile of the incremental task methods, i.e., DNN-EWC, Mean-IMM, Mode-IMM, and DEN, are better than the traditional incremental methods, which verified the effectiveness of mechanism for handling forgetting; 3) Except "oracle", CE-IDM achieves the best performance comparing with other baseline methods on most datasets with various performance measures, which reveals that CE-IDM approach is a high-competitive method handling both the capacity scalability and sustainability challenges.

For a more intuitive measurement of forgetting, we study the degree of forgetting among different models, which defines the forgetting of a particular task as the difference between maximum knowledge gained from that task throughout learning process and we currently have about it, and the smaller difference the better. The results show that CE-IDM has the least forgetting. The $N/A$ in Adwin-b is owning to the incapability of getting the intermediate result in the training process, thereby the forgetting profile cannot be calculated. Negative value in forgetting profile means not only without forgetting, but also has positive influence for the future classification.

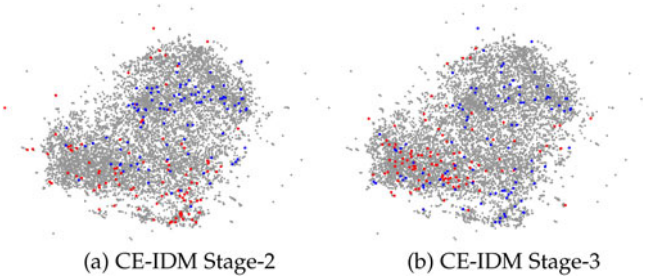



Fig. 7. Visualization of the accumulated queried instances on CIFAR10 dataset.

### 4.7 Ablation Study

To explore the effectiveness of each module in CE-IDM, we compare more variants: 1) w/o R, our active sampling strategy without representativeness degree; 2) w/o U, our active sampling strategy without informativeness degree; 3) Margin+, our active sampling strategy replaces the representativeness degree with traditional margin degree for shallow networks; 4) w/o Update, our active sampling technique without class center update; 5) w/o F, comparing the CE-IDM with baseline without using the regularization; and 6) 'w/o $\alpha$' refers to CE-IDM without the hierarchical attention network. The results in Table 3 reveal that: 1) CE-IDM achieves the superior performance to w/o R and w/o U on all datasets, which indicates the effectiveness by considering both representativeness and informativeness; 2) CE-IDM achieves better performance than Margin+, which validates that our designed representativeness degree can measure the structure better; 3) CE-IDM performs better than w/o Update, which verifies that the concept drift impacts the learning of class center; 4) our proposed method is superior to w/o F on all metrics of the four datasets. The phenomenon indicates that our method can perform classification task with fewer bias; and 5) comparing 'w/o $\alpha$' to CE-IDM, we can find that there is a gap between these performances. We can infer that the attention mechanism utilizes different grained features and predictions, which reflects the data stream evolution of different concepts. With the attention mechanism, CE-IDM is able to depict the most influential prediction layer and give its prediction higher weights. These results verify the effectiveness of hierarchical attention mechanism.

### 4.8 Visualization of Attention Weight

In this section, we evaluate the weight distribution (parameter $\alpha$) of different layers learned by CE-IDM over different stages. We extract data from different stages at intervals of $25\%$, and analyse the mean weight distribution in different stages. Fig. 8 shows the results on MNIST and CIFAR10 datasets. The block 1 in CIFAR10 network is used for the basic feature extraction as mentioned before. The results

TABLE 2
Comparison Results of CE-IDM With Compared Methods Using Our Sampling Method

| Methods | Average Accuracy ↑ | | | | | | Average Precision ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Oracle | .698 | .925 | .688 | .946 | .771 | .827 | .701 | .928 | .682 | .947 | .764 | .829 |
| Adwin-b | .648 | .510 | .577 | .748 | .740 | .763 | .647 | .500 | .576 | .744 | .739 | .761 |
| DNN-SGD | .603 | .830 | .592 | .727 | .645 | .711 | .602 | .829 | .588 | .724 | .626 | .706 |
| EFDT | .590 | .747 | .591 | .677 | .633 | .698 | .588 | .758 | .589 | .678 | .614 | .698 |
| ODLD | .585 | .821 | .581 | .831 | .627 | .742 | .586 | .820 | .578 | .823 | .609 | .741 |
| DNN-Base | .585 | .827 | .614 | .628 | .638 | .685 | .587 | .826 | .600 | .612 | .622 | .692 |
| DNN-L2 | .631 | .819 | .604 | .584 | .676 | .734 | .632 | .818 | .602 | .581 | .667 | .730 |
| DNN-EWC | .641 | .844 | .591 | .839 | .669 | .734 | .638 | .842 | .591 | .837 | .657 | .728 |
| Mean-IMM | .548 | .860 | .611 | .740 | .632 | .689 | **.723** | .855 | .611 | .743 | .616 | .692 |
| Mode-IMM | .632 | .867 | .622 | .761 | .725 | .742 | .633 | .866 | .618 | .758 | .720 | .744 |
| DEN | .591 | .776 | .623 | .785 | .654 | .700 | .592 | .774 | .621 | .785 | .645 | .699 |
| CE-IDM | **.666** | **.871** | **.647** | **.901** | **.759** | **.807** | .666 | **.872** | **.643** | **.902** | **.744** | **.798** |

| Methods | Average Recall ↑ | | | | | | Average F1 ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Oracle | .719 | .917 | .725 | .909 | .782 | .834 | .703 | .922 | .690 | .915 | .772 | .831 |
| Adwin-b | .649 | .619 | .631 | .763 | .745 | .771 | .647 | .458 | .565 | .717 | .741 | .763 |
| DNN-SGD | .621 | .851 | .617 | .750 | .639 | .717 | .560 | .821 | .591 | .666 | .648 | .711 |
| EFDT | .603 | .749 | .616 | .621 | .636 | .701 | .589 | .749 | .589 | .621 | .632 | .700 |
| ODLD | .607 | .865 | .628 | .847 | .630 | .743 | .606 | .812 | .582 | .806 | .629 | .741 |
| DNN-Base | .621 | .866 | .646 | .640 | .641 | .692 | .555 | .823 | .598 | .613 | .635 | .684 |
| DNN-L2 | .637 | .854 | .638 | .504 | .676 | .735 | .628 | .811 | .599 | .522 | .675 | .734 |
| DNN-EWC | .641 | .876 | .636 | .840 | .672 | .738 | .641 | .836 | .580 | .805 | .668 | .735 |
| Mean-IMM | .608 | .879 | .621 | .809 | .633 | .692 | **.691** | .857 | .609 | .731 | .630 | .688 |
| Mode-IMM | .632 | .880 | .638 | .829 | .741 | .743 | .632 | .865 | .619 | .729 | .732 | .744 |
| DEN | .641 | .806 | **.679** | .746 | .662 | .709 | .553 | .766 | .631 | .755 | .654 | .697 |
| CE-IDM | **.668** | **.888** | .678 | **.861** | **.773** | **.815** | .664 | **.869** | **.649** | **.878** | **.759** | **.808** |

| Methods | Average AUC ↑ | | | | | | Forgetting↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| Oracle | .694 | .955 | .829 | .960 | .771 | .829 | - | - | - | - | - | - |
| Adwin-b | .647 | .723 | .765 | .842 | .738 | .762 | N/A | N/A | N/A | N/A | N/A | N/A |
| DNN-SGD | .603 | .905 | .773 | .829 | .650 | .711 | .132 | .107 | .095 | .385 | .177 | .159 |
| EFDT | .591 | .854 | .779 | .798 | .633 | .689 | .108 | .089 | .093 | .296 | .152 | .148 |
| ODLD | .585 | .899 | .767 | .894 | .628 | .737 | .116 | .110 | .100 | .363 | .160 | .132 |
| DNN-Base | .585 | .903 | .785 | .767 | .637 | .687 | .132 | .102 | .089 | .354 | .144 | .136 |
| DNN-L2 | .631 | .899 | .780 | .739 | .675 | .740 | .123 | .077 | .113 | .411 | .121 | .113 |
| DNN-EWC | .641 | .912 | .770 | .899 | .669 | .738 | .069 | .055 | .124 | .175 | .093 | .072 |
| Mean-IMM | .548 | .921 | .783 | .837 | .632 | .686 | .098 | .043 | .099 | .196 | .113 | .101 |
| Mode-IMM | .631 | .925 | .791 | .851 | .738 | .749 | .034 | .051 | .067 | .150 | .089 | .074 |
| DEN | .592 | .875 | .791 | .866 | .654 | .704 | .095 | .074 | .074 | .231 | .105 | .087 |
| CE-IDM | **.666** | **.929** | **.804** | **.938** | **.761** | **.810** | **-.003** | **.029** | **.052** | **.094** | **.032** | **.041** |

reveal that in the initial phase (first stage), the maximum weight locates at the shallow classifier. In the second stage, slightly deeper classifiers have picked up some weights, and in the following stages, deeper classifiers get more weights. Thus, the weight evolution shows that CE-IDM has the ability to perform model selection. Meanwhile, different stages have different depth indicates that CE-IDM learns more discriminative features with more data. In other words, CE-IDM uses the deeper classifiers to learn better features.

### 4.9 Evaluation of Forgetting

Due to page limitation, we report the performance of different models on the first stage for different datasets in the top row of Fig. 9, and the performance of CE-IDM on different stages in the bottom row. In fact, we construct Hyperplane, CIFAR10, UCF101 and MNIST datasets as streaming form with abrupt drift considering different mechanism, i.e., image cropping or adding noise. In result, Hyperplane, CIFAR10 and UCF101 datasets have three stages, and MNIST dataset has four stages. For compared methods, note that DEN utilized the timestamp to save the model of each stage for prediction, whereas the testing data are always unpredictable of the source stage in real applications

TABLE 3
Comparison Results of CE-IDM With Compared Methods Using Our Sampling Method

| Methods | Average Accuracy ↑ | | | | | | Average Precision ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| w/o F | .591 | .790 | .576 | .746 | .634 | .702 | .584 | .790 | .578 | .743 | .621 | .694 |
| w/o R | .627 | .806 | .628 | .834 | .725 | .775 | .626 | .804 | .627 | .833 | .722 | .768 |
| w/o U | .630 | .818 | .633 | .845 | .733 | .783 | .629 | .817 | .632 | .847 | .729 | .772 |
| Margin+ | .644 | .811 | .630 | .839 | .736 | .776 | .639 | .809 | .629 | .836 | .730 | .767 |
| w/o Update | .641 | .822 | .639 | .855 | .739 | .790 | .641 | .823 | .638 | .851 | .734 | .778 |
| w/o $\alpha$ | .646 | .815 | .617 | .845 | .712 | .768 | .645 | .815 | .616 | .840 | .717 | .754 |
| CE-IDM | **.666** | **.834** | **.645** | **.882** | **.747** | **.795** | **.666** | **.833** | **.644** | **.877** | **.739** | **.790** |

| Methods | Average Recall ↑ | | | | | | Average F1 ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| w/o F | .607 | .823 | .628 | .750 | .635 | .717 | .614 | .782 | .579 | .714 | .635 | .711 |
| w/o R | .633 | .835 | .656 | .828 | .735 | .782 | .625 | .798 | .628 | .799 | .727 | .775 |
| w/o U | .649 | .846 | .661 | .832 | .740 | .788 | .631 | .810 | .633 | .815 | .733 | .786 |
| Margin+ | .654 | .840 | .658 | .830 | .742 | .780 | .645 | .803 | .630 | .808 | .738 | .781 |
| w/o Update | .642 | .849 | .660 | .843 | .746 | .801 | .641 | .813 | .636 | .832 | .739 | .789 |
| w/o $\alpha$ | .652 | .830 | .639 | .841 | .718 | .782 | .645 | .805 | .612 | .818 | .713 | .766 |
| CE-IDM | **.669** | **.854** | **.663** | **.883** | **.760** | **.808** | **.665** | **.827** | **.643** | **.862** | **.749** | **.793** |

| Methods | Average AUC ↑ | | | | | | Forgetting↓ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec | Hyperplane | MNIST | CIFAR10 | UCF101 | Wea | Elec |
| w/o F | .591 | .870 | .767 | .809 | .633 | .711 | .122 | .110 | .099 | .358 | .175 | .160 |
| w/o R | .628 | .892 | .790 | .894 | .724 | .774 | .012 | .059 | .066 | .144 | .061 | .059 |
| w/o U | .632 | .896 | .795 | .907 | .731 | .785 | .009 | .051 | .063 | .112 | .053 | .055 |
| Margin+ | .637 | .894 | .791 | .897 | .735 | .779 | .005 | .042 | .062 | .138 | .055 | .058 |
| w/o Update | .641 | .898 | .794 | .909 | .739 | .792 | .006 | .033 | .057 | .107 | .045 | .052 |
| w/o $\alpha$ | .646 | .893 | .778 | .900 | .715 | .759 | .043 | .055 | .072 | .162 | .083 | .070 |
| CE-IDM | **.661** | **.907** | **.802** | **.926** | **.746** | **.803** | **-.002** | **.031** | **.054** | **.095** | **0.44** | **.049** |

under our setting, so we only use the latest model of DEN for testing. The top row reveals that the performance of methods without considering the forgetting regularization (e.g., DNN-SGD, ODLD) steady declines. Meanwhile, CE-IDM shows stable performance on almost all the datasets with slow forgetting, and superior to other fisher regularization based methods with the adaptive attention mechanism. IMM methods need to add multi-task layer for further adjustment after training all stages, which leads to decreasing performance in the early stage (i.e., using SGD for

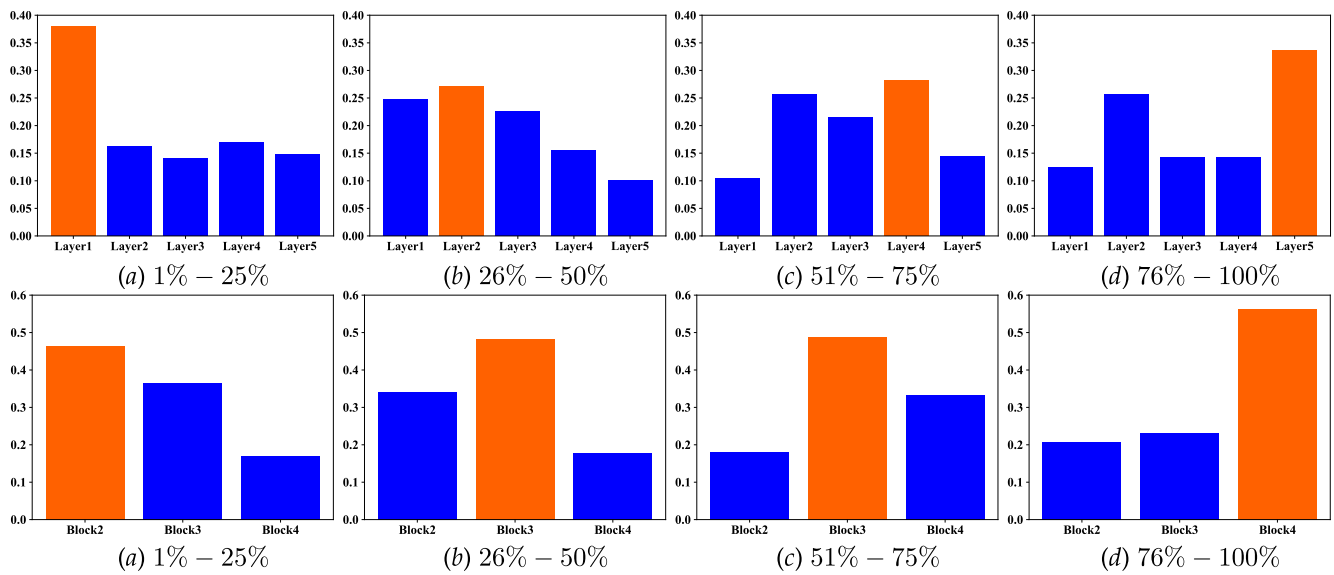



Fig. 8. Evolution of weight distribution over various stages. Top row is MNIST, bottom row is CIFAR10.

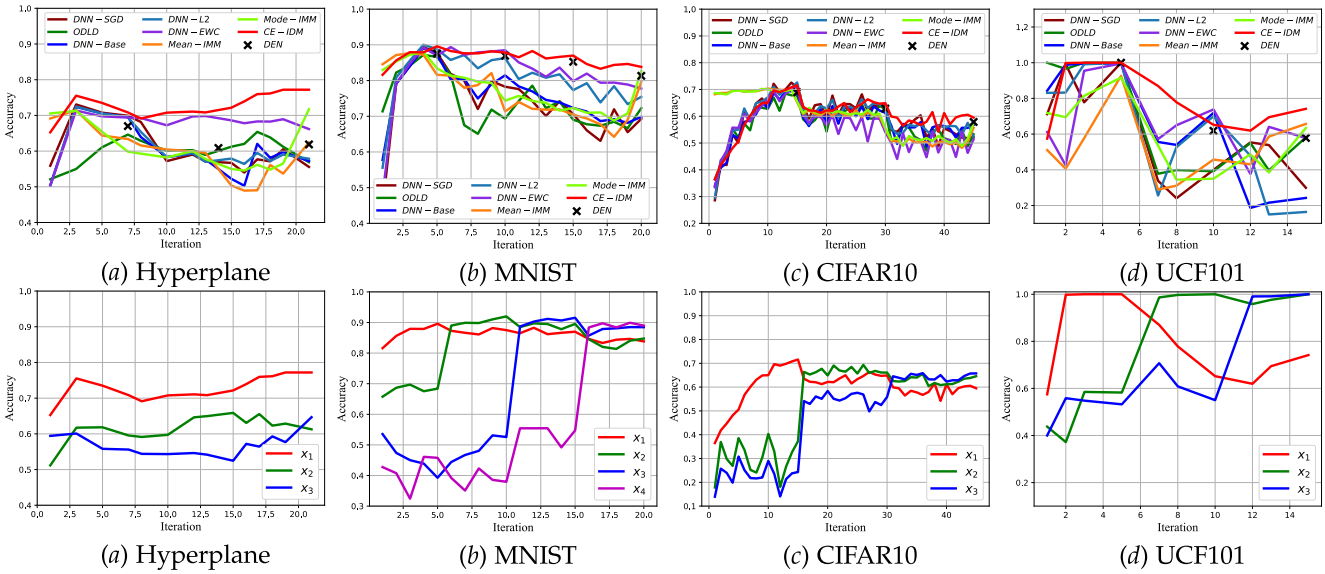



Fig. 9. Accuracy performance of different models and stages. The top row is the results of first stage about different methods over sequential stages, the bottom row is the results of different stages about IDM over sequential stages.

training), and rising performance at the end (last point is the results using fine-tuned IMM methods). The bottom row reveals that, at the transition of different stages, the performance of previous stages will not fall rapidly, which shows that CE-IDM can prevent forgetting efficiently. The background of examples in the first stage of the UCF101 dataset is easy to be classified, thus the initial accuracy is high. As we can infer from Fig. 9a, the accuracy for Mode-IMM and Mean-IMM drastically rises at the last iteration. Because during the whole training process of multiple concepts over the streaming data, IMM uses a Gaussian distribution to approximate the posterior distribution of parameters, and finds the optimal parameter for each task. Mean-IMM and Mode-IMM fuse the optimal parameters into a single model when conducting the final inference process, which is capable of solving all the tasks in streaming data. Consequently, the performance of Mean-IMM and Mode-IMM drastically rises at the last iteration.

### 4.10 Stability of Parameter

In order to explore the influence of parameter $\lambda$, more experiments are conducted. We tune $\lambda$ in $\{1, 10, 100, 1000\}$ and record the average accuracy and forgetting in Fig. 10. Due to the page limitation, we only list two datasets for verification, i.e., MNIST and UCF101. From Fig. 10, we can find that CE-IDM achieves a stable performance on each dataset,

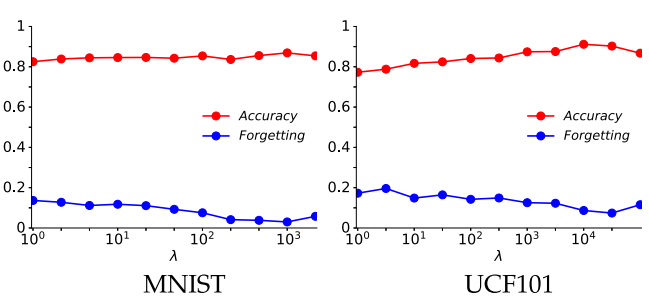



Fig. 10. Influence of the parameter $\lambda$ on MNIST and UCF101 datasets.

which indicates the insensitivity of CE-IDM to parameters. Besides, the regularization is negligible, which would theoretically result in catastrophic forgetting. However, experimentally we observed that this can be circumvented by using a high value ($\approx 10^4$) for the hyperparameter $\lambda$ [13].

## 5 Conclusion

This paper investigated how to develop incremental deep model by labeling the least amount of unlabeled data, which is extended from our preliminary research [45]. Indeed, despite the traditional challenges of capacity scalability and capacity sustainability, there exists another practical problem: the streaming data are unlabeled before learning. It is important to sample most discriminative instances for querying with least cost to update the IDM. Therefore, in this paper, we aim to deal with these three challenges in one unified framework. Along this line, we developed a Cost-Effective Incremental Deep Model (CE-IDM), which has a carefully designed attention model for the hidden layer and novel selection criterion considering both representativeness and informativeness. Moreover, CE-IDM enables capacity scalability by learning deep models with adaptive depth changing from shallow to deep, and has the ability to embedding the attention weights into fisher information matrix, which can incrementally match the posterior distribution of the neural network trained on all stages. Finally, experiments on numerous real-world data showed the effectiveness of CE-IDM for incremental learning.

## Acknowledgments

Yang Yang and Da Wei Zhou contributed equally to this work. We build the model with MindSpore tool [62].

## References

[1] C. Zhang *et al.*, "TrioVecEvent: Embedding-based online local event detection in geo-tagged tweet streams," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 595–604.

[2] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1187–1196.

[3] Y. Tong *et al.*, "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1653–1662.

[4] X. Chen and J. C. S. Lui, "Mining graphlet counts in online social networks," in *Proc. IEEE 16th Int. Conf. Data Mining*, 2016, pp. 71–80.

[5] S. C. H. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," *Mach. Learn.*, vol. 90, no. 2, pp. 289–316, 2013.

[6] Y. Zhu, K. M. Ting, and Z.-H. Zhou, "New class adaptation via instance generation in one-pass class incremental learning," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 1207–1212.

[7] X.-C. Li, D.-C. Zhan, J.-Q. Yang, and Y. Shi, "Deep multiple instance selection," *Sci. China Inf. Sci.*, vol. 64, 2020, Art. no. 130102.

[8] C. Li, Y. Mao, R. Zhang, and J. Huai, "A revisit to mackay algorithm and its application to deep network compression," *Front. Comput. Sci.*, vol. 14, no. 4, 2020, Art. no. 144304.

[9] Y. Yang, Y.-F. Wu, D.-C. Zhan, Z.-B. Liu, and Y. Jiang, "Complex object classification: A multi-modal multi-instance multi-label deep network with optimal transport," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2594–2603.

[10] S. Huang, G. Li, W. Huang, and S. Li, "Incremental multi-label learning with active queries," *J. Comput. Sci. Technol.*, vol. 35, no. 2, pp. 234–246, 2020.

[11] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, 2014.

[12] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychol. Rev.*, vol. 97, no. 2, 1990, Art. no. 285.

[13] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," in *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, 2017, pp. 3521–3526.

[14] A. Chaudhry *et al.*, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 532–547.

[15] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4655–4665.

[16] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1936–1949, Oct. 2014.

[17] Y. Yang, D.-C. Zhan, and Y. Jiang, "Learning by actively querying strong modal features," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2280–2286.

[18] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Joint transfer and batch-mode active learning," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. 253–261.

[19] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowl. Inf. Syst.*, vol. 35, no. 2, pp. 249–283, 2013.

[20] C.-L. Li, C.-S. Ferng, and H.-T. Lin, "Active learning using hint information," *Neural Comput.*, vol. 27, no. 8, pp. 1738–1765, 2015.

[21] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 287–294.

[22] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 1995, pp. 150–157.

[23] Y. Xu, F. Sun, and X. Zhang, "Literature survey of active learning in multimedia annotation and retrieval," in *Proc. 5th Int. Conf. Internet Multimedia Comput. Service*, 2013, pp. 237–242.

[24] A. Budiman, M. I. Fanany, and C. Basaruddin, "Adaptive convolutional ELM for concept drift handling in online stream data," *CoRR*, vol. abs/1610.02348, 2016.

[25] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021.

[26] L. Jun Zhang, T. Yang, R. Jin, Y. Xiao, and Z.-H. Zhou, "Online stochastic linear optimization under one-bit feedback," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 392–401.

[27] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online deep learning: Learning deep neural networks on the fly," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2660–2666.

[28] S.-W. Lee, C. Yeon Lee, D.-H. Kwak, J. Kim, J. Kim, and B.-T. Zhang, "Dual-memory deep learning architectures for lifelong learning of everyday human behaviors," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1669–1675.

[29] G. Zhou, K. Sohn, and H. Lee, "Online incremental feature learning with denoising autoencoders," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1453–1461.

[30] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 23:1–23:36, 2017.

[31] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2000, pp. 487–494.

[32] R. Klinkenberg, "Learning drifting concepts: Example selection versus example weighting," *Intell. Data Anal.*, vol. 8, no. 3, pp. 281–300, 2004.

[33] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 2323–2331.

[34] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[35] A. Gepperth and C. Karaoguz, "A bio-inspired incremental learning architecture for applied perceptual problems," *Cogn. Computation*, vol. 8, no. 5, pp. 924–934, 2016.

[36] R. Coop Mishtal, and I. Arel, "Ensemble learning in fixed expansion layer networks for mitigating catastrophic forgetting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1623–1634, Oct. 2013.

[37] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3390–3398.

[38] J. Lee, J. Yoon, E. Yang, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *CoRR*, vol. abs/1708.01547, 2017.

[39] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. 17th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1994, pp. 2–12.

[40] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2001.

[41] J. Vandoni, E. Aldea, and S. L. Hegarat-Mascle, "Evidential query-by-committee active learning for pedestrian detection in high-density crowds," *Int. J. Approx. Reasoning*, vol. 104, pp. 166–184, 2019.

[42] S. Dasgupta and D. J. Hsu, "Hierarchical sampling for active learning," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2008, pp. 208–215.

[43] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *Proc. Eur. Conf. Mach. Learn.*, 2007, pp. 116–127.

[44] S.-J. Huang, J.-W. Zhao, and Z.-Y. Liu, "Cost-effective training of deep CNNs with active model adaptation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1580–1588.

[45] Y. Yang, D.-W. Zhou, D.-C. Zhan, H. Xiong, and Y. Jiang, "Adaptive deep models for incremental learning: Considering capacity scalability and sustainability," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 74–82.

[46] W. R. Knight, "A computer method for calculating kendall's tau with ungrouped data," *J. Amer. Statist. Assoc.*, vol. 61, no. 314, pp. 436–439, 1966.

[47] T. Chen, I. J. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," *CoRR*, vol. abs/1511.05641, 2015.

[48] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[49] R. Pascanu and Y. Bengio, "Revisiting natural gradient for deep networks," 2013, *arXiv:1301.3584*.

[50] R. Viswanathan, "A note on distributed estimation and sufficiency," *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1765–1767, 1993.

[51] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. Berlin, Germany: Springer, 2006.

[52] W. Fan, "Systematic data selection to mine concept-drifting data streams," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 128–137.
[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[54] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
[55] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.
[56] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.
[57] M. Harries, *Splice-2 Comparative Evaluation: Electricity Pricing*, Newport, U.K.: South Wales Univ., 1999.
[58] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proc. IEEE Int. Conf. Data Mining*, 2007, pp. 443–448.
[59] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1953–1962.
[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[61] P. E. Rauber, A. X. Falcao, and A. C. Telea, "Visualizing time-dependent data using dynamic t-SNE," in *Proc. Eurographics / IEEE VGTC Conf. Vis., Short Papers*, 2016, pp. 73–77.
[62] Mindspore. 2020. [Online]. Available: https://www.mindspore.cn/

**Yang Yang** received the PhD degree in computer science from Nanjing University, China, in 2019. At the same year, he became a faculty member with the Nanjing University of Science and Technology, China. He is currently a professor with the School of Computer Science and Engineering. His research interests include primarily in machine learning and data mining, including heterogeneous learning, model reuse, and incremental mining. He has published more than 10 papers in leading international journal/conferences. He serves as PC in leading conferences such as IJCAI, AAAI, ICML, NIPS, etc.

**Da-Wei Zhou** is currently working toward the MSc degree at the National Key Lab for Novel Software Technology, Department of Computer Science & Technology, Nanjing University, China. His research interests include primarily in machine learning and data mining, including multi-modal learning.

**De-Chuan Zhan** received the PhD degree in computer science from Nanjing University, China, in 2010. At the same year, he became a faculty member with the Department of Computer Science and Technology, Nanjing University, China. He is currently a professor with the Department of Computer Science and Technology, Nanjing University. His research interests include mainly in machine learning, data mining and mobile intelligence. He has published more than 20 papers in leading international journal/conferences. He serves as an editorial board member of IDA and IJAPR, and serves as SPC/PC in leading conferences such as IJCAI, AAAI, ICML, NIPS, etc.

**Hui Xiong** (Fellow, IEEE) is currently a full professor with Rutgers, State University of New Jersey, where he received the ICDM-2011 Best Research Paper Award, and the 2017 IEEE ICDM Outstanding Service Award. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published prolifically in refereed journals and conference proceedings (four books, more than 80 journal papers, and more than 100 conference papers). He is a co-editor-in-chief of Encyclopedia of GIS, an associate editor of *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Big Data*, *ACM Transactions on Knowledge Discovery from Data*, and *ACM Transactions on Management Information Systems*. He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for KDD-2012, a Program co-chair for ICDM-2013, a General co-chair for ICDM-2015, and a Program co-chair of the Research Track for KDD-2018. For his outstanding contributions to data mining and mobile computing, he was elected an ACM Distinguished Scientist in 2014.

**Yuan Jiang** received the PhD degree in computer science from Nanjing University, China, in 2004. At the same year, she became a faculty member with the Department of Computer Science & Technology, Nanjing University, China and currently is a professor. She was selected in the Program for New Century Excellent talents in University, Ministry of Education in 2009. Her research interests include mainly in artificial intelligence, machine learning, and data mining. She has published more than 50 papers in leading international/national journals and conferences.

**Jian Yang** (Member, IEEE) received the PhD degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002. In 2003, he was a postdoctoral researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a postdoctoral fellow with Biometrics Centre, The Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a postdoctoral fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey, USA. He is currently a Chang-Jiang professor with the School of Computer Science and Engineering, NUST. He has authored more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 6000 times in the Web of Science and 15,000 times in the Scholar Google. His current research interests include pattern recognition, computer vision, and machine learning. He is a fellow of IAPR. He is currently an associate editor of Pattern Recognition, Pattern Recognition Letters, the *IEEE Transactions On Neural Networks and Learning Systems*, and *Neurocomputing*.