Highlights

Randomized Neural Network with Adaptive Forward Regularization for Online Task-free Class Incremental Learning

Junda Wang, Minghui Hu, Ning Li, Abdulaziz Al-Ali, Ponnuthurai Nagaratnam Suganthan

- To better acclimate OTCIL scenarios, forward knowledge is exploited to reduce regret and deliver efficient decision-making for ensemble Randomized NN learning in long task streams. This framework realizes one-pass incremental updates with less loss and superiority over ridge.
- Based on the framework, edRVFL-kF algorithm with adjustable forward regularization is derived, effectively avoiding previous replay and catastrophic forgetting.
- To overcome the intractable tuning and distribution drifting of -kF, we further propose edRVFL-kF-Bayes with ks synchronously self-adapted based on Bayesian learning in non-i.i.d OTCIL streams.
- Extensive experiments were conducted on image datasets and the results were analyzed from multiple views (including 6 metrics, dynamic behaviors, and ablation tests), revealing the outstanding performance of edRVFL-kF-Bayes and robustness even with a large PTM.

Randomized Neural Network with Adaptive Forward Regularization for Online Task-free Class Incremental Learning

Junda Wang[a,b], Minghui Hu[c], Ning Li[a,b], Abdulaziz Al-Ali[d], Ponnuthurai Nagaratnam Suganthan[d,*]

[a]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China*
[b]*Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China*
[c]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798, Singapore*
[d]*Computer Science and Engineering Department, College of Engineering, Qatar University, Doha, P.O. Box 2713, Qatar*

Abstract

Class incremental learning (CIL) requires an agent to learn distinct tasks consecutively with knowledge retention against forgetting. Problems impeding the practical applications of CIL methods are twofold: (1) non-i.i.d batch streams and no boundary prompts to update, known as the harsher online task-free CIL (OTCIL) scenario; (2) CIL methods suffer from memory loss in learning long task streams, as shown in Fig. 1 (a). To achieve efficient decision-making and decrease cumulative regrets during the OTCIL process, a randomized neural network (Randomized NN) with forward regularization (-F) is proposed to resist forgetting and enhance learning performance. This general framework integrates unsupervised knowledge into recursive convex

*Corresponding author

Email addresses: jundawang@sjtu.edu.cn (Junda Wang), e200008@e.ntu.edu.sg (Minghui Hu), ning_li@sjtu.edu.cn (Ning Li), a.alali@qu.edu.qa (Abdulaziz Al-Ali), p.n.suganthan@qu.edu.qa (Ponnuthurai Nagaratnam Suganthan)

[1]This work was supported by the National Natural Science Foundation of China under Grant 62273230 and 62203302, and the State Scholarship Fund of China Scholarship Council under Grant 202206230182. This paper was submitted to an Elsevier journal in Feb. 2025.

optimization, has no learning dissipation, and can outperform the canonical ridge style (-R) in OTCIL. Based on this framework, we derive the algorithm of the ensemble deep random vector functional link network (edRVFL) with adjustable forward regularization (-kF), where k mediates the intensity of the intervention. edRVFL-kF generates one-pass closed-form incremental updates and variable learning rates, effectively avoiding past replay and catastrophic forgetting while achieving superior performance. Moreover, to curb unstable penalties caused by non-i.i.d and mitigate intractable tuning of -kF in OTCIL, we improve it to the plug-and-play edRVFL-kF-Bayes, enabling all hard ks in multiple sub-learners to be self-adaptively determined based on Bayesian learning. Experiments were conducted on 2 image datasets including 6 metrics, dynamic performance, ablation tests, and compatibility, which distinctly validates the efficacy of our OTCIL frameworks with -kF-Bayes and -kF styles.

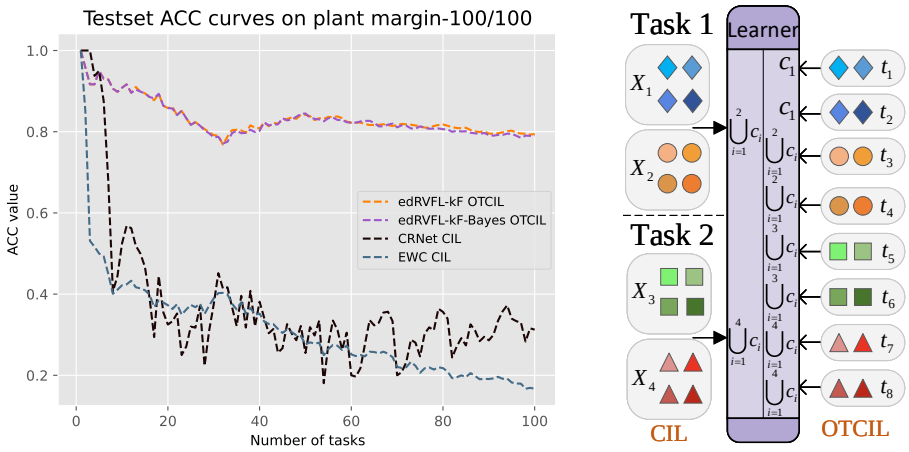
Keywords: Continual learning, Randomized neural networks, Multiple output layers, Random vector functional link, Online task-free class incremental learning

---

1. Introduction

Deep neural networks (DNN) have excelled at various complex tasks such as computer vision and natural language processing [3, 4]. A key premise is that offline data is independently and identically distributed (i.i.d), ensuring the weights gradually converge to a consistent optimum for empirical risk minimization (ERM) in the feasible zone. Unfortunately, issues such as catastrophic forgetting and performance degradation occur if DNN works directly in the continuous learning (CL) context, defined as learning on a sequence of distinct tasks, including never-seen classes [5, 6]. In contrast, human brains can grasp many skills consecutively [7]. The discrepancy motivates researchers to emphasize CL techniques for learners, paving the road for industrial lifelong model deployment and artificial general intelligence (AGI) [1].

The challenging class incremental learning (CIL) scenario in CL does not have a task identity given in the testing, while it is expected to infer predictions in the $\mathcal{Y} = \bigcup_{i=1}^{T} \mathcal{Y}_i$ space, requiring the learner to be adequate for all previous tasks. In general, learners face *challenges* immeasurable to ERM [1, 8] in (1) *non-i.i.d stream*: the continuum of tasks breaks i.i.d for



(a) (b)

Figure 1: (a) The task accuracy of CIL methods on UCI dataset plant margin-100/100. Each task contained one class and replay was forbidden. EWC [1] and CRNet [2] learned in CIL while our edRVFL-kF and edRVFL-kF-Bayes learned in harsher OTCIL without task i.i.d and boundary. Classic methods show serious knowledge forgetting and intractable adjustments in long task streams, while ours achieve immediate response, high efficacy, and less loss. (b) The diagram of CIL and OTCIL. Dashed line denotes task boundary.

T tasks; (2) *catastrophic forgetting*: the offsets of network optima and lacks of correct gradient regularization cause biased weights and ERM failure [9]; (3) *identity absence*: task index is not given in testing, and task order is unpredictable [10]. The data from previous tasks are at most partially or not accessible, hindering ERM and misguiding the global gradient.

Recent research on CIL problems falls into three categories: model-, replay-, and regularization-based methods [10, 11]. However, these approaches are based on some *hypothetical settings* where: (1) the task amount T is small and sample data $|\mathcal{D}_i|$ is large. The non-immediate learner can cost multiple training epochs on each task before responding [8, 12]; (2) the whole dataset of new tasks can be observed once tasks switch, i.e. task-wise locally i.i.d; (3) task boundary demarcating the contents of learning reveals the occasions to update and infer the learner. In these settings, pioneers such as EWC [1] and GEM [8] achieved good results. Unfortunately, some conditions are rarely satisfied in actual real-time scenes such as robotics [13], industrial detection systems [14, 15, 16], and resource dispatches [17], which demand prompt learning, immediate decision-making, and fewer mistakes in data streams.

Although current CIL methods may manage partial *challenges* well, they require repetition training yet still suffer from learning dissipation on long task streams as in Fig. 1 (a), high complexity and storage consumption, and intractable parameter adjustments (see Section 4.4).

To remove the above hypothesis and adapt to more practical scenarios in which task data is streamed online without task-wise i.i.d. or boundary index, our work focuses on the emerging online task-free CIL (OTCIL) scenario [12, 18, 19]. OTCIL concept chart is shown in Fig. 1 (b). It is expected to develop more human-like learners that continually evolve and respond to decisions immediately, because the online learner should not keep waiting until the entire dataset associated with the current task is completed or the boundary is informed before updating itself [20, 18]. Some classical methods like SI [21] and Online EWC [22] move steps forward to OTCIL, as they compute consolidation weights online and analyze batches one by one from local i.i.d tasks [7, 8]. However, they still need task boundaries and cause more regret, so they are not qualified in OTCIL [18].

To achieve favorable decisions about OTCIL tasks in hand, the learner is faced with tough *challenges* which are summarized as (1) *task-wise non-i.i.d*: data comes as a non-stationary batch sequence where concept drifting occurs, instead of i.i.d being held within each task [23]; (2) *catastrophic forgetting*: the learner should balance stability and plasticity between old and new tasks well [24]; (3) *delayed optimization*: Most CIL learners are subjected to time-consuming training epochs and intractable hyper-parameters (HP) as updating relies on gradient descent (GD) of non-convex changing ERM. This laborious process struggles to be completed rapidly before the upcoming task for immediate response [25]. Moreover, continual fitting the ERM leads to a significant decline in accuracy and decision quality on long task streams (e.g. Fig. 1 (a)); (4) *resource consumption*: network volume and memory usage of some CIL methods gradually increase as tasks progress, resulting in subsequent slow learning; (5) *replay on long streams*: task amount T is relatively large and sample amount $|\mathcal{D}_i|$ is small, while the data requires least replay for privacy protection; (6) *other expectations* consist of fast updating on new tasks, no growing structure, and no task boundary [8]. Due to these harsh requirements, previous DNN-based CIL methods were ineffective in OTCIL scenarios. Developing an OTCIL framework that attempts to tackle the above *challenges* is still worthy of a study.

Randomized neural networks (Randomized NN) present an alternative way of deep learning that attracted attention in large-scale computing due to

their effectiveness and efficiency [26, 27, 28]. We employ the well-performed ensemble deep random vector functional link network (edRVFL) as the backbone to quickly learn representations, and promote it to the OTCIL scenario [29, 30]. Practically, the learner is trained on the current task while the next batch is possibly collected or labeled. This inspired us that, from the online learning perspective, if future unsupervised or reserved prior knowledge can be partially utilized to prematurely rectify the GD direction of the learner's trainable parameters, it can enhance precognition of future knowledge, which benefits predictions and further reduces learning regrets in OTCIL.

Forward regularization (-F) can lead to better relative loss bounds than the canonical ridge (-R) in online convex optimization (OCO) [31, 32]. This paper proposes a more general -F with an adjustable k factor to form the -kF style, hoping to adopt controllable unsupervised knowledge to improve posterior updates, reduce predictive loss, and tighten the regret boundary for the learner. The k of -kF coordinates the intensity of the -F penalty, and -kF degenerates into -R when no knowledge is available. Furthermore, we propose -kF-Bayes style with k synchronously self-adapted to overcome the complex distribution drifting and intractable tuning of ks by regarding the -kF incremental updates as a Bayesian learning process during OTCIL. Finally, we present the edRVFL-kF and ready-to-work edRVFL-kF-Bayes algorithms to respond above *challenges*, which update incrementally, retain knowledge well without replay, have better immediate responses, and help the learners escape the dilemmas of plasticity and stability. More excitingly, our algorithms can be made compatible with models with high transferability.

2. Related works

Catastrophic forgetting is the core problem that needs to be addressed in (OT)CIL scenarios, first noted in [9]. Current approaches can be categorized into model-, replay-, and regularization-based methods. OTCIL is a more severe context of CIL, so some CIL methods can still be applied to OTCIL. Algorithms specifically designed for OTCIL are labeled by † in the following.

Model-based methods. This branch of CIL algorithms is mainly based on parameter isolation (PI), knowledge distillation (KD), and architecture augmentation (AA) to enable learning across growing joint subspaces. HAT explicitly optimizes a binary mask to select task-specific nodes and parameters [33]. PCL designs multi-head output for seen tasks and limits the drift of network weights on a new task [34]. DYSON† learns new classes by aligning

the feature space with geometry [19]. PI is restricted by network capacity and is rarely used alone. KD and AA require high computing resources for retraining, and the model scale increases with the task, which is suitable for large models but not for OTCIL without i.i.d and boundary premises. Our work adopts a non-growing or non-transferring model with low resource requirements for OTCIL.

Replay-based method. These CIL algorithms are distinguished by storing partial old representations or generative models to overcome forgetting by retraining while learning new tasks. GEM† uses old samples in the buffer pool to bootstrap the current gradient to escape loss increases [8]. GSS† formulates sample selection as a constraint reduction problem in OTCIL [12]. Although RanPAC is partially similar to ours in pre-trained models (PTM) and embedded random layers, it is very different from the theoretical point of view [35]. iCaRL performs KD on old and new training samples [36]. TRAF alternates between standard and intraprocess updates, which requires large memory [37]. Our work includes no previous replay or generators (pool), and allows used data to be discarded for privacy protection.

Regularization-based methods. This direction is characterized by controlling gradients, which usually collect previous knowledge into explicit regularization terms. A typical implementation is imposing penalties on targets to restrict important weights from wrong changes. The importance can be calculated using the Fisher information matrix (FIM), such as EWC [1]. Online EWC achieves cycles of active learning followed by consolidation in CIL [22]. SI approximates the importance of the parameter by its update length throughout the training trajectory [21]. MAS also adopts a similar approach [38]. MAS† is improved to work in OTCIL with assumptions of task boundary and local i.i.d removed [18]. OWM updates the parameter in the orthogonal directions of the previous input space [39]. The emerging CRNet applies the EWC principle to randomized NN and provides closed-form solutions that enable it to learn streaming data [2]. Our backbone is also a Randomized NN, and previous and future knowledge are all involved in the regularization terms, guiding the learning gradients in OTCIL.

Our work focuses on a general Randomized NN-kF framework in OTCIL of long task streams, realizing on-the-fly decision-making and better performance to -R with the help of self-adaptive forward knowledge intervention. It is also concerned with low updating dissipation and regrets, non-replay and non-growing architectures, and delivering competitive results together with multiple PTMs or fine-tuning techniques on hard image tasks.

3. Preliminary

3.1. CIL scenario

The CIL environment can be described as learning a sequence of Q tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_Q\}$ ($Q \to \infty$ for a lifelong infinite stream). Each task $\mathcal{T}_q = (\mathcal{X}_q, \mathcal{Y}_q)$ and $\mathcal{T}_q = \{(x_q^i, y_q^i)_{i=1}^{|\mathcal{X}_q|} | x_q^i \in \mathbb{R}^s, y_q^i \in \mathbb{R}^{m_q}, 1 \leq q \leq Q\}$, where q is task index, $|\mathcal{X}_q|$ denotes the total number of sample pair (x_q^i, y_q^i) in $\mathcal{T}_q$, and s, $m_q = |\mathcal{Y}_q|$ are feature dimension and space dimension in $\mathcal{T}_q$ respectively. This explains the task-wise local i.i.d and task partition by boundary q. The feature vector x_q^i can denote a tabular example or an image representation. Dataset $\mathcal{T}$ has a total $\sum_{q=1}^{Q} |\mathcal{X}_q|$ samples belonging to $m = |\bigcup_{q=1}^{Q} \mathcal{Y}_q|$ classes.

During training on $\mathcal{T}_q$ ($q > 1$), an expected constraint is the unavailability of previous data $\bigcup_{i=1}^{q-1} \mathcal{T}_i$, i.e. $\bigcap_{i=1}^{q} \mathcal{T}_i = \emptyset$, resulting in catastrophic forgetting on $\mathcal{T}_q$. For CIL, the target (1) is trying to continually minimize the learner $f_q(\theta)$'s ERM on the seen data while keeping the learned $\mathcal{P}(\mathcal{Y}_{1..q-1}|\mathcal{X}_{1..q-1})$ stable.

$$\min_{\theta} \mathcal{L}(f_q(\mathcal{X}_q; \theta), \mathcal{Y}_q) + \mathbb{E}[\mathcal{L}(f_q(\mathcal{X}_{1..(q-1)}; \theta), \mathcal{Y}_{1..(q-1)})] \tag{1}$$

After training on $\mathcal{T}_q$, the learner $f_{q+1}(\theta)$ is requested to enable predictions in all classes encountered in $\bigcup_{i=1}^{q} \mathcal{Y}_i$ without the identifier $1..q$. The objective of (OT)CIL is to train a general learner $f_{Q+1} : \mathcal{X} \to \mathcal{Y}$ that can be incrementally updated, accumulate knowledge over time, resist past forgetting, and make fast decisions on the entire $\mathcal{T}$.

3.2. Offline edRVFL network

As a state-of-the-art Randomized NN, edRVFL stacks multiple RVFL sub-learners, reconnects the original input as residuals, and employs ensemble strategy to improve performance on hard tasks [29]. We first introduce the offline edRVFL design, which will be qualified to work in OTCIL in Section 4.

Given the non-continual edRVFL scheme with L hidden layers, N nodes per layer, the $\mathcal{T}_q$'s dataset $(\mathcal{X}_q, \mathcal{Y}_q)$, and the projection $f_{edRVFL} : \mathcal{X}_q \to \mathcal{Y}_q$, the extracted feature H of the 1-st hidden layer is defined as:

$$H_{1,q} = g_1(\mathcal{X}_q \cdot W_1). \tag{2}$$

For layer $1 < l \leqslant L$ it is defined as:

$$H_{l,q} = g_l([H_{l-1,q}|\mathcal{X}_q] \cdot W_l), \tag{3}$$

where g is an activation function (e.g. ReLU, Sigmoid, Swish), innate randomized weights $W_1 \in \mathbb{R}^{s \cdot N}$ and $W_l \in \mathbb{R}^{(s+N) \cdot N}$, and layer feature $H_{l,q} \in \mathbb{R}^{|\mathcal{X}_q| \cdot N}$. Due to the differentiable convex target with respect to (w.r.t) learnable θ, edRVFL is trained to the optimum in L snapshots without GD therapy:

$$\theta_{l,q+1} = \arg\min_{\theta_l} \|[H_{l,q}|\mathcal{X}_q] \cdot \theta_l - \mathcal{Y}_q\|_F^2 + \lambda_l \cdot \|\theta_l\|_F^2, \tag{4}$$

and the closed-form solution for future prediction:

$$\begin{aligned} primal : \theta_{l,q+1} &= (D_{l,q}^T D_{l,q} + \lambda_l I)^{-1} D_{l,q}^T \mathcal{Y}_q \\ dual : \theta_{l,q+1} &= D_{l,q}^T (D_{l,q} D_{l,q}^T + \lambda_l I)^{-1} \mathcal{Y}_q, \end{aligned} \tag{5}$$

where $D_{l,q} = [H_{l,q}|\mathcal{X}_q]$, $\theta_{l,q+1} \in \mathbb{R}^{(s+N) \cdot |\mathcal{Y}_q|}$ denotes the trainable output weight matrix. Final predictions can be obtained by the ensemble strategy, normally of aggregating averaged or median values of sub-learners for regression and those after SoftMax for classification tasks, which can be denoted as $En(\{D_{l,q}\theta_{l,q+1}\}_{l=1}^L)$. For training, $\mathcal{O}(f_{edRVFL})$ has minima of $\mathcal{O}(L \cdot (N+s)^3)$ and $\mathcal{O}(L \cdot |\mathcal{X}_q|^3)$ time complexity compared to $\inf \mathcal{O}(f_{MLP}) = \mathcal{O}(T_{it} \cdot L \cdot N^2)$ with T_{it} iterations every epoch. In fact, edRVFL always spends much less time as MLP needs multistep gradient computations for each layer. edRVFL has been widely used in industry [40, 41].

3.3. Bregman divergence and OCO problem

We regard the OTCIL of edRVFL as a typical OCO problem, where -F can reduce regret compared to using -R [31]. Relative cumulative regret is defined as the cost difference between an online learner using the -F style and an online expert in hindsight, which is defined as $\mathcal{R}^F = \sum_t \mathcal{L}^F - \min_{\theta \in \Omega} \mathcal{L}^E(\theta)$ [32]. Although recent papers studied multi-column -F, adaptive OCO, and discounted OCO theories, they have not focused on the more general -kF nor on adaptive -kF-Bayes, considering the appropriate penalty intensity and distribution drifting in OTCIL [42, 43, 44].

We describe OCO's (OT)CIL process using Bregman divergence as it can represent penalties and result in general conclusions.

Lemma 1. [31] *Bregman divergence is used to measure relative projection distance between HP sets or distributions. For a real-valued differentiable convex projection* $G : \boldsymbol{\beta} \in \mathbf{B} \to \mathbb{R}$*, Bregman divergence* Δ *is defined as:*

$$\Delta_G(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta}) := G(\tilde{\boldsymbol{\beta}}) - G(\boldsymbol{\beta}) - (\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \cdot \nabla_{\boldsymbol{\beta}} G(\boldsymbol{\beta}), \tag{6}$$

where $\boldsymbol{\beta}$ *is a vector, and* $\nabla_{\boldsymbol{\beta}}$ *denotes the gradient operator w.r.t* $\boldsymbol{\beta}$*.*

Some properties of Bregman divergence are listed in Appendix A. Offline learning can be described as follows:

Lemma 2. [31, 32] *Offline learning refers to the learning process of an expert for global tasks. Assume loss* $\mathcal{L}_q$ *and* $U_0(\boldsymbol{\beta})$ *are differentiable and convex, and there always exists a solution:*

$$\boldsymbol{\beta}_{Q+1} = \arg\min_{\boldsymbol{\beta} \in \mathbf{B}} U_{Q+1}(\boldsymbol{\beta}), \tag{7}$$

where $U_{Q+1}(\boldsymbol{\beta}) = \Delta_{U_0}(\boldsymbol{\beta}, \boldsymbol{\beta}_0) + \mathcal{L}_{1..Q}(\boldsymbol{\beta})$*, subscript 0 denotes prior distributions, and* $\boldsymbol{\beta}_{Q+1}$ *represents the updated parameter after* Q *tasks completed.*

Function U_0 in the Bregman divergence can be a l_2 quadric form of Gaussian distribution: $U_0(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T \eta_0^{-1} \boldsymbol{\beta}$ with symmetric positive definite matrix η_0 for example. (4) shows the separate sub-learner working on one task, and (7) represents the offline expert on the entire dataset $\mathcal{T}$.

Our random network's closed-form update ensures stable adaptation to new classes, reminiscent of the Neural Tangent Kernel (NTK) [45], where wide networks evolve linearly in parameter space. Although our network has finite width, the NTK perspective provides a theoretical lens to interpret its incremental learning behavior [46, 47].

4. Methodology

We follow the OTCIL setup in [18, 19], where some restrictions break the original CIL assumptions. (1) Non-i.i.d data and concept drift in each task. The system observes some randomly drawn samples $(x_q^i, y_q^i)_{i=1}^b$ in one batch. (2) No task boundary. The distribution $\mathcal{P}_q$ could itself undergo sudden or gradual shifts to $\mathcal{P}_{q+1}$ at any moment, and the system is unaware of when these changes in distribution occur. OTCIL is more prone to catastrophic forgetting. Our method targets settings where the total number of tasks is large, namely long task streams, while data arrive in batches.

4.1. Randomized NN with -kF for OTCIL

As the learner is requested to comply with OTCIL setups, we formulate as the non-stationary data stream $\mathcal{T} = \{(X_t, Y_t)_{t=1}^{T} | X_t \in \mathbb{R}^{b \cdot s}, Y_t \in \mathbb{R}^{b \cdot m}, 1 \leq t \leq T\}$, where each batch is isolated by unknown boundaries, m can be the designed load of total classes $|\mathcal{Y}|$, b is the batch size, and $b \cdot T = \sum_1^Q |\mathcal{X}_q|$. This demands the model to learn quickly from the fed data.

During OTCIL for $1 \leq t \leq T$, the X_t is extracted to $\{D_{l,t} = [H_{l,t}|X_t]\}_{l=1,t=1}^{L,T}$ using (2) and (3) due to randomized weights in L sub-learners shared across tasks. With the edRVFL output regarded as multiple continuous sub-learners $\{D_{l,t} \cdot \theta_{l,t+1} \to Y_t\}_{l=1}^{L}$ over T times, for the sake of simplicity, the following analysis is based on the feature stream $(D, Y) = \{(D_{l,t}, Y_t)\}_{l=1,t=1}^{L,T}$.

Theorem 1. *Incremental offline learning using -kF regularization is defined here. For $0 \leqslant t \leqslant T$, the following equations are optimized:*

$$\begin{aligned} \theta_{t+1} &= \arg\min_{\theta} U_{t+1}(\theta) \\ U_{t+1}(\theta) &= \Delta_{U_0}(\theta, \theta_0) + \mathcal{L}_{1..t}(\theta) + k \cdot \hat{\mathcal{L}}_{t+1}(\theta), \end{aligned} \tag{8}$$

where $\hat{\mathcal{L}}$ is the estimated loss generated by current θ on upcoming data (the entire set is not necessary).

The optimization target of -kF is defined as an extra adjustable regularization added to -R. This shows that -kF adopts guidance from future unsupervised knowledge to incur loss and affects the gradients of optimization (also can be seen as a posterior optimum) when updating the present parameters.

Theorem 1 repeatedly retrains on all data after observing t-th data as Lemma 2 does. It is a full replay-based method that requires much retrieval and memory storage for cumulative data, resulting in retrospective retraining. We deduce the non-replay version of -kF for ensemble sub-learners.

Theorem 2. *(OT)CIL framework of Randomized NN using -kF regularization. For $0 \leqslant t \leqslant T$ and $1 \leq l \leq L$, with previous $\mathcal{L}_{l,1..t-1}$ preserved by Bregman divergence $\Delta U_{l,t}$, the framework is achieved by continuous optimization on data streams without replay.*

$$\theta_{l,t+1} = \arg\min_{\theta_l} \Delta_{U_{l,t}}(\theta_l, \theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) + k \cdot \hat{\mathcal{L}}_{l,t+1}(\theta_l) - k \cdot \hat{\mathcal{L}}_{l,t}(\theta_l) \tag{9}$$

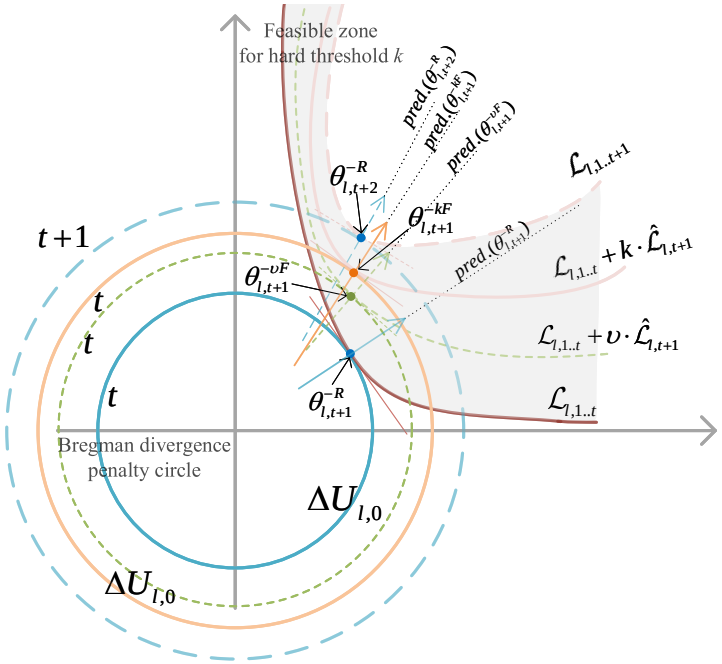


Figure 2: The description to one sub-learner using -R and -kF styles in OTCIL process. The concentric circles denote varied Bregman divergence penalties, and the curves denote losses on data. At time t, the learner using -R (blue solid circle) tries to match $\mathcal{L}_{l,1..t}$ (crimson solid curve) and evolve into $\theta^{-R}_{l,t+1}$ and predict. The learner using optimized -kF (yellow solid circle) tries to meet $\mathcal{L}_{l,1..t}+k\cdot\hat{\mathcal{L}}_{l,t+1}$ (pink solid curve). The -υF ($\upsilon \neq k$) is an instance of non-optimized -kF in k feasible zone (gray area), shown in green dashed style. Then at time $t+1$, the same logic follows for $\theta^{-R}_{l,t+2}$. We recommend -kF as it reduces decision regret on new data compared to -R. k should be carefully picked in the current feasible zone for better performance. -kF is with hard threshold k and we introduce -kF-Bayes with soft adaptive ones later.

Proof: See Appendix B.

The proof also suggests that the effect of optimizing (9) to update the Randomized NN-kF in (OT)CIL is equal to employing a synchronous offline expert (8) using -kF regularization to be trained incrementally on the global data observed so far. (9) can collect foregone knowledge in $\Delta_{U_{l,t}}$ without revisiting previous data, which shows that our methods are non-replay thoroughly.

Remark 1. *There is no learning dissipation for the learner using -kF in (OT)CIL because its optimization remains the same as a simultaneous offline*

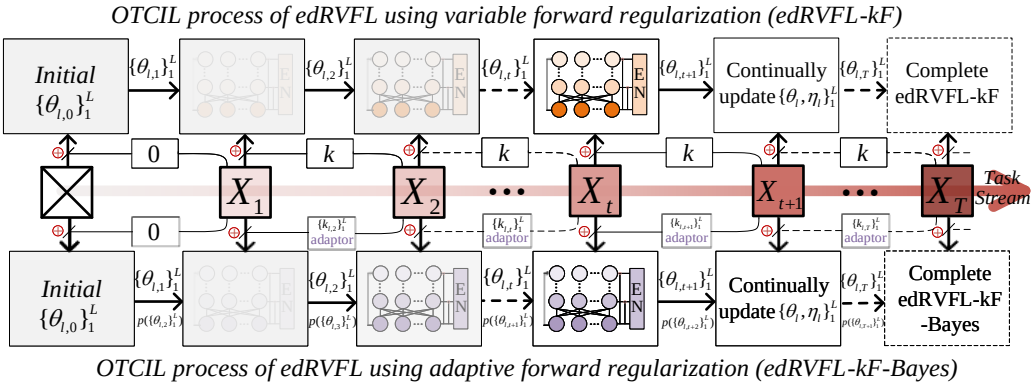


Figure 3: OTCIL processes of edRVFL-kF and edRVFL-kF-Bayes. Task stream over time is shown in a varied rufous arrow. Representations inside edRVFL multiple sub-learners are painted respectively in varied brightness yellow and purple to distinguish two algorithmic styles. Clustered sub-learners are progressively updated and uncertainty is gradually removed (shown by increasing sharpness) to present improving performance as batches come. $\oplus$ denotes unsupervised forward knowledge participates in the current decision, where the k_l is an adjustable constant in -kF and adaptive variable in -kF-Bayes.

one, and non-replay is realized if $\Delta_{U_{l,t}}$ can be reasonably provided.

As stated in [48] and Remark 1, the randomized feature of the past data still achieves completeness and universal approximation because the function clusters $D_{l,t}$ with smooth non-polynomial g maintaining column-wise linear irrelevance in the Hilbert space for $0 \leqslant t \leqslant T$. On the basis of these features, -F achieves improved optima and less relative loss compared to -R by using novel regularization terms in OCO. The experiments in Section 5.4 will exhibit the above interactions.

The estimated penalty on unlabeled data is expected to decorate the gradients and learning rates. We will show the details in Section 4.2. The diagram for one sub-learner using -R and -kF strategies is shown in Fig. 2.

4.2. edRVFL-kF algorithm

We materialize the Theorem 2 and derive the edRVFL-kF algorithm with the multi-dimensional Vovk-Azoury-Warmuth (VAW) form [42].

Theorem 3. *For the OTCIL process of edRVFL-kF, as shown in Fig. 3, step-wise updates of learnable weights follow:*

$$\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(D_{l,t}^T D_{l,t} + kD_{l,t+1}^T D_{l,t+1} - kD_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t] \quad (10)$$

where k is a constant, $\theta_{l,1} = \theta_{l,0} = 0$ for no prior knowledge, and variable learning rates $\eta_{l,t+1} = ((\eta_{l,0})^{-1} + \sum_{i=1}^{t} D_{l,i}^T D_{l,i} + k \cdot D_{l,t+1}^T D_{l,t+1})^{-1}$ with the initial value $\eta_{l,0} = (\lambda_l \cdot I)^{-1}$. The OTCIL process of edRVFL-kF is non-replay and non-growing in architecture and can respond to decisions quickly without forgetting.

Proof: See Appendix C.

Detailed algorithmic procedures of Theorem 3 can be found in **Algorithm 1**, and the process flow chart is shown in Fig. 3. **Algorithm 1** can be configured to be directly applicable to tabular data or representations commonly used in industry.

The iterative weight updating mechanism could be exploited to fine-tune generative models with incremental data (for example, progressively improving deepfake synthesis). However, the reliance of our method on task-bound updates (see experiments in Section 5) inherently limits its direct applicability to generative tasks, as it optimizes for classification rather than data synthesis.

Corollary 1. *The -kF style is more general. If $k = 0$, Theorem 3 degenerates to -R style:*

$$\begin{aligned} \theta_{l,t+1} &= \arg\min_{\theta_l} \Delta_{U_{l,t}}(\theta_l, \theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) \\ \theta_{l,t+1} &= \theta_{l,t} - \eta_{l,t+1}[D_{l,t}^T D_{l,t}\theta_{l,t} - D_{l,t}^T Y_t] \end{aligned} \tag{11}$$

If $k = 1$, Theorem 3 degenerates to -F style:

$$\begin{aligned} \theta_{l,t+1} &= \arg\min_{\theta_l} \Delta_{U_{l,t}}(\theta_l, \theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) + \hat{\mathcal{L}}_{l,t+1}(\theta_l) - \hat{\mathcal{L}}_{l,t}(\theta_l) \\ \theta_{l,t+1} &= \theta_{l,t} - \eta_{l,t+1}[D_{l,t+1}^T D_{l,t+1}\theta_{l,t} - D_{l,t}^T Y_t] \end{aligned} \tag{12}$$

This further suggests that the performance of using -kF must not be inferior to that of -R or -F if k can be properly designed. Such k values exist with

[1]Usually set $\lambda_1 = \lambda_2 \cdots = \lambda_L$ and $k_1 = k_2 \cdots = k_L$ in sub-learners for simplicity (adaptive k scheme in Section 4.3).

[2]Here we consider a severe but more practical condition: the user is completely unaware of any prior knowledge. So, $\eta_{l,2}^{\dagger}$ is set to initial $\eta_{l,0}$ instead of $((\eta_{l,0})^{-1} + D_{l,1}^T D_{l,1})^{-1}$.

[3]A pseudo incomplete $\eta_{l,t+1}$, marked by $\eta_{l,t+1}^{\dagger}$, is maintained as an intermediate variable for computation convenience.

Algorithm 1 edRVFL-kF for OTCIL.

Input: data stream $\mathcal{T} = \{(X_t, Y_t)_{t=1}^T\}$, edRVFL $\{L, N, g_l, \lambda_l\}$, k_l
Output: $\{\theta_{l,T+1}\}_{l=1}^L$

Initialize: $\{W_l\}_{l=1}^L$, $\{\theta_{l,0} = \theta_{l,1} = 0\}_{l=1}^L$, $\eta_{l,0} = (\lambda_l \cdot I)^{-1}$. [1]
OTCIL Procedure:
For $1 \leqslant t \leqslant T$, do:#Assume X_{T+1} accessible.
    Observe (X_{t+1}, Y_t) $\#X_t$ already in-hand.
    For $1 \leqslant l \leqslant L$, do:
        Compute $D_{l,t+1}$ by (2)-(3), and guess with $\theta_{l,t}$.
        If t==1: $\eta_{l,2}^\dagger = \eta_{l,0}$. [2]
        Else: Update $\eta_{l,t+1}^\dagger = \eta_{l,t}^\dagger - \eta_{l,t}^\dagger D_{l,t}^T (I + D_{l,t}\eta_{l,t}^\dagger D_{l,t}^T)^{-1} D_{l,t}\eta_{l,t}^\dagger$ [3].
        Update $\eta_{l,t+1}$ again:
            $\eta_{l,t+1} = \eta_{l,t+1}^\dagger - \eta_{l,t+1}^\dagger k_l D_{l,t+1}^T (I + k_l D_{l,t+1}\eta_{l,t+1}^\dagger D_{l,t+1}^T)^{-1} D_{l,t+1}\eta_{l,t+1}^\dagger$
        Update $\theta_{l,t+1}$:
            $\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(D_{l,t}^T D_{l,t} + k \cdot D_{l,t+1}^T D_{l,t+1} - k \cdot D_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t]$
    End For
    Response on Request: Let $Y_{te} \subseteq \bigcup_1^t Y_i$
    Calculate $\{D_{l,te}\}_{l=1}^L$ by (2)-(3).
    Decisions: $\widehat{Y_{te}} = softmax\{D_{l,te}\theta_{l,t+1}\}_{l=1}^L$
    Accuracy: $acc(En(\widehat{Y_{te}}), Y_{te})$
End For

high probability in practice.

For the classification of offline learning in Lemma 2, penalty factors of diagonal $\eta_{l,0}$ come partially from the secondary moment of Bayesian prior distributions and are usually set to the same constant in L sub-learners. This defaults to the learnable weight components being priorly identically distributed in any sub-learner, regardless of what the feature distribution is or whether the correlations in the weight matrix should be considered.

However, not as expected in Lemma 2 where the entire dataset simultaneously participates in training, the OTCIL process emphasizes CL where separate task data come in batches and the previous is invisible, exacerbates distribution drifting and transition.

Remark 2. *It is unreasonable to keep the penalty factors k_l fixed in Theorem 2 for $0 \leqslant t \leqslant T$ and $1 \leqslant l \leqslant L$. The OTCIL process demands the k_l*

change properly to accommodate variations in feature distributions in time and structural directions.

From the time point of view, the Bayesian prior distribution of the weight matrix in each sub-learner changes with the learning progress in (C.5), where the covariance matrix is symmetric full instead of static diagonal. It is also affected by task batch capacity, varying non-i.i.d distribution, accumulated data volume, etc. From the network architecture point of view, k_l remaining constant in sub-learners is unfair because feature distributions vary between layers. The above facts result in skewing and imbalance in the actual severity of penalties, and impact learning gradients. Unfortunately, optimizing k_l is unrealistic in real-time or based on all observed data, while how to adjust k_l is critical to improving the -kF style.

4.3. Determination of k_t for learners

It is obvious from Theorem 2 and (C.5) that the choice of k is critical, as it determines how much future unsupervised knowledge is involved in each sub-learner. When $k = 0$, the -k F algorithm degrades to -R style in Corollary 1. When $0 < k \leq 1$, it can be regarded as a partial forward knowledge intervention because the ERM on $D_{l,t}$ is larger than the forward penalty of $D_{l,t+1}$. Having $k > 1$ leads to excess in forward regularization.

However, the unchanged k alone cannot sufficiently control the degree of forward knowledge involvement in one sub-learner as the penalty strength on different tasks changes. k should not be set as a constant because the prior distribution of learnable weights keeps changing as OTCIL progresses rather than staying the same as offline learning, and the covariance matrix is no longer diagonal. Remark 2 also suggests that many facts disturb k. To avoid regularization instability changes in OTCIL and rely on costly searching, we propose the edRVFL-kF-Bayes with a self-adaptive strategy of penalty coefficient $k_{l,t}$ to maintain the balance of penalties in the statistical models.

From the Bayesian perspective, (4) and (7) can be represented as:

$$-\ln p(\theta|D_q, \mathcal{Y}_q) \propto -\ln[p(\mathcal{Y}_q|D_q, \theta)p(\theta)] \propto ||D_q\theta - \mathcal{Y}_q||_F^2 + \frac{\varepsilon_0^2}{\varepsilon_1^2}||\theta||_F^2 \quad (13)$$

where ε_0^2 and ε_1^2 are the variance of the residual error and column of the weight matrix, respectively. In fact, (13) implies some assumptions: (1) $d^i\theta - y^i \sim \mathcal{N}(0, \varepsilon_0^2)$ for each sample because of noise i.i.d; (2) it assumes no correlation among weight elements in each θ's vector and $\theta \sim \mathcal{N}(0, diag(\varepsilon_1^2, ...\varepsilon_1^2))$,

although they maybe non-i.i.d; (3) it independently analyzes each element of θ matrix. This paradigm is sound, as the model is trained on the whole dataset. Unfortunately, weight distribution $p(\theta)$ changes dynamically at different periods of OTCIL, which can be described by Bayesian learning:

$$p(\theta_{t+1}|\mathcal{Y}_1, ...\mathcal{Y}_t) \propto \arg\max_{\theta \in \theta_{t+1}} p(\mathcal{Y}_t|\theta, \mathcal{Y}_1, ...\mathcal{Y}_{t-1}) \cdot p(\theta|\mathcal{Y}_1, ...\mathcal{Y}_{t-1}) \tag{14}$$

where the last term denotes a prior distribution concerning the current HP. The prediction loss of posterior distribution on two adjacent batches of data is a decreasing trend, which inhibits the growth of regrets. Our OTCIL process using -kF can be regarded as a regularization-based method that constantly modifies the posterior distribution $p(\theta_{t+1})$ based on available information to minimize ERM without rehearsal. This is reflected in the Bregman divergence $\Delta_{U_{l,t}}(\theta_l, \theta_{l,t})$ in Theorem 2.

We first improve Theorem 2 to the following:

Theorem 4. *To enable the -kF to be self-adaptive in OTCIL, the rigid coefficient k in the recursive optimization target (9) is replaced with soft variables. Let $U_{l,t+1}(\theta_l) = \Delta_{U_{l,0}}(\theta_l, \theta_{l,0}) + \mathcal{L}_{l,1..t}(\theta_l) + k_{l,t+1} \cdot \hat{\mathcal{L}}_{l,t+1}(\theta_l)$, the OTCIL processes can be modeled as follows:*

$$\theta_{l,t+1} = \arg\min_{\theta_l} \Delta_{U_{l,t}}(\theta_l, \theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) + k_{l,t+1}\hat{\mathcal{L}}_{l,t+1}(\theta_l) - k_{l,t}\hat{\mathcal{L}}_{l,t}(\theta_l) \tag{15}$$

Similarly following the steps in Theorem 3, we can obtain:

$$\begin{aligned}
\theta_{l,t+1} = \arg\min_{\theta_l} & \sum\nolimits_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t})^T[(\eta_{l,0})^{-1} \\
& + \sum\nolimits_{j=1}^{t-1} D_{l,j}^T D_{l,j} + k_{l,t} \cdot D_{l,t}^T D_{l,t}](\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t}) \\
& + \frac{1}{2}||D_{l,t}\theta_l - Y_t||_F^2 - \frac{k_{l,t}}{2}||D_{l,t}(\theta_l - \theta_{l,0})||_F^2 \\
& + \frac{k_{l,t+1}}{2}||D_{l,t+1}(\theta_l - \theta_{l,0})||_F^2
\end{aligned} \tag{16}$$

where $(\eta_{l,t})^{-1} = (\eta_{l,0})^{-1} + \sum_{i=1}^{t-1} D_{l,i}^T D_{l,i} + k_{l,t} \cdot D_{l,t}^T D_{l,t}$. The recursive updating policy between $\theta_{l,t}$ and $\theta_{l,t+1}$ can be given correspondingly as follows:

$$\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(k_{l,t+1}D_{l,t+1}^T D_{l,t+1} + (1 - k_{l,t})D_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t] \tag{17}$$

where $(\eta_{l,t+1})^{-1} = (\eta_{l,0})^{-1} + \sum_{i=1}^{t} D_{l,i}^T D_{l,i} + k_{l,t+1} D_{l,t+1}^T D_{l,t+1}$.

For computational convenience, our algorithm assumes column-wise independence in θ_l and sets the shared variance η_l within the columns. By redefining the Bregman projection, one can also set $\{\eta_{l,i,0}\}_{i=1}^{m} \in \mathbb{R}^{m\cdot(s+N)^2}$ or $\eta_{l,0} \in \mathbb{R}^{[m\cdot(s+N)]^2}$ for joint correlation modeling of the Bayesian prior distribution.

Here we still maintain the previous setup and analyze the distribution model column-wise $\theta_{l,t}$ in OTCIL. According to the a posterior computation in (16), the Bayesian prior distribution for the current batch has changed to $p(\theta_{l_i,t+1}) \sim \mathcal{N}(\theta_{l_i,t}, \eta_{l,t})$, so that the estimated distribution $p(D_{l,t+1}\theta_{l_i,t+1}) \sim \mathcal{N}(D_{l,t+1}\theta_{l_i,t}, D_{l,t+1}\eta_{l,t}D_{l,t+1}^T)$ and $p(D_{l,t}\theta_{l_i,t+1}) \sim \mathcal{N}(D_{l,t}\theta_{l_i,t}, D_{l,t}\eta_{l,t}D_{l,t}^T)$. This is because the last two terms can be regarded as a Bayesian prior distribution w.r.t. $D_{l,t+1}\theta_{l,t+1}$ and $D_{l,t}\theta_{l,t+1}$ respectively. The $k_{l,t+1}$ and $k_{l,t}$ can be set to:

$$\begin{aligned} k_{l,t+1} &= \left(\frac{Tr[(D_{l,t+1}\eta_{l,t}D_{l,t+1}^T)^{-1}]}{b}\right)^{-1} \\ k_{l,t} &= \left(\frac{Tr[(D_{l,t}\eta_{l,t}D_{l,t}^T)^{-1}]}{b}\right)^{-1} \end{aligned} \tag{18}$$

$k_{l,t} = (random\_pick(D_{l,t}\eta_{l,t}D_{l,t}^T)^{-1})^{-1}$ or $k_{l,t} = Tr[(D_{l,t}\eta_{l,t}D_{l,t}^T)]/b$ could be another two choices for quick computation. To avoid the ill-posed problem and allow manual adjustment, (18) is calculated by Moore-Penrose as follows:

$$\begin{aligned} k_{l,t+1} &= \kappa \cdot \left(\frac{Tr[(D_{l,t+1}\eta_{l,t}D_{l,t+1}^T + \sigma I)^{-1}]}{b}\right)^{-1} \\ k_{l,t} &= \kappa \cdot \left(\frac{Tr[(D_{l,t}\eta_{l,t}D_{l,t}^T + \sigma I)^{-1}]}{b}\right)^{-1} \end{aligned} \tag{19}$$

where σ is a small positive number (e.g. 10^{-5}) and κ is a positive constant (e.g. 1.0). edRVFL-kF-Bayes algorithm can be found in **Algorithm 2**, and the process flow chart is shown in Fig. 3. **Algorithm 2** can also be applied to tabular data or other representations in the industry, and k no longer has to be manually set. The new HP (i.e., κ, σ) are less sensitive and affect performance only to a limited extent, so this algorithm can be used in a *plug-and-play* fashion.

Algorithm 2 edRVFL-kF-Bayes for OTCIL.

Input: data stream $\mathcal{T} = \{(X_t, Y_t)_{t=1}^T\}$, edRVFL $\{L, N, g_l, \lambda_l\}$
Output: $\{\theta_{l,T+1}\}_{l=1}^L$

Initialize: $\{W_l\}_{l=1}^L$, $\{\theta_{l,0} = \theta_{l,1} = 0\}_{l=1}^L$, $\eta_{l,0} = (\lambda_l \cdot I)^{-1}$.
OTCIL Procedure:
For $1 \leqslant t \leqslant T$, do:#Assume X_{T+1} accessible.
  Observe (X_{t+1}, Y_t) #X_t already in-hand.
  For $1 \leqslant l \leqslant L$, do:
    Compute $D_{l,t+1}$ by (2)-(3), and guess with $\theta_{l,t}$.
    If t==1:
      $\eta_{l,1}^\dagger = \eta_{l,0}$.
      Get $k_{l,t+1}(\eta_{l,1}^\dagger)$ and $k_{l,t}(\eta_{l,1}^\dagger)$ by (19)
    Else:
      Update $\eta_{l,t}^\dagger = \eta_{l,t-1}^\dagger - \eta_{l,t-1}^\dagger D_{l,t}^T (I + D_{l,t}\eta_{l,t-1}^\dagger D_{l,t}^T)^{-1} D_{l,t}\eta_{l,t-1}^\dagger$.
      Get $k_{l,t+1}(\eta_{l,t})$ and $k_{l,t}(\eta_{l,t})$ by (19)
    Update $\eta_{l,t+1}$:
      $\eta_{l,t+1} = \eta_{l,t}^\dagger - \eta_{l,t}^\dagger k_{l,t+1} D_{l,t+1}^T (I + k_{l,t+1} D_{l,t+1}\eta_{l,t}^\dagger D_{l,t+1}^T)^{-1} D_{l,t+1}\eta_{l,t}^\dagger$
    Update $\theta_{l,t+1}$:
      $\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(k_{l,t+1} D_{l,t+1}^T D_{l,t+1} + (1-k_{l,t}) D_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t]$
  End For
  Response on Request: Let $Y_{te} \subseteq \bigcup_1^t Y_i$
  Calculate $\{D_{l,te}\}_{l=1}^L$ by (2)-(3).
  Decisions: $\widehat{Y_{te}} = softmax\{D_{l,te}\theta_{l,t+1}\}_{l=1}^L$
  Accuracy: $acc(En(\widehat{Y_{te}}), Y_{te})$
End For

4.4. Comparisons with CRNet

The novel CRNet proposed in [2] is a strong competitor to ours, which is based on the EWC principle and protects the performance of previous tasks by constraining current optimized weights within a low-error region using consolidation factors in Randomized NN. Specifically, it adopts the Laplacian approximation to match a posterior distribution $p(\theta_{t+1}|\mathcal{Y}_1, ...\mathcal{Y}_{t-1})$ (assumed by the Gaussian distribution) at the latest optimized weights. In this way, Gaussian covariance of the regularization term is estimated by:

$$\mathbb{E}(\sigma^{-2}) = -\mathbb{E}(\nabla_\theta^2 \log p(\theta|\mathcal{Y}_{t-1})) = \mathbb{E}(\nabla_\theta \log p(\theta|\mathcal{Y}_{t-1})^2) \tag{20}$$

and consolidation factors are given by the empirical Fisher information matrix (EFIM) finally.

Although the CRNet is also based on Randomized NN and updated one-shot using closed-form solution, it has some obvious differentiae compared to our methods such as: (1) only for CIL scenario with task-wise i.i.d. and boundary but not working in OTCIL; (2) trade-off λ_q measuring task relative importance greatly affects performance, while properly setting them is hard, considering of large Q and other HP; (3) although it uses closed-form updates, the numerical fitting loss of a posterior $p(\theta|\mathcal{Y})$ with EFIM and its Taylor series exists, as shown in Fig. 1, which increases as more tasks come. Our proposed methods focus on more practical OTCIL with long task stream and batch non-i.i.d premises, enable crucial adaptive factors, and maintain a posterior $p(\theta_{t+1}|\mathcal{Y}_1, ...\mathcal{Y}_{t-1})$ in Bregman divergence.

5. Experiments

This section documents experimental results on public image datasets. We used multiple measurements to assess the static and dynamic performance of our methods and compare them with SOTA baselines. We also explored the efficacy of edRVFL-kF and edRVFL-kF-Bayes in different scenarios and further studied ablation testing.

5.1. Implementation details

(1) **Evaluation metrics**: An in-depth analysis of the experimental results was performed by introducing six metrics that characterized the accuracy of the immediate test set, the accuracy of incremental tasks, the ability to retain knowledge, the degree of knowledge loss, the immediate regret and the Kullback-Leibler divergence (KL).

Average task accuracy (ACC) is defined in CL literature as the average accuracy of all previously learned tasks.

$$ACC = \frac{1}{|Q|} \sum\nolimits_{q=1}^{Q} R_{Q,q} \tag{21}$$

where $R_{Q,q}$ is the classification accuracy of the learner on task q after learning on task $Q(Q \geq q)$. It reflects the task-wise accuracy variation in (OT)CIL processes.

Backward transfer (BWT) is defined as the average difference between the accuracy of all task completions and the first learning of one task:

$$BWT = \frac{1}{|Q|-1}\sum_{q=1}^{Q-1} R_{Q,q} - R_{q,q} \tag{22}$$

BWT indicates the knowledge retention ability of the algorithm when larger values are desired.

Forward transfer (FWT) is defined as the average difference between the accuracy of first learning of one task and using an independent expert on it:

$$FWT = \frac{1}{|Q|-1}\sum_{q=2}^{Q} R_{q,q} - R_{q}^{ind} \tag{23}$$

where R_q^{ind} denotes the testing accuracy of an independent expert trained only on task q. Higher FWT indicates that the learner can acquire more knowledge from newly seen tasks.

Immediate accuracy ($acc.(t)$) denotes the immediate testing accuracy on the entire testset after the learner finishes $t-th$ learning on $\mathcal{T}$ in OTCIL.

$$acc.(t) = R_{t,1..Q} \quad 1 \leq t \leq T \tag{24}$$

It is used to study the dynamic learning performance of using -kF and -kF-Bayes styles precisely.

Immediate regret ($regret(t)$) is a common metric in online stream learning and is defined as the real-time cost incurred by the learner:

$$regret(t) = ||\frac{\sum_{l=1}^{L} soft\max(D_{l,te}\theta_{l,t}) - L \cdot Y_{te}}{L \cdot |\mathcal{X}_{te}|}||_F^2 \tag{25}$$

where $1 \leq t \leq T$ and the subscript denotes Frobenius norm. We also offer a cumulative regret to describe the total loss incurred on the testset in the OTCIL process.

Immediate KL ($KL(t)$) is another metric for measuring the learner's loss in OTCIL from the perspective of polynomial distribution.

$$KL(t) = \frac{\sum_m Y_{te} \odot \ln(\frac{L \cdot Y_{te}}{\sum_{l=1}^{L} soft\max(D_{l,te}\theta_{l,t})})}{|\mathcal{X}_{te}|} \tag{26}$$

where $1 \leq t \leq T$. Algorithms that incurred lower losses in OTCIL are thought to perform better. The two indices are used to study the dynamic and entire performance in OTCIL.

(2) **Dataset**: To verify the proposed framework of edRVFL-kF and study the efficacy of edRVFL-kF-Bayes in OTCIL scenarios, we conducted experiments on Fashion-MNIST[4] and CIFAR-100[5]. The terminology [DATASET]-$|\mathcal{Y}|/\mathcal{Q}$ denotes the $\mathcal{T}$ with $|\mathcal{Y}|$ classes is divided into Q tasks, namely m/Q classes in each task [2]. In addition, $\mathcal{T}$ could be split into batches to learn as our proposed methods worked without task boundaries. Data chunks of tasks were sequentially fed to the learner and the previous revisit was not allowed except for replay-based methods.

Fewer tasks, such as FashionMNIST-10/5 and CIFAR-100/5, tended to examine the learner's ability to learn difficult tasks continually, while long task streams paid attention to knowledge retention like FashionMNIST-10/10 and CIFAR-100/100. Implementation details are recorded in subsections.

(3) **Compared methods**: The compared methods included model-, replay- and regularization-based approaches. In each experimental part, classical and latest methods, such as EWC (2017), CRNet (2023), and NICE (2024), would be involved. We also added specially designed methods for OTCIL, such as GEM (2017) and DYSON (2024). We used official open-source codes or popular third-party codes for all methods.

5.2. FashionMNIST dataset

For the FashionMNIST-10/5, each task containing 2 classes of approximate difficulty was sequentially learned for CIL methods, and the number of tasks would be extended to more for OTCIL. We carried out experiments on a wide range of methods, including classical (e.g. EWC [1], MAS [18], PCL [34], GEM [8]) and SOTA algorithms (e.g. CRNet [2], RanPAC [35], NICE [49], CGAN+ResNet-50 [50]). The backbone architectures and HP selection followed the setups in [2], or adhered to open source codes in the original papers (see the footnotes in Table 1 for details). The task order was randomly sorted and baselines were tested 10 times for consistent results. Only the current task/batch data were present, and the task boundary was merely revealed to the CIL methods. Our methods still learned tasks using one-pass and without revisits, and had no prior update before incoming data.

[4]https://www.kaggle.com/datasets/zalando-research/fashionmnist

[5]https://www.cs.toronto.edu/~kriz/cifar.html

Table 1: Testset accuracy comparisons of CIL algorithms on FashionMNIST

Category	Algorithm	Task batch sum.	Metric		
			$ACC \pm std.$	$BWT \pm std.$	$FWT \pm std.$
Non-CL	offline edRVFL[29]	1	0.9876±1.09e-4	-	-
	separate edRVFL	1	0.9899±0.0043	-	-
	fine-tuning edRVFL	1	0.2655±0.0202	-	-
	non-incremental edRVFL	1	0.1974±0.0030	-	-
Regularization	EWC*[1]	5	0.4021±0.0422	-0.6550±0.0798	-0.0596±0.0296
	SI*[21]	5	0.4151±0.0424	-0.6525±0.0608	-0.0379±0.0329
	MAS*†[18]	5	0.3883±0.0258	-0.7077±0.1074	-0.0893±0.0395
		10	0.3564±0.0237	-0.7235±0.0857	-0.1023±0.0284
	Online EWC*[22]	5	0.4535±0.0706	-0.5808±0.0811	-0.0592±0.0438
	DMC*[51]	5	0.7081±0.0229	-0.3327±0.3748	-0.2793±0.0478
	OWM*[39]	5	0.7931±0.0103	-0.1791±0.0230	-0.0670±0.0119
	CRNet-I*◇[2]	5	0.9472±0.0075	-0.0328±0.0086	-0.0184±0.0025
	CRNet-II*[2]	5	0.9205±0.0054	-0.0534±0.0071	-0.0299±0.0034
	IF2NET◇[52]	5	0.9501±0.0047	-0.0189±0.0074	-0.0267±0.0112
Model	EFT*[53]	5	0.7826±0.0205	-0.1571±0.0624	-0.0598±0.0234
	PCL*[34]	5	0.8327±0.0049	-0.1258±0.0301	-0.0681±0.0131
	PathNet*[54]	5	0.6957±0.0482	-	-
	HAT*[33]	5	0.7092±0.0184	-	-
	CAT*[55]	5	0.7774±0.0410	-	-
	FS-DGPM[56]	5	0.8245±0.0055	-0.1471±0.0237	-0.1340±0.0220
	RanPAC[35]	5	0.8745±0.0353	-0.0526±0.0245	-0.0423±0.0065
	RanDumb◇[57]	5	0.9364±0.0130	-0.0456±0.0083	-0.0303±0.0052
	NICE[49]	5	0.7538±0.0263	-0.3735±0.0412	-0.0324±0.0185
	DYSON†[19]	5	0.8333±0.0192	-0.1300±0.0217	-0.0196±0.0211
		10	0.7950±0.0232	-0.1547±0.0225	-0.0357±0.0228
	AUTOACTIVATOR◇[58]	5	0.8846±0.0006	$\overline{-0.0850 \pm 0.0750}$	-
Replay	RPS-Net*[59]	5	0.7984±0.0192	-0.0228±0.0131	-0.0326±0.0412
	GEM*†[8]	5	0.8282±0.0107	-0.0689±0.0392	-0.0915±0.0399
		10	0.7526±0.0152	-0.0935±0.0238	-0.1103±0.0343
	LOGD*[60]	5	0.8316±0.0191	-0.1001±0.0077	-0.0546±0.0215
	IL2M*[61]	5	0.8687±0.0235	-0.0604±0.0313	-0.0166±0.0028
	GSS†[12]	5	0.8636±0.0523	-0.0435±0.0157	-0.0201±0.0088
		10	0.8246±0.0360	-0.0634±0.0133	-0.0364±0.0122
	ARI[62]	5	0.8376±0.0950	-0.0871±0.0238	-0.0653±0.0254
	DER++[63]	5	0.7157±0.0233	-0.2609±0.0523	-0.0746±0.0203
	X-DER[64]	5	0.8345±0.0126	-0.0935±0.0075	-0.0398±0.0124
	CVAE+ResNet-50◇[50]	5	$\overline{0.9750 \pm \mathbf{0.0000}}$	$\overline{-0.0250}$	-
	CGAN+ResNet-50◇[50]	5	$\overline{0.9350 \pm 0.0050}$	$\overline{-0.1050}$	-
Ours	edRVFL-R†	5	0.4136±0.0891	-0.0067±0.0336	-0.6008±0.1243
		10	0.5789±0.1427	0.0883±0.1183	-0.4645±0.1639
		20	0.9358±0.0267	0.0895±0.0838	-0.1180±0.0747
	edRVFL-kF†◇	5	0.9787±0.0145	0.3206±0.1144	-0.3389±0.1250
		10	0.9867±0.0002	0.0064±**0.0036**	-0.0065±0.0027
		20	0.9876±**7.07e-5**	0.0077±0.0110	**-0.0062**±0.0024
	edRVFL-kF-Bayes†◇	5	0.9804±0.0066	**0.3581**±0.0437	-0.4130±0.0475
		10	**0.9879**±0.0008	0.0063±**0.0036**	-0.0066±0.0028
		20	0.9872±2.24e-4	0.0108±0.0093	-0.0066±**0.0023**

Note: the † denotes this algorithm can be applied to OTCIL scenario, so that we add additional experiments on varied task stream length; algorithm with * denotes the setups follow in [2], and without * means using recommendations; ◇ signifies the best top-2 benchmarks in the groups; higher metrics with lower *std.* is better; the best value of each metric is shown in **bold**; the underline and *overline* indicate the upper and lower bound respectively; the HP of our proposed methods can be found in Table 2; low BWT indicates learning new tasks severely impacts the performance of previous ones; low FWT indicates learning new tasks hardly benefits from previous tasks.

Our proposed methods adopted a simple sparse autoencoder-based PTM $\mathcal{F}(\cdot)$ without a convolution layer, which was introduced in CRNet (see Fig. 2 in [2]). We used the same PTMs for other algorithms for a fair comparison unless one already had a specific PTM. The metric values with *std.* are also reported in Table 1. In the non-CL group, the offline and separate edRVFL can be seen as the accuracy upper bound for the entire dataset and every single task, respectively. In contrast, the non-incremental version which forgets previous task information is the lower bound of ACC.

For the group based on regularization, SOTA CRNet and IF2NET outperform others, which can be attributed to effective PTM and closed-form solutions to adjust elastic weight penalties, reducing retention loss during CIL. These methods have a good resistance to random task order. Current OTCIL methods, such as MAS, DYSON, GEM, and GSS, experience performance degradation when encountering long task streams. RanDumb, the best model-based method, has similarities to ours in using the randomized projector and linear learner, but ours introduces unsupervised knowledge and ensemble learning. CVAE+ResNet-50 and CGAN+ResNet-50 in the replay-based group also achieve high accuracy because they use strong PTM ResNet-50 and allow revisits. As shown in Table 1, our methods based on -F improve ACC to more than 0.97, compared to 0.71 for DER++ and 0.83 for DYSON. This demonstrates that our approaches mitigate catastrophic forgetting better and without relying on replay buffers (unlike DER++) or complex meta-learning components (unlike DYSON).

The edRVFL-kF and edRVFL-kF-Bayes exhibit excellence in metrics. The following conclusions can be drawn from Table 1: (1) -kF-Bayes obtain a relatively superior and stable accuracy without intractable k optimization in 3 varied task streams, in contrast to -R and -kF. The -R style is greatly affected when $T = 5$ because of no boundary and no prior knowledge premises, which proves the efficacy of -kF and -kF-Bayes; (2) the performance of the proposed methods is remarkably close to the offline expert's. This, together with positive BWT and small FWT, also suggests performance recovery and little learning dissipation in OTCIL even when starting with a naive model; (3) -kF is not worse than using -R, under the premise of optimized k; (4) based on above analysis, we recommend employing edRVFL-kF-Bayes with PTM in OTCIL. The ACC curves on FashionMNIST-10/5 can be found in Fig. 4, and the dynamic adaptive process of k in -kF-Bayes is shown in Fig. 5. In practical industrial applications, tabular, visual, or textual data can be extracted as numerical features by specific PTMs and then fed into our

algorithms for representation learning.

Table 2: Partial HP values in FashionMNIST tests

Parameters Algorithms↓	N	L	$\log_2(\lambda^{-1})$	k
edRVFL-R	2	1	0	-
edRVFL-kF	2	1	2	1.4142127133
edRVFL-kF-Bayes	2	1	6	*dynamic* k_l

Note: we did not optimize the networks' HP by SMAC3 except for λ and the k of -kF style; instead, we used minimum architectures, namely $N = 2$ and $L = 1$ to realize competitive results in Table 1; $\mathcal{F}(\cdot)$ was with $N1 = 10$, $N2 = 10$,$N3 = 670$ as the same with CRNet.

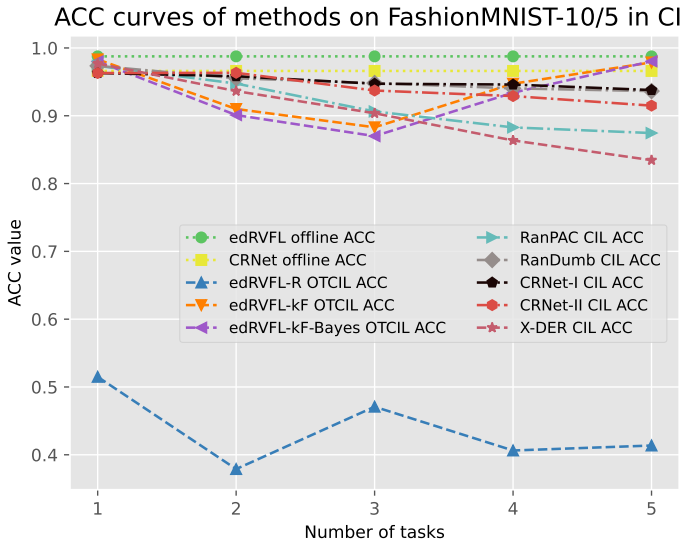


Figure 4: ACC curves of partial methods of Table 1 on FashionMNIST-10/5. Methods learned 2 classes in each task. The edRVFL-R, edRVFL-kF, and edRVFL-kF-Bayes learned under OTCIL scenario without task boundary.

Next, we want to explore the dynamic performance and ablation tests of edRVFL-kF and edRVFL-kF-Bayes. The methods were first configured by SMAC3 with the best HP sets to achieve optimal performance, where the N and L in Table 2 were increased to 512 and 5 respectively, and each class

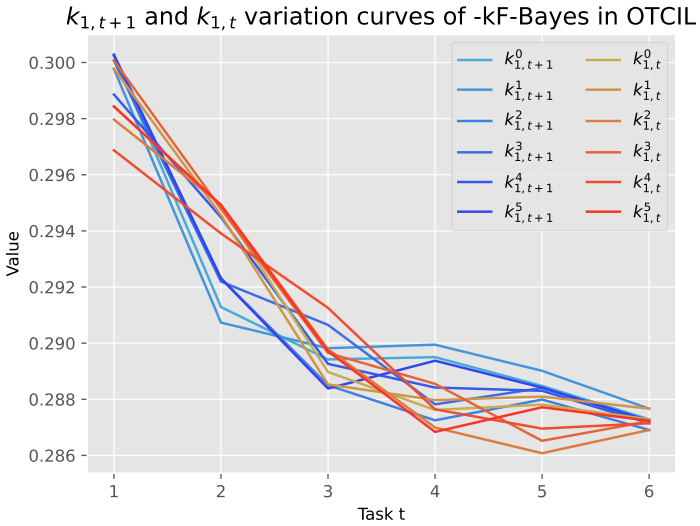


Figure 5: $k_{l,t+1}$ and $k_{l,t}$ variation curves of -kF-Bayes learning 5 tasks in OTCIL. The results of the first 6 experiments are shown here, marked by superscripts. It shows the dynamic adjustments and a phase shift between the two variables. Trends illustrate the effect of maintaining the balance of penalties in changing optimization targets.

was learned in two batches. The following describes the conducted ablation experiments.

(1) *PTM-less operation* demands that our methods work without a feature extractor or PTM. As shown in Fig. 6, removing PTM affects the final immediate testset $acc.(t)$ where all accuracies drop to over 10%. The -kF and -kF-Bayes respond quickly, and in the end, -R and -kF-Bayes obtain higher $acc.(T)$ as they grasp nearly full knowledge halfway through each task. Combined with Fig. 7, although -kF starts with a smaller error, it does not perform as well overall as -kF-Bayes. This reveals the difficulties in balancing all aspects of achieving good performance by simply adjusting k in -kF.

(2) *Varying the number of tasks split* investigates performance variation on learning task streams of different lengths. As shown in Fig. 8, edRVFL-R suffers from large accuracy loss, and sub-learners fluctuate widely on the 20-batch task stream, while this situation is alleviated when learning on the 30-batch task stream; even sub-learners still vary greatly in their behavior. By contrast, little loss is incurred in using -kF and -kF-Bayes, and all sub-learners work well. The related ACC curves can be found in Fig. 9. More

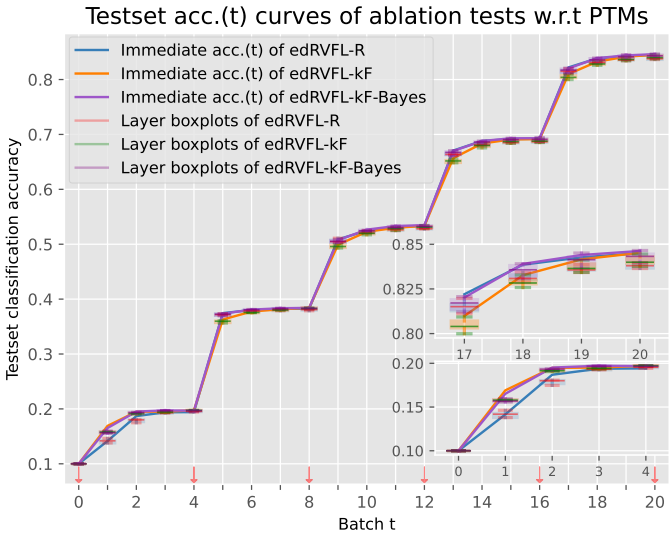


Figure 6: The FashionMNIST testset immediate $acc.(t)$ curves of edRVFL using -R, -kF, and -kF-Bayes styles without PTM in OTCIL process. Accuracy increases in steps with time. X-axis denotes batches where algorithms learn one class in two batches, and Y-axis shows accuracy values. Boxplot shows statistics of immediate accuracy for interior sub-learners. The ↓ reveals task boundary. X-axis is locally enlarged in the inlaid subfigures.

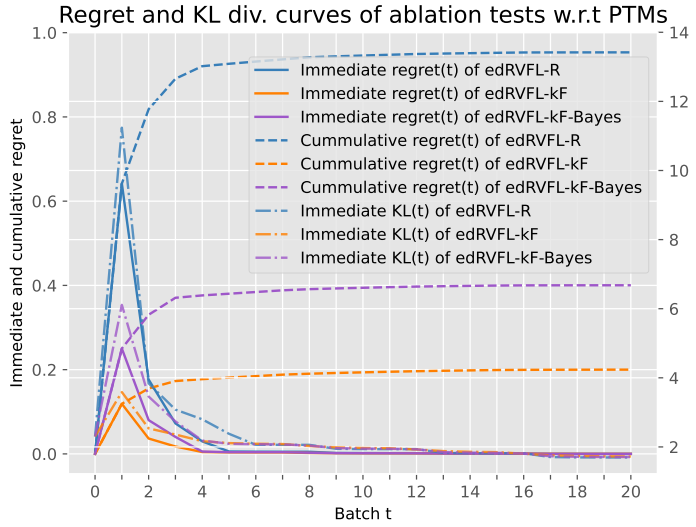


Figure 7: The FashionMNIST testset immediate $regret(t)$ and $KL(t)$ curves of edRVFL using -R, -kF, and -kF-Bayes styles in OTCIL process. -kF has the minimum loss at the beginning. X-axis denotes batches, Y-axis (left) is regret and cumulative regret, and Y-axis (right) is KL divergence.

batches of updates improve their performance. Given that the methods are learning from a disturbed start, normal R needs more batches to recover the learning process and increase accuracy, while the sub-learners using -kF and -kF-Bayes can be more stable in harsh contexts and maintain expected results.

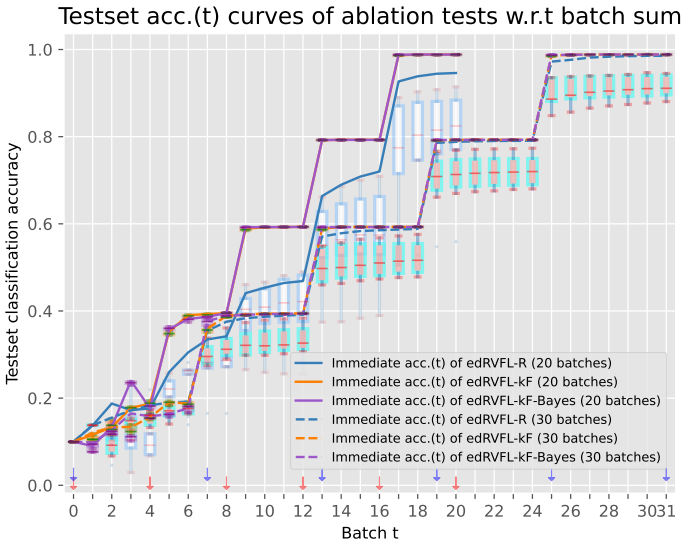


Figure 8: The FashionMNIST testset immediate $acc.(t)$ curves of edRVFL with -R, -kF, and -kF-Bayes styles learning on varied task streams. X-axis denotes batches, and Y-axis is $acc.(t)$ value. Boxplot shows statistics of immediate accuracy for interior sub-learners. The ↓ and ↓ respectively reveal varied task boundaries of 20- and 30-batch task streams.

5.3. CIFAR-100 dataset

For the CIFAR-100 dataset, we split it into 3 CIL environments: CIFAR-100/5, CIFAR-100/10, and CIFAR-100/100 to implement experiments focusing on continual learning and knowledge retention, and also explored ablation tests on network architecture. We selected well-behaved methods from Table 1, such as EWC [1], MAS [18], CRNet [2] for the regularization-based group, RanPAC [35], NICE [49], DYSON [19] for the model-based methods, and GEM [8], GSS [12] for the replay-based branch. A standard resnet-56 was employed as PTM and used to extract features from CIFAR-100, which inherited the setup in [34, 39, 2] for fair comparison, and to enable benchmarks to learn image representations. We offered the same PTM for all algorithms

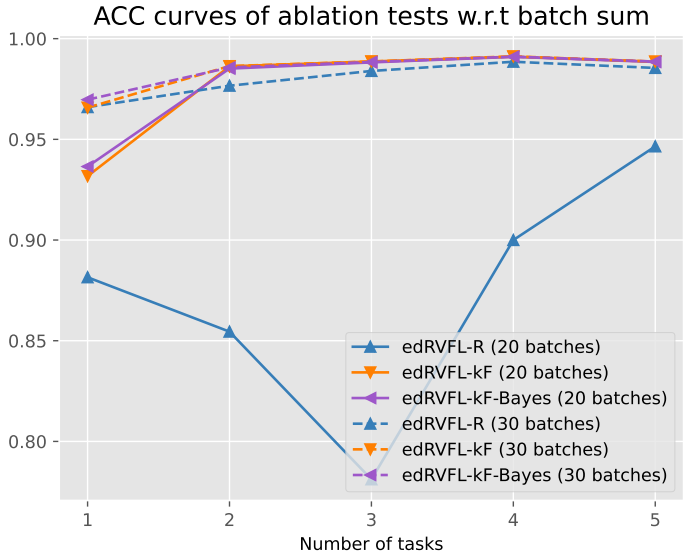


Figure 9: The FashionMNIST testset ACC curves of edRVFL with -R, -kF, and -kF-Bayes styles learning on varied task streams. X-axis denotes the task number, and Y-axis is the ACC value. Updating in more batches can improve our method performance.

unless they are already equipped with one, such as for NICE and DYSON. After resnet-56, the PTM $\mathcal{F}(\cdot)$ was followed in our proposed methods for enhancement. The backbone architecture and HP setups adhered to original papers as CIFAR-100 was a common dataset in the CL community. The task order was randomly sorted and baselines were tested 10 times. No task boundary was given to OTCIL methods. Our methods still learned every task in one-pass, had no revisits, and we abode by severe no prior update before the data arrival.

For CIFAR-100/5, the methods were subjected to learning 20 classes in each task. As the ACC results are shown in Fig. 10, fine-tuning experienced the lowest accuracy because it destroyed past learned weights in the current tuning. Other methods (e.g. MAS, EWC) exhibit clear signs of forgetfulness when continually learning tough tasks. The IL2M and Randumb achieve remarkable accuracy and GSS also shows better performance supported by the replay buffer. CRNet-I/II obtains around 0.76 on ACC. Our methods were roughly adjusted in SMAC3, the number of trials was 200, $T = 400$ meant each class was finished in 4 batches. Other HP are listed in Table 3. Our method edRVFL-kF-Bayes realizes the best accuracy at 0.80, and

it also demonstrates a fast increase. The edRVFL-R and edRVFL-kF get 0.7863 and 0.7875 accuracy, respectively. The corresponding variation of k_t of edRVFL-kF-Bayes is shown in Fig. 11, which shows the process of tracking during OTCIL and being adjusted automatically to optimal values, and the OTCIL dynamic performance described by $acc.(t)$ is drawn in Fig. 12. In contrast, the constant k of -kF is only set to a rigid intermediate value by SMAC3.

Table 3: Structural HP values on CIFAR-100 tests

Algorithms Main HP↓	edRVFL-R	edRVFL-kF	edRVFL-kF-Bayes
$N1^*$	20	30	20
$N2^*$	10	10	10
$N3^*$	1000	1000	900
N	512	8	512
L	1	5	1
$\log_2(\lambda^{-1})$	$4^\dagger$	$4^\dagger$	$4^\dagger$
g	Leaky ReLU	ReLU	tanh
k	-	2.2867217520	*dynamic* $k_l{}^\dagger$
κ	-	-	$1.0^\dagger$

Note: HP marked by * is belonging to PTM $\mathcal{F}(\cdot)$; HP value marked by † denotes it is not included in SMAC3 optimization; the dimension of HP configuration space of -kF is one higher than others; subsequently we continued to use these HP for CIFAR-100 dataset.

Then we extended the task amount to 10 by creating CIFAR-100/10. The task order and choice of classes in each task were randomized. The well-behaved algorithms in Fig. 10 were selected to be studied further in Fig. 13. Combined with Fig. 14 below, the proposed edRVFL-kF-Bayes still maintains an obvious advantage in performance. The ultimate experiment was CIFAR-100/100, where this pattern required single CIL and presented a rigorous inspection of knowledge retention skills. We only compared with CRNets and gave some observations based on Fig. 15: (1) the initial harsh environments slowed down our algorithms' learning, and the impacts were gradually moderated in OTCIL process. Finally, the $ACC(T)$ of ready-to-work edRVFL-kF-Bayes reached 0.829; (2) edRVFL-kF performed badly in the new CIFAR-100/100 scenario while we still used the previous HP in Table 3, which demonstrated -kF was poorly adapted and demanded precise

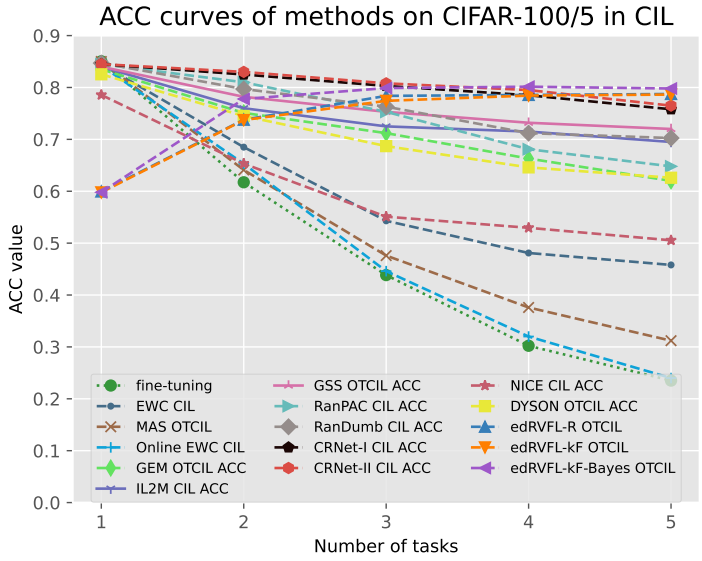


Figure 10: ACC curves of methods on CIFAR-100/5 in CIL. Each task contained 20 classes. The edRVFL-R, edRVFL-kF, and edRVFL-kF-Bayes learned tasks in OTCIL.

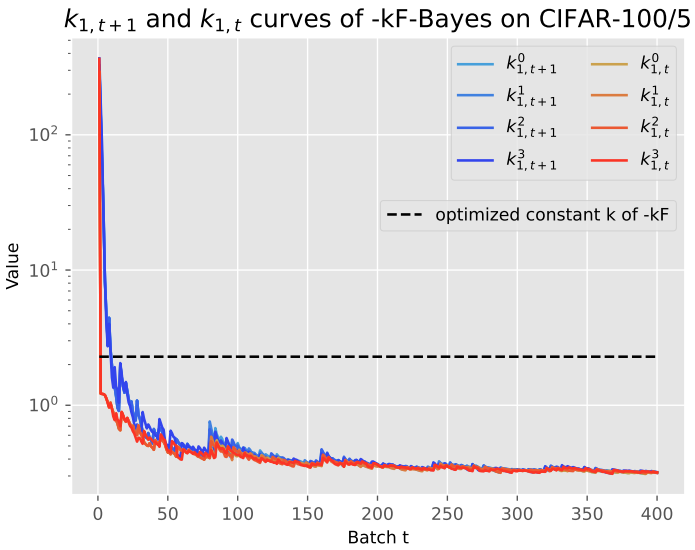


Figure 11: $k_{l,t+1}$ and $k_{l,t}$ variation curves of -kF-Bayes in OTCIL on CIFAR-100/5 task stream. X-axis denotes batch number, Y-axis is the logarithmic value. The results of the first 4 experiments are shown here, marked by superscripts. The phase shift between the two k_l is observed again. Similar trends illustrate the effect of maintaining the balance of penalties in changing optimization targets.

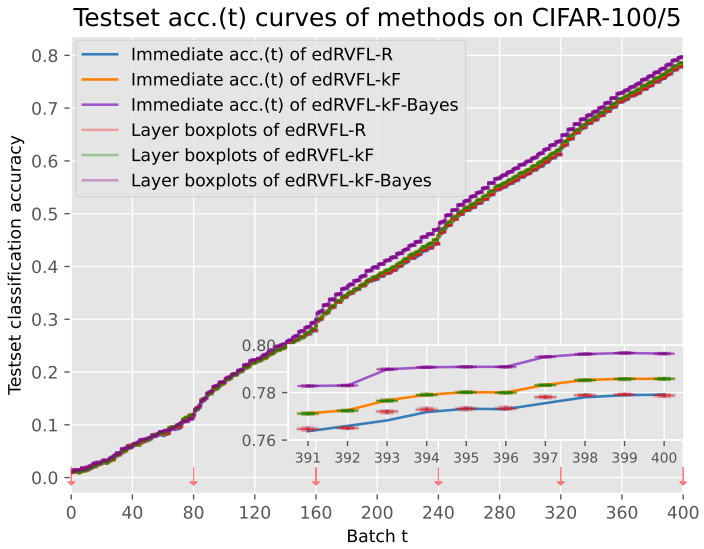


Figure 12: The CIFAR-100/5 testset immediate $acc.(t)$ curves of edRVFL using -R, -kF, and -kF-Bayes styles in OTCIL process. Accuracy steps up as more batches arrive. X-axis denotes batch number, and Y-axis represents accuracy value. Boxplot shows statistics of immediate accuracy for interior sub-learners. The ↓ reveals task boundary. X-axis is locally enlarged in the inlaid subfigures.

adjustments in different tasks. Unfortunately, it is impractical to set k of -kF simultaneously as that requires previous data. So we need to reduce the dependence on the exact configuration of sensitive k; (3) the CRNets underwent large oscillations during CIL and got around 0.76 accuracy. We likewise point out that CRNet-I took over 20 times longer than ours.

The ablation experiments for CIFAR-100/10 were concentrated on network architectures. The structural HP mainly refers to the number of sub-learners' layers L with N neurons inside each. For the ablation tests w.r.t L or N, we separately increased all the layers L by 2 and set N to 1024, and then retrained the 3 models on CIFAR-100/10. The resulting ACC curves of ablation tests on L and N are shown in Fig. 16 and Fig. 17 respectively. The results suggest that the advantages of using -kF-Bayes still remain, and the methods mainly differ in the dynamic response speed. The -kF and -kF-Bayes strategies were relatively stable in the ablation tests.

In addition, we would like to explain why BWT is positive and the reason behind the increasing trend at the start of our algorithms, which can be attributed to two factors: (1) the severe environment. To simulate the prac-

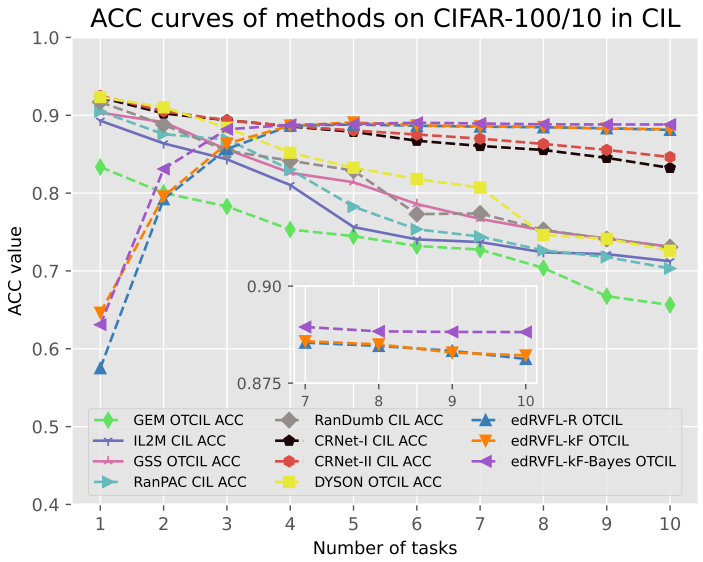


Figure 13: ACC curves of methods on CIFAR-100/10 in CIL. Each task contained 10 classes. The edRVFL-R, edRVFL-kF, and edRVFL-kF-Bayes learned tasks in OTCIL. X-axis is locally enlarged in the inlaid subfigures.

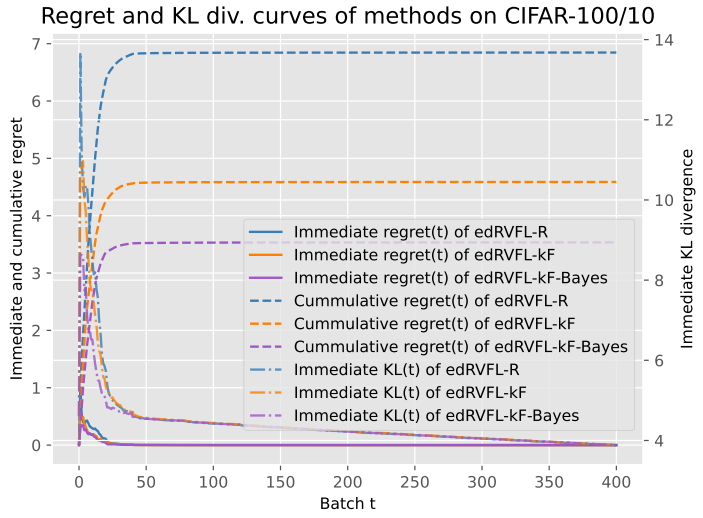


Figure 14: The CIFAR-100/10 testset immediate $regret(t)$ and $KL(t)$ curves of edRVFL using -R, -kF, and -kF-Bayes styles in OTCIL process. -kF-Bayes has the minimum loss in the whole process. X-axis denotes batch number, Y-axis (left) is regret and cumulative regret, and Y-axis (right) is KL.

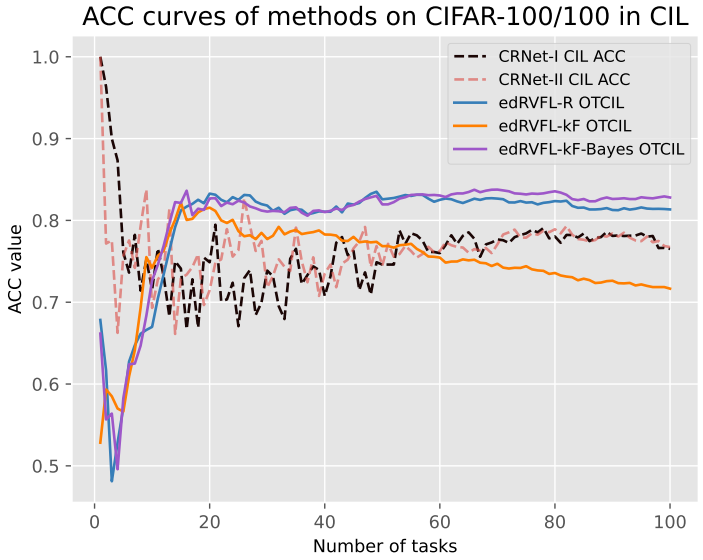


Figure 15: ACC curves of methods on CIFAR-100/100 in CIL. Each task contained 1 class. The edRVFL-R, edRVFL-kF, and edRVFL-kF-Bayes learned tasks in OTCIL, and CRNet learned tasks in CIL.

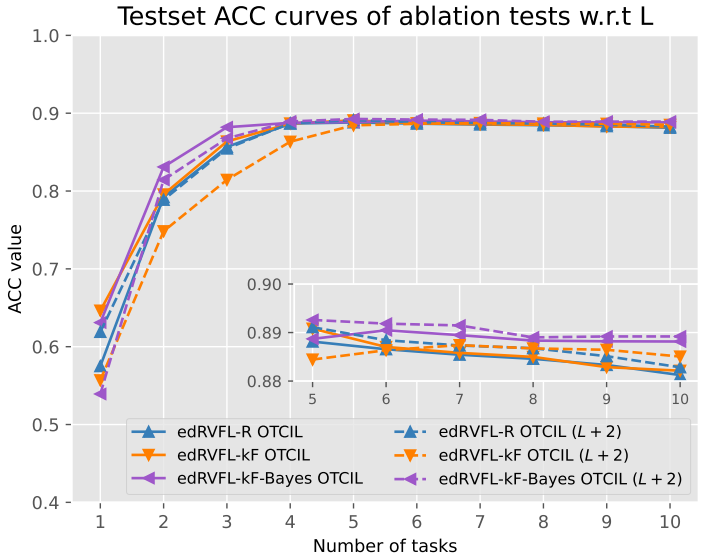


Figure 16: CIFAR-100/10 *ACC* curves of ablation tests w.r.t L. L was increased by 2 in all methods, and other HP were held. The edRVFL-kF-Bayes gets the best performance.

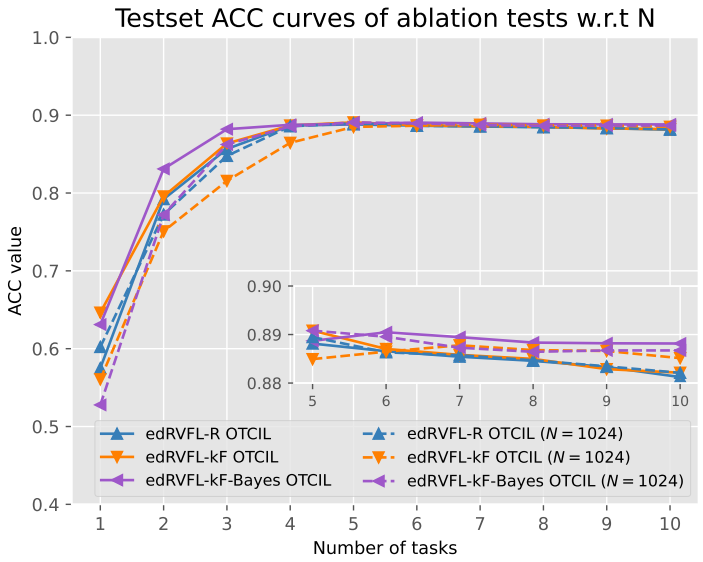


Figure 17: CIFAR-100/10 *ACC* curves of ablation tests w.r.t N. N was set to 1024 in all methods, and other HP were held. The edRVFL-kF-Bayes gets the best performance.

tical environment, we did not assume any prior knowledge or initialization before data arrival, so our initial HP setups were unaware of any incoming task information, but only relied on the regular OTCIL process to reasonably update. This mitigation behavior can be observed in the dashed lines of Fig. 18, which also shows why algorithms perform better when one task is split into more batches; (2) no task boundary information. We strictly adhered to the rules in OTCIL, and consequently, the learnable weights would be scattered among other unseen classes, resulting in misjudgment. This phenomenon was more pronounced in CIL when one task contained more classes and fewer batches. A small batch or task capacity was beneficial to the stability of the solution, and a given boundary limited the scattering of weights, further improving the accuracy. When task boundaries were known, this led to enhanced results are shown in Fig. 18's solid lines.

5.4. Generalizability on large ViT

Our algorithms with preposed PTMs achieved good results in image classification experiments, confirming the validity of using PTMs. Recently released official models were trained on massive datasets with handcrafted tricks, which have brought strong generalizability and made PTMs more ac-

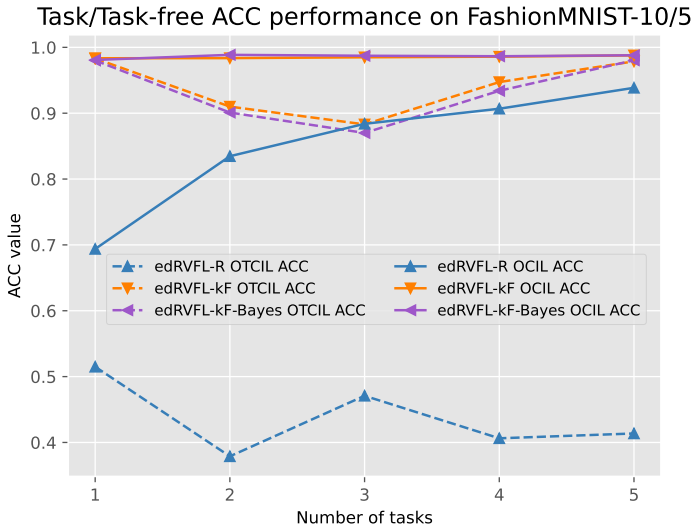


Figure 18: Task/Task-free CIL *ACC* performance on FashionMNIST-10/5. Compare the *ACC* changes with Fig. 4. This experiment is also an ablation test that reveals the influence of removing task boundary information.

cessible for downstream tasks [65]. Popular work on PTM-based CIL can be found in [66, 67].

As the SimpleCIL combining large ViT with similarity measurements outperformed SOTA baselines [65], we explore next whether our algorithms could still work effectively with large PTMs. Two issues were important to note: (1) PTMs should be frozen during (OT)CIL, which guaranteed consistent convex optimization and closed-form solutions in our algorithm. (2) A significant domain gap still existed between pre-trained datasets and incremental streams (e.g., ImageNet and CIFAR), requiring our algorithms to bridge distribution drift for downstream tasks. Based on the setups in [65], we integrated a ViT from timm[6] into our proposed algorithms to generate generalizable embeddings for feasibility and transferability studies.

The backbone ViT-B/16-IN21K with over 85M parameters was pre-trained on ImageNet-21k (14M images, more than 21K classes) at a resolution of 224x224. This PTM $\mathcal{F}(\cdot)$ was designated as [CLS] tokens of the Transformer

[6]Use timm.create_model('vit_base_patch16_224_in21k',pretrained = True, num_classes = 0) with timm==0.6.12 or https://huggingface.co/google/vit-base-patch16-224-in21k to load PTM.

instead of patches, which were fed into our algorithms. In this case, edRVFL-kF and edRVFL-kF-Bayes could still avoid the exemplar replay and catastrophic forgetting. The downstream task was CIFAR-100/20, which aligns with CIFAR B0 Inc5 in [65].

Table 4: Testset accuracy comparisons of CIL algorithms with large ViT PTM on CIFAR-100/20

Algorithms	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	ACC
Finetune	38.90	20.17	-
Finetune Adapter[68]	60.51	49.32	-
LwF[69]	46.29	41.07	-
SDC[70]	68.21	63.05	-
L2P[71]	85.94	79.93	-
DualPrompt[72]	87.87	81.15	-
CODA-Prompt[73]	89.11	81.96	-
CPP[74]	85.21	78.64	-
SimpleCIL	87.57	81.26	81.28
APER w/ Finetune	87.67	81.27	81.27
APER w/ VPT-Shallow	90.43	84.57	81.80
APER w/ VPT-Deep	88.46	82.17	82.65
APER w/ SSF	87.78	81.98	82.01
APER w/ Adapter	90.65	85.15	85.14
edRVFL-R (ViT)	93.17	86.17	90.13
edRVFL-kF (ViT)	93.72	86.14	90.32
edRVFL-kF-Bayes (ViT)	**94.48**	**86.28**	**90.92**

Note: APER-based methods and defined accuracy ($\bar{\mathcal{A}}$, $\mathcal{A}_B$) for comparison could be found in [65]; the best value of each metric is shown in **bold**.

Table 4 records the results of the above experiments. Our methods used common parameter configurations rather than special tuning: $N = 256$, $L = 2$, $g = Sigmoid$ for all backbone edRVFLs, $k = 2.0$ for -kF style, and $\sigma = 10^{-3}$ and $\kappa = 128$ for -kF-Bayes style. We only fine-tuned the λ of precarious edRVFL-R to ensure robust results, as some singularities dropped its accuracy by more than 30%. The following conclusions can be drawn: (1) under the condition that large PTMs were accessible, our proposed methods still yielded remarkable results in all algorithms. (2) edRVFL-kF-Bayes outperforms others, and edRVFL-kF is the second best in our group, though they shared the same backbone structure. These prove that edRVFL-kF-

Bayes remains efficient for large PTMs.

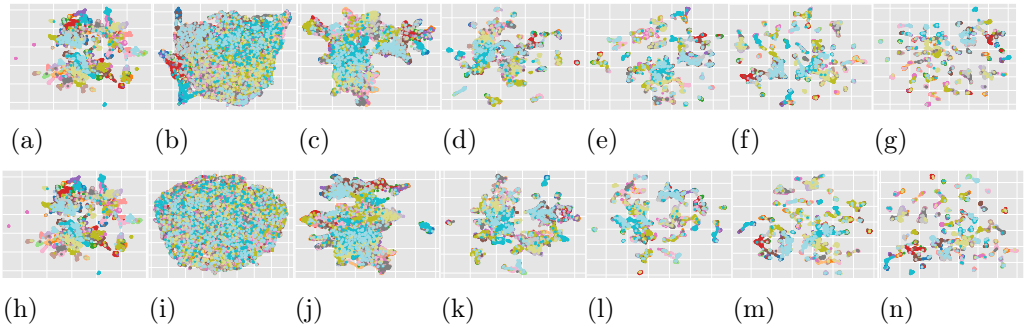
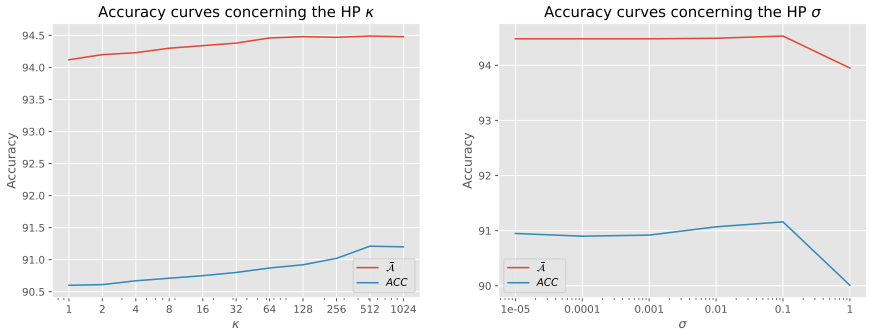


Figure 19: t-SNE visualization of the decision boundary on CIFAR-100/20, where colorful dots denote data representations after ViT and randomization. (a)-(g): boundary evolution of using edRVFL-R (ViT). (h)-(n): boundary evolution of using edRVFL-kF-Bayes (ViT). (a), (h) are the initial features; (b)-(g) represent the results after learning 0, 20, 40, 60, 80, 100 classes respectively, as do (i)-(n).

Next, we investigated how the random projections interacted with the forward style by visualizing the decision boundaries using t-SNE during the OTCIL process. In Fig. 19, we compared edRVFL-R (ViT) and edRVFL-kF-Bayes (ViT). It can be observed that: (1) the random mapping provides complete representations, and the regularized loss functions can gradually partition the boundary of the random projections as OTCIL progresses, proving that algorithms work. (2) edRVFL-kF-Bayes (ViT) employs enhanced gradients that result in lower future regrets in OTCIL, leading to sparser between-class margins and improved classification results. Visualizations reveal the evolution mechanism and effectiveness of using edRVFL-kF-Bayes (ViT) in OTCIL.

Finally, we would like to explain the specific HP (i.e., σ, κ) selections of edRVFL-kF-Bayes. Although this appears to introduce extra HP over edRVFL-kF, it is actually more stable and easier to tune because the performance of edRVFL-kF-Bayes is less sensitive and dependent on these two HP. In Fig. 20, the magnitudes of the performance variations due to different κ are less than 0.5% for $\bar{\mathcal{A}}$ and 0.7% for ACC; the magnitudes of variations in performance due to different σ are less than 0.5% for $\bar{\mathcal{A}}$ and 1.2% for ACC. The results demonstrate that edRVFL-kF-Bayes will maintain similar superiority and robustness over wide ranges of its HP. Therefore, we recommend manual adjustment or linear search to adjust them.



(a) (b)

Figure 20: Effect of main HP on edRVFL-kF-Bayes on CIFAR-100/20. (a): accuracy variation under HP κ changes. (b): accuracy variation under HP σ changes.

6. Conclusion

To achieve immediate decision-making and reduce loss of knowledge retention in OTCIL scenarios, especially containing long task streams, a framework of multi-layer Randomized NN with controllable unsupervised knowledge is proposed to enhance precognition learning and guide gradients of continuous optimization, hoping to outperform the canonical -R and have lower regret. Regarding the framework as the OCO, we theoretically derive the algorithm of edRVFL-kF with adjustable k forward regularization, which features one-pass closed-form incremental updates and variable learning rate. It also effectively resists catastrophic forgetting and replay. Moreover, to avoid unstable penalties in OTCIL and mitigate intractable tuning of hard k, we propose the ready-to-work edRVFL-kF-Bayes with soft k self-adapted, based on Bayesian learning and distribution retaining in the ever-changing optimization target of non-i.i.d streams. We carried out comprehensive experiments in multiple scenarios, from which we observed the superiority of using edRVFL-kF-Bayes in OTCIL processes. Our future work will focus on the theoretical foundation of regret bounds using the -kF style in OTCIL.

Beyond technical merits, it is important to note the potential misuse stemming from the method's incremental efficiency. Since the closed-form incremental update enables rapid adaptation to new facial identities or behavioral patterns, deployment without proper governance can lead to unauthorized surveillance systems that continuously integrate unseen individuals' data.

Acknowledgements

The authors thank the National Natural Science Foundation of China (No. 62273230, 62203302) and the State Scholarship Fund of China Scholarship Council (No. 202206230182) for their financial support. We also thank the editors and reviewers for their valuable comments and suggestions.

Declaration of Interest

The authors declare that they have no competing interests or personal relationships that could influence the work reported in this paper.

CRediT Author Statement

Junda Wang: Methodology, Software, Writing - Original Draft. Minghui Hu: Conceptualization, Validation. Ning Li: Supervision, Funding acquisition. Abdulaziz Al-Ali: Supervision, Resources. Ponnuthurai Nagaratnam Suganthan: Supervision, Project administration.

References

[1] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the national academy of sciences 114 (13) (2017) 3521–3526.

[2] D. Li, Z. Zeng, Crnet: A fast continual learning framework with random theory, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (9) (2023) 10731–10744.

[3] M. Kim, Theoretical bounds of generalization error for generalized extreme learning machine and random vector functional link network, Neural Networks 164 (2023) 49–66.

[4] Q. Wei, W. Zhang, Class-incremental learning with balanced embedding discrimination maximization, Neural Networks 179 (2024) 106487.

[5] R. M. French, Catastrophic forgetting in connectionist networks, Trends in cognitive sciences 3 (4) (1999) 128–135.

[6] J. Kim, W. Lee, M. Eo, W. Rhee, Improving forward compatibility in class incremental learning by increasing representation rank and feature richness, Neural Networks 183 (2025) 106969.

[7] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International conference on machine learning, PMLR, 2017, pp. 3987–3995.

[8] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Advances in neural information processing systems 30 (2017).

[9] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of learning and motivation, Vol. 24, Elsevier, 1989, pp. 109–165.

[10] G. Sun, B. Ji, L. Liang, M. Chen, Cecr: Cross-entropy contrastive replay for online class-incremental continual learning, Neural Networks 173 (2024) 106163.

[11] S. Tian, L. Li, W. Li, H. Ran, X. Ning, P. Tiwari, A survey on few-shot class-incremental learning, Neural Networks 169 (2024) 307–324.

[12] R. Aljundi, M. Lin, B. Goujaud, Y. Bengio, Gradient based sample selection for online continual learning, Advances in neural information processing systems 32 (2019).

[13] R. Zhou, W. Zhu, S. Han, M. Kang, S. Lü, Vcsap: Online reinforcement learning exploration method based on visitation count of state-action pairs, Neural Networks (2025) 107052.

[14] R. Wang, J. Wu, X. Cheng, X. Liu, H. Qiu, Adaptive expert fusion model for online wind power prediction, Neural Networks (2024) 107022.

[15] J. Fan, Y. Ge, X. Zhang, Z. Wang, H. Wu, J. Wu, Learning the feature distribution similarities for online time series anomaly detection, Neural Networks 180 (2024) 106638.

[16] Y. Li, X. Yang, Q. Gao, H. Wang, J. Zhang, T. Li, Cross-regional fraud detection via continual learning with knowledge transfer, IEEE Transactions on Knowledge and Data Engineering 36 (12) (2024) 7865–7877. doi:10.1109/TKDE.2024.3451161.

[17] L. Yang, Z. Luo, S. Zhang, F. Teng, T. Li, Continual learning for smart city: A survey, IEEE Transactions on Knowledge and Data Engineering 36 (12) (2024) 7805–7824. doi:10.1109/TKDE.2024.3447123.

[18] R. Aljundi, K. Kelchtermans, T. Tuytelaars, Task-free continual learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 11254–11263.

[19] Y. He, Y. Chen, Y. Jin, S. Dong, X. Wei, Y. Gong, Dyson: Dynamic feature space self-organization for online task-free class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 23741–23751.

[20] Y. Zhang, B. Gao, D. Yang, W. L. Woo, H. Wen, Online learning of wearable sensing for human activity recognition, IEEE Internet of Things Journal 9 (23) (2022) 24315–24327.

[21] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International conference on machine learning, PMLR, 2017, pp. 3987–3995.

[22] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, R. Hadsell, Progress & compress: A scalable framework for continual learning, in: International conference on machine learning, PMLR, 2018, pp. 4528–4537.

[23] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, J. Van De Weijer, Class-incremental learning: survey and performance evaluation on image classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (5) (2022) 5513–5533.

[24] H. An, J. Yang, X. Zhang, X. Ruan, Y. Wu, S. Li, J. Hu, A class-incremental learning approach for learning feature-compatible embeddings, Neural Networks 180 (2024) 106685.

[25] Y. Ghunaim, A. Bibi, K. Alhamoud, M. Alfarra, H. A. Al Kader Hammoud, A. Prabhu, P. H. Torr, B. Ghanem, Real-time evaluation in online continual learning: A new hope, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11888–11897.

[26] P.-B. Zhang, Z.-X. Yang, A new learning paradigm for random vector functional-link network: Rvfl+, Neural Networks 122 (2020) 94–105.

[27] Y. Xiao, M. Adegoke, C.-S. Leung, K. W. Leung, Robust noise-aware algorithm for randomized neural network and its convergence properties, Neural Networks 173 (2024) 106202.

[28] A. K. Malik, R. Gao, M. Ganaie, M. Tanveer, P. N. Suganthan, Random vector functional link network: recent developments, applications, and future directions, Applied Soft Computing 143 (2023) 110377.

[29] Q. Shi, R. Katuwal, P. N. Suganthan, M. Tanveer, Random vector functional link neural network based ensemble deep learning, Pattern Recognition 117 (2021) 107978.

[30] R. Gao, R. Li, M. Hu, P. N. Suganthan, K. F. Yuen, Online dynamic ensemble deep random vector functional link neural network for forecasting, Neural Networks (2023).

[31] K. S. Azoury, M. K. Warmuth, Relative loss bounds for on-line density estimation with the exponential family of distributions, Machine learning 43 (2001) 211–246.

[32] R. Ouhamma, O.-A. Maillard, V. Perchet, Stochastic online linear regression: the forward algorithm to replace ridge, Advances in Neural Information Processing Systems 34 (2021) 24430–24441.

[33] J. Serra, D. Suris, M. Miron, A. Karatzoglou, Overcoming catastrophic forgetting with hard attention to the task, in: International conference on machine learning, PMLR, 2018, pp. 4548–4557.

[34] W. Hu, Q. Qin, M. Wang, J. Ma, B. Liu, Continual learning by using information of each class holistically, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 7797–7805.

[35] M. D. McDonnell, D. Gong, A. Parvaneh, E. Abbasnejad, A. van den Hengel, Ranpac: Random projections and pre-trained models for continual learning, Advances in Neural Information Processing Systems 36 (2024).

[36] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010.

[37] Y. Kong, L. Liu, M. Qiao, Z. Wang, D. Tao, Trust-region adaptive frequency for online continual learning, International Journal of Computer Vision 131 (7) (2023) 1825–1839.

[38] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, T. Tuytelaars, Memory aware synapses: Learning what (not) to forget, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 139–154.

[39] G. Zeng, Y. Chen, B. Cui, S. Yu, Continual learning of context-dependent processing in neural networks, Nature Machine Intelligence 1 (8) (2019) 364–372.

[40] R. Cheng, R. Gao, M. Hu, P. N. Suganthan, K. F. Yuen, Wind speed forecasting using an ensemble deep random vector functional link neural network based on parsimonious channel mixing, in: 2024 International Joint Conference on Neural Networks (IJCNN), 2024, pp. 1–8. doi:10.1109/IJCNN60899.2024.10650817.

[41] A. Bhambu, R. Gao, P. N. Suganthan, Recurrent ensemble random vector functional link neural network for financial time series forecasting, Applied Soft Computing 161 (2024) 111759.

[42] M. Hihat, G. Garrigos, A. Fermanian, S. Bussy, Multivariate online linear regression for hierarchical forecasting, arXiv preprint arXiv:2402.14578 (2024).

[43] Z. Zhang, D. Bombara, H. Yang, Discounted adaptive online learning: Towards better regularization, in: Forty-first International Conference on Machine Learning, 2024.

[44] A. Jacobsen, A. Cutkosky, Online linear regression in dynamic environments via discounting, in: Forty-first International Conference on Machine Learning, 2024.

[45] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, Advances in neural information processing systems 31 (2018).

[46] J. Liu, Z. Ji, Y. Yu, J. Cao, Y. Pang, J. Han, X. Li, Parameter-efficient fine-tuning for continual learning: A neural tangent kernel perspective, arXiv preprint arXiv:2407.17120 (2024).

[47] A. S. Benjamin, C. Pehle, K. Daruwalla, Continual learning with the neural tangent ensemble, Advances in Neural Information Processing Systems 37 (2024) 58816–58840.

[48] D. Needell, A. A. Nelson, R. Saab, P. Salanevich, O. Schavemaker, Random vector functional link networks for function approximation on manifolds, Frontiers in Applied Mathematics and Statistics 10 (2024) 1284706.

[49] M. B. Gurbuz, J. M. Moorman, C. Dovrolis, Nice: Neurogenesis inspired contextual encoding for replay-free class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 23659–23669.

[50] A. Krawczyk, A. Gepperth, An analysis of best-practice strategies for replay and rehearsal in continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4196–4204.

[51] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, C.-C. J. Kuo, Class-incremental learning via deep model consolidation, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2020, pp. 1131–1140.

[52] D. Li, T. Wang, B. Xu, K. Kawaguchi, Z. Zeng, P. N. Suganthan, If2net: Innately forgetting-free networks for continual learning, arXiv preprint arXiv:2306.10480 (2023).

[53] V. K. Verma, K. J. Liang, N. Mehta, P. Rai, L. Carin, Efficient feature transformations for discriminative and generative continual learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13865–13875.

[54] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, D. Wierstra, Pathnet: Evolution channels gradient descent in super neural networks, arXiv preprint arXiv:1701.08734 (2017).

[55] Z. Ke, B. Liu, X. Huang, Continual learning of a mixed sequence of similar and dissimilar tasks, Advances in neural information processing systems 33 (2020) 18493–18504.

[56] D. Deng, G. Chen, J. Hao, Q. Wang, P.-A. Heng, Flattening sharpness for dynamic gradient projection memory benefits continual learning, Advances in Neural Information Processing Systems 34 (2021) 18710–18721.

[57] A. Prabhu, S. Sinha, P. Kumaraguru, P. H. Torr, O. Sener, P. K. Dokania, Randumb: A simple approach that questions the efficacy of continual representation learning, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2024).

[58] D. Li, T. Wang, J. Chen, W. Dai, Z. Zeng, Harnessing neural unit dynamics for effective and scalable class-incremental learning, in: Forty-first International Conference on Machine Learning, 2024.

[59] J. Rajasegaran, M. Hayat, S. H. Khan, F. S. Khan, L. Shao, Random path selection for continual learning, Advances in neural information processing systems 32 (2019).

[60] S. Tang, D. Chen, J. Zhu, S. Yu, W. Ouyang, Layerwise optimization by gradient decomposition for continual learning, in: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, 2021, pp. 9634–9643.

[61] E. Belouadah, A. Popescu, Il2m: Class incremental learning with dual memory, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 583–592.

[62] R. Wang, Y. Bao, B. Zhang, J. Liu, W. Zhu, G. Guo, Anti-retroactive interference for lifelong learning, in: European Conference on Computer Vision, Springer, 2022, pp. 163–178.

[63] P. Buzzega, M. Boschini, A. Porrello, D. Abati, S. Calderara, Dark experience for general continual learning: a strong, simple baseline, Advances in neural information processing systems 33 (2020) 15920–15930.

[64] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, S. Calderara, Class-incremental continual learning into the extended der-verse, IEEE transactions on pattern analysis and machine intelligence 45 (5) (2022) 5497–5512.

[65] D.-W. Zhou, Z.-W. Cai, H.-J. Ye, D.-C. Zhan, Z. Liu, Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need, International Journal of Computer Vision 133 (3) (2025) 1012–1032.

[66] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, Z. Liu, Class-incremental learning: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[67] D.-W. Zhou, H.-L. Sun, J. Ning, H.-J. Ye, D.-C. Zhan, Continual learning with pre-trained models: A survey, arXiv preprint arXiv:2401.16386 (2024).

[68] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, P. Luo, Adaptformer: Adapting vision transformers for scalable visual recognition, Advances in Neural Information Processing Systems 35 (2022) 16664–16678.

[69] Z. Li, D. Hoiem, Learning without forgetting, IEEE transactions on pattern analysis and machine intelligence 40 (12) (2017) 2935–2947.

[70] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, J. van de Weijer, Semantic drift compensation for class-incremental learning. 2020 ieee, in: CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6980–6989.

[71] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, T. Pfister, Learning to prompt for continual learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 139–149.

[72] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al., Dualprompt: Complementary prompting for rehearsal-free continual learning, in: European conference on computer vision, Springer, 2022, pp. 631–648.

[73] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, Z. Kira, Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 11909–11919.

[74] Z. Li, L. Zhao, Z. Zhang, H. Zhang, D. Liu, T. Liu, D. N. Metaxas, Steering prototypes with prompt-tuning for rehearsal-free continual learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 2523–2533.

Supplementary material

Appendix A. Bregman divergence property

Lemma 3. *[31] Bregman divergence property.*

1. *The divergence is a linear operator.* $\forall \mu \geqslant 0$, $\Delta_{G_1+\mu G_2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta}) = \Delta_{G_1}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta}) + \mu \Delta_{G_2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta})$.
2. *If* $G_1(\boldsymbol{\beta}) - G_2(\boldsymbol{\beta}) = \omega \boldsymbol{\beta} + \upsilon, \omega \in \mathbb{R}^{|\boldsymbol{\beta}|}, \upsilon \in \mathbb{R}$, *then* $\Delta_{G_1}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta}) = \Delta_{G_2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\beta})$.
3. *If* $G = \frac{1}{2}||\cdot||_2^2$, $G' = \frac{1}{2}||\cdot||_F^2$, A *and* B *are matrices of the same size,* $\boldsymbol{a_i}$ *and* $\boldsymbol{b_i}$ *are respective column vectors, then* $\sum \Delta_G(\boldsymbol{a_i}, \boldsymbol{b_i}) = \Delta_{G'}(A, B)$.

Proof: The proofs for the first two can be found in [31]. For the third item, we prove here:

$$\begin{aligned} \Delta_{G'}(A, B) &= \frac{1}{2}||A||_F^2 - \frac{1}{2}||B||_F^2 - \langle (A - B), B \rangle_F \\ &= \sum \frac{1}{2}||a_i||_2^2 - \sum \frac{1}{2}||b_i||_2^2 - \sum \langle (a_i - b_i), b_i \rangle \\ &= \sum \Delta_G(\boldsymbol{a_i}, \boldsymbol{b_i}) \end{aligned} \tag{A.1}$$

(A.1) shows that the Bregman divergence of a matrix measured by the Frobenius norm can be computed separately by column vectors, which ensures that we generalize our -kF algorithm to the multi-dimensional Vovk-Azoury-Warmuth (VAW) form [42].
Proof finished. □

Appendix B. Theorem 2 proof

Proof: (9) can be expanded to:

$$\begin{aligned} \theta_{l,t+1} = \arg\min_{\theta_l} & U_{l,t}(\theta_l) - U_{l,t}(\theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) \\ & - \sum\nolimits_{ij} (\theta_l - \theta_{l,t}) \otimes \nabla U_{l,t}(\theta_{l,t})_{ij} + k \cdot \hat{\mathcal{L}}_{l,t+1}(\theta_l) - k \cdot \hat{\mathcal{L}}_{l,t}(\theta_l), \end{aligned} \tag{B.1}$$

where the $\otimes$ denotes Hadamard product. In fact, the $\nabla U_{l,t}(\theta_{l,t})_{ij} = 0$ because of the latest convex optimization. So that (B.1) can be simplified to:

$$\theta_{l,t+1} = \arg\min_{\theta_l} U_{l,t+1}(\theta_l) - U_{l,t}(\theta_{l,t}) \tag{B.2}$$

$$= \arg\min_{\theta_l} \Delta_{U_{l,0}}(\theta_l, \theta_{l,0}) + \mathcal{L}_{l,1..t}(\theta_l) + k \cdot \hat{\mathcal{L}}_{l,t+1}(\theta_l) - const.$$

Solutions to both models (i.e. online learner (9) and offline expert (8)) remain the same at each time point t if designating the last target as Bregman divergence function. This also suggests that -kF suffers no learning dissipation compared to the offline expert because they have similar optima at each step.
Proof finished. □

Appendix C. Theorem 3 proof

Proof: Following the loss function in (4) expressed by MLE of Gaussian distribution, we define the forms of $\Delta_{U_{l,0}}$, $\mathcal{L}_{l,t}$, and $\hat{\mathcal{L}}_{l,t+1}$. The immediate incurred loss on $D_{l,t}$ is denoted by $\mathcal{L}_{l,t}(\theta_l) = \frac{1}{2}||D_{l,t}\theta_l - Y_t||_F^2$, and $\mathcal{L}_{l,1..t}(\theta_l) = \sum_{i=1}^{t} \frac{1}{2}||D_{l,i}\theta_l - Y_i||_F^2$ is the cumulative loss on t data batches. Similarly define the forward predictive loss to $\hat{\mathcal{L}}_{l,t+1}(\theta_l) = \frac{1}{2}||D_{l,t+1}(\theta_l - \theta_{l,0})||_F^2$.

Let the initial Bregman function be $U_{l,0}(\theta_l) = Tr(\frac{1}{2}\theta_l^T(\eta_{l,0})^{-1}\theta_l)$, where Tr signifies matrix trace, $(\eta_{l,0})^{-1}$ is a symmetric positive definite matrix. Based on Lemma 3, initial Bregman divergence is defined as $\Delta_{U_{l,0}}(\theta_l, \theta_{l,0}) = \sum_{i=1}^{m} \Delta_{U_{l_i,0}}(\theta_{l_i}, \theta_{l_i,0}) = \sum_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,0})^T(\eta_{l,0})^{-1}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,0})$ because the $U_{l,0}$ is a Frobenius norm projector w.r.t θ_l. Define the learning processes of sub-learners of edRVFL-kF in OTCIL to the same form as Theorem 1. For $0 \leqslant t \leqslant T$ we have:

$$\theta_{l,t+1} = \arg\min_{\theta_l} U_{l,t+1}(\theta_l), 1 \leq l \leq L \tag{C.1}$$

$$\begin{aligned} U_{l,t+1}(\theta_l) = &\sum\nolimits_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,0})^T(\eta_{l,0})^{-1}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,0}) \\ &+ \sum\nolimits_{i=1}^{t} \frac{1}{2}||D_{l,i}\theta_l - Y_i||_F^2 + \frac{k}{2}||D_{l,t+1}(\theta_l - \theta_{l,0})||_F^2 \end{aligned}$$

Convert (C.1) into the form of (9) described by Theorem 2 to avoid the retrospective retraining, and derive variable learning rate via Lemma 3:

$$\begin{aligned} &U_{l,t}(\theta_l) + \frac{1}{2}||D_{l,t}\theta_l - Y_t||_F^2 + \frac{k}{2}||D_{l,t+1}(\theta_l - \theta_{l,0})||_F^2 - \frac{k}{2}||D_{l,t}(\theta_l - \theta_{l,0})||_F^2 \\ &= U_{l,0}(\theta_l) - U_{l,0}(\theta_{l,0}) - \sum\nolimits_{ij} (\theta_l - \theta_{l,0}) \otimes \nabla U_{l,0}(\theta_{l,0})_{ij} \end{aligned}$$

$$
\begin{aligned}
&+\sum\nolimits_{i=1}^{t}\frac{1}{2}||D_{l,i}\theta_l - Y_i||_F^2 + \frac{k}{2}||D_{l,t+1}(\theta_l-\theta_{l,0})||_F^2 \\
&\Rightarrow U_{l,t}(\theta_l) - U_{l,0}(\theta_l) \\
&=\sum\nolimits_{i=1}^{t-1}\frac{1}{2}||D_{l,i}\theta_l - Y_i||_F^2 + \frac{k}{2}||D_{l,t}(\theta_l-\theta_{l,0})||_F^2 \\
&-Tr(\frac{1}{2}\theta_{l,0}^T(\eta_{l,0})^{-1}\theta_{l,0}) - \sum\nolimits_{i=1}^{m}(\theta_{l_i}-\theta_{l_i,0})^T\cdot(\eta_{l,0})^{-1}\theta_{l_i,0} \qquad \text{(C.2)} \\
&=\sum\nolimits_{i=1}^{t-1}\frac{1}{2}||D_{l,i}\theta_l - Y_i||_F^2 + \frac{k}{2}||D_{l,t}(\theta_l-\theta_{l,0})||_F^2 \\
&+Tr(\frac{1}{2}\theta_{l,0}^T(\eta_{l,0})^{-1}\theta_{l,0}) - Tr(\theta_l^T(\eta_{l,0})^{-1}\theta_{l,0}) \\
&=Tr(\frac{1}{2}\theta_l^T\sum\nolimits_{i=1}^{t-1}D_{l,i}^TD_{l,i}\theta_l) + Tr(\frac{1}{2}\sum\nolimits_{i=1}^{t-1}Y_i^TY_i) + Tr(\frac{1}{2}\theta_{l,0}^T(\eta_{l,0})^{-1}\theta_{l,0}) \\
&-Tr(\sum\nolimits_{i=1}^{t-1}Y_i^TD_{l,i}\theta_l) - Tr(\theta_l^T(\eta_{l,0})^{-1}\theta_{l,0}) + Tr(\frac{k}{2}(\theta_l-\theta_{l,0})^TD_{l,t}^TD_{l,t}(\theta_l-\theta_{l,0}))
\end{aligned}
$$

Based on Lemma 3, (C.2) can be simplified to:

$$
\begin{aligned}
&U_{l,t}(\theta_l) - (U_{l,0}(\theta_l) + Tr(\frac{1}{2}\theta_l^T\sum\nolimits_{i=1}^{t-1}D_{l,i}^TD_{l,i}\theta_l) + Tr(\frac{k}{2}\theta_l^TD_{l,t}^TD_{l,t}\theta_l)) \qquad \text{(C.3)} \\
&=Tr(\omega\cdot\theta_l) + const.
\end{aligned}
$$

where $\omega \in \mathbb{R}^{m\cdot(s+N)}$. If set $V_{l,t} = Tr(\frac{1}{2}\theta_l^T(\sum_{i=1}^{t-1}D_{l,i}^TD_{l,i} + k\cdot D_{l,t}^TD_{l,t})\theta_l)$, relation can be deduced from (C.3) and Lemma 3:

$$
\begin{aligned}
&\Delta_{U_{l,t}}(\theta_l,\theta_{l,t}) = \Delta_{U_{l,0}+V_{l,t}}(\theta_l,\theta_{l,t}) = \Delta_{U_{l,0}}(\theta_l,\theta_{l,t}) + \Delta_{V_{l,t}}(\theta_l,\theta_{l,t}) \qquad \text{(C.4)} \\
&=\sum\nolimits_{i=1}^{m}\frac{1}{2}(\boldsymbol{\theta}_{l_i}-\boldsymbol{\theta}_{l_i,t})^T[(\eta_{l,0})^{-1} + \sum\nolimits_{j=1}^{t-1}D_{l,j}^TD_{l,j} + k\cdot D_{l,t}^TD_{l,t}](\boldsymbol{\theta}_{l_i}-\boldsymbol{\theta}_{l_i,t})
\end{aligned}
$$

The above derivation applies well to the ERM paradigm combined with Gaussian MLE and differentiable convex regularization terms, which could bring beneficial effects and enhancement to the learning process in OTCIL. (C.4) shows that it alleviates the previous data revisit with all the knowledge absorbed in Bregman divergence. The solution to (C.4) that needs a stepwise solver during the OTCIL process will change with $(\eta_{l,t})^{-1} = (\eta_{l,0})^{-1} + \sum_{i=1}^{t-1}D_{l,i}^TD_{l,i} + k\cdot D_{l,t}^TD_{l,t}$, also termed the variable learning rate. The essence of the edRVFL-kF algorithm is also to provide time-varying optimization targets to guide the optimal updates of the network in task streams.

This optimization objective in Theorem 2 can be expressed further as:

$$\theta_{l,t+1} = \arg\min_{\theta_l} \Delta_{U_{l,t}}(\theta_l, \theta_{l,t}) + \mathcal{L}_{l,t}(\theta_l) + k \cdot \hat{\mathcal{L}}_{l,t+1}(\theta_l) - k \cdot \hat{\mathcal{L}}_{l,t}(\theta_l) \tag{C.5}$$
$$= \arg\min_{\theta_l} \sum\nolimits_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t})^T[(\eta_{l,0})^{-1} + \sum\nolimits_{j=1}^{t-1} D_{l,j}^T D_{l,j} + k \cdot D_{l,t}^T D_{l,t}](\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t})$$
$$+ \frac{1}{2}||D_{l,t}\theta_l - Y_t||_F^2 + k \cdot \frac{1}{2}||D_{l,t+1}(\theta_l - \theta_{l,0})||_F^2 - k \cdot \frac{1}{2}||D_{l,t}(\theta_l - \theta_{l,0})||_F^2$$

In order to maintain the generality of the above derivation steps to any OCO problems, (C.5) is transformed to the following constrained optimization problem:

$$\min_{\theta_l \in \theta_{l,t+1}} \sum\nolimits_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t})^T(\eta_{l,t})^{-1}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t}) \tag{C.6}$$
$$+ \frac{1}{2}||\xi_1||_F^2 + k \cdot \frac{1}{2}||\xi_2||_F^2 - k \cdot \frac{1}{2}||\xi_3||_F^2$$
$$s.t.\ D_{l,t}\theta_l - Y_t = \xi_1; D_{l,t+1}(\theta_l - \theta_{l,0}) = \xi_2; D_{l,t}(\theta_l - \theta_{l,0}) = \xi_3, \forall t$$

where $\xi_{\{1,2,3\}}$ denotes the gap between ground truth and prediction. The Lagrangian function of problem (C.6) is:

$$\ell(\theta_l, \xi_{\{1,2,3\}}, \mu_{\{1,2,3\}}) = \sum\nolimits_{i=1}^{m} \frac{1}{2}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t})^T(\eta_{l,t})^{-1}(\boldsymbol{\theta}_{l_i} - \boldsymbol{\theta}_{l_i,t}) \tag{C.7}$$
$$+ \frac{1}{2}||\xi_1||_F^2 + k \cdot \frac{1}{2}||\xi_2||_F^2 - k \cdot \frac{1}{2}||\xi_3||_F^2 + Tr(\mu_1^T(D_{l,t}\theta_l - Y_t - \xi_1))$$
$$+ Tr(\mu_2^T(D_{l,t+1}(\theta_l - \theta_{l,0}) - \xi_2)) + Tr(\mu_3^T(D_{l,t}(\theta_l - \theta_{l,0}) - \xi_3)),$$

where μ denotes the Lagrangian multiplier and is the same size as ξ. The Lagrangian function (C.7) can be tackled through Karush-Kuhn-Tucker (KKT) conditions, which can be built into the following formulation:

$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \theta_l} = 0 \Rightarrow (\eta_{l,t})^{-1}(\theta_l - \theta_{l,t}) + D_{l,t}^T\mu_1 + D_{l,t+1}^T\mu_2 + D_{l,t}^T\mu_3 = 0$$
$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \xi_1} = 0 \Rightarrow \xi_1 = \mu_1$$
$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \xi_2} = 0 \Rightarrow k\xi_2 = \mu_2$$

$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \xi_3} = 0 \Rightarrow -k\xi_3 = \mu_3 \tag{C.8}$$
$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \mu_1} = 0 \Rightarrow D_{l,t}\theta_l - Y_t = \xi_1$$
$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \mu_2} = 0 \Rightarrow D_{l,t+1}(\theta_l - \theta_{l,0}) = \xi_2$$
$$\frac{\partial \ell(\theta_l, \xi, \mu)}{\partial \mu_3} = 0 \Rightarrow D_{l,t}(\theta_l - \theta_{l,0}) = \xi_3$$

Based on (C.8), we can obtain the recursive updating policy between $\theta_{l,t}$ and $\theta_{l,t+1}$ as follows:

$$\begin{aligned}\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(D_{l,t}^T D_{l,t} + k \cdot D_{l,t+1}^T D_{l,t+1} - k \cdot D_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t] \\ + k \cdot \eta_{l,t+1}(D_{l,t+1}^T D_{l,t+1} - D_{l,t}^T D_{l,t})\theta_{l,0}\end{aligned} \tag{C.9}$$

Without loss of generality, given $\theta_{l,0} = 0$ or $\{D_{l,t+1}, D_{l,t}\}$ both drawn from i.i.d assumption, to allow the learner to start from no prior knowledge under strict conditions, we have:

$$\theta_{l,t+1} = \theta_{l,t} - \eta_{l,t+1}[(D_{l,t}^T D_{l,t} + kD_{l,t+1}^T D_{l,t+1} - kD_{l,t}^T D_{l,t})\theta_{l,t} - D_{l,t}^T Y_t] \tag{C.10}$$

This suggests that the relevant weight is not learned and remains 0 until after a new class arrives. Using this methodology, we can elegantly obtain the variable learning rate and weight updating policy. (C.10) endows the edRVFL-kF with the innate ability of *non-replay* and *non-increase parameter*, because the values are only relevant to the chunks of current and next data batches. The model can deliver immediate and accurate decisions, profiting from recursive closed-form solutions. The algorithmic procedures of edRVFL-kF in OTCIL could be found in **Algorithm 1**.
Proof finished. □