# Online Learning from Evolving Feature Spaces with Deep Variational Models

Heng Lian, Di Wu, Bo-Jian Hou, Jian Wu, and Yi He*

**Abstract**—In this paper, we explore a novel online learning setting, where the online learners are presented with "doubly-streaming" data. Namely, the data instances constantly streaming in are described by feature spaces that over-time evolve, with new features emerging and old features fading away. The main challenge of this problem lies in the fact that the newly emerging features are described by very few samples, resulting in *weak* learners that tend to make error predictions. A seemingly plausible idea to overcome the challenge is to establish a relationship between the old and new feature spaces, so that an online learner can leverage the knowledge learned from the old features to better the learning performance on the new features. Unfortunately, this idea does not scale up to high-dimensional feature spaces that entail very complex feature interplay. Specifically. a tradeoff between *onlineness*, which biases shallow learners, and *expressiveness*, which requires deep models, is inevitable. Motivated by this, we propose a novel paradigm, named *Online Learning Deep models from Data of Double Streams (OLD$^3$S), where a shared latent subspace is discovered to summarize information from the old and new feature spaces, building an intermediate feature mapping relationship. A key trait of OLD$^3$S is to treat the* model capacity *as a learnable semantics, aiming to yield optimal model depth and parameters* jointly *in accordance with the complexity and non-linearity of the input data streams in an online fashion. To ablate its efficacy and applicability, two variants of OLD$^3$S are proposed namely, OLD-Linear that learns the relationship by a linear function; and OLD-FD learns that two consecutive feature spaces pre-and-post evolution with fixed deep depth. Besides, instead of re-starting the entire learning process from scratch, OLD$^3$S learns multiple newly emerging feature spaces in a lifelong manner, retaining the knowledge from the learned and vanished feature space to enjoy a jump-start of the new features' learning process. Both theoretical analysis and empirical studies substantiate the viability and effectiveness of our proposed approach. The code is available online at github.com/X1aoLian/OLD3S-L.*

**Index Terms**—Data Streams, Online Learning, Streaming Algorithms, Open Feature Spaces

## 1 Introduction

Machine learning has become a fundamental building block in many cyber infrastructures, provides an automated hence scalable apparatus to analyze the high-dimensional data streams (*e.g.*, images, texts, videos) pervading all corners of the Internet [1]–[3]. Examples include multimedia retrieval [4], [5], online speech analytics [6], [7], recommender systems [8]–[12], to just name a few. Generally speaking, wherever it is infeasible to inspect and process the data growing in an increasingly unmanageable volume with manpower, machine learning prevails.

Despite their fashionability, a prominent drawback shared by most existing machine learning methods is their limited *generalization capability* [13]. As a matter of fact, machine learning models usually do well in practice only if the data arriving in future tend to follow a nearly identical distribution as the data they were trained on [14], [15]. This so-called *i.i.d.* assumption inevitably limits the model expressiveness to our society that constantly evolves.

To aid the situation, a new learning paradigm termed *online learning from doubly-streaming data* has emerged with both algorithmic designs [16]–[29] and domain applications [30]–[34]. Its key idea is to generalize learning models in two spaces. First, the *sample space*, where the data instances are generated ceaselessly, requiring to train learners on-the-fly, making real-time predictions as the data arrive. As such, if the patterns underlying data changed, an online learner can be updated instantly to adapt to the shift, thereby retaining its accuracy over time [35]–[37].

Second, the *feature space*, where sets of features describing the arriving data samples evolve, with new features emerge and old features stop to be generated. To wit, a smart manufacturing pipeline may employ a set of sensing techniques to detect unqualified products [38], where each sensor coheres to a feature. The feature space evolves, when the old sensors wear out and a batch of new sensors are deployed [16]. Tangibly, as the new and old sensors (*i.e.*, features) often differ in terms of amount, version, metric, and positions, a new classifier needs to be initialized. Yet, this new classifier may stay *weak* and error-prone before the training samples carrying these new features grows to a sufficiently large volume. Meanwhile, the old classifier becomes unusable with the unobserved features, leading to substantial waste of the data collection and training effort. A relationship between the pre-and-post evolving feature spaces must be established, so that the old features can be *reconstructed* from the new ones. Online learners can thus harvest the information embedded in the old classifier to aid the weak new classifier, enjoying a boosted learning performance [21], [23], [24], [26]. One more advantage of establishing such relationship is to avoid waste of multi train-

- *H.Lian, J.Wu, and Y.He were with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA. E-mail: {hlian, j1wu}@odu.edu, yihe@cs.odu.edu*
- *D. Wu was with the College of Computer and Information Science, Southwest University, Chongqing, China. E-mail: wudi.cigit@gmail.com*
- *B.J. Hou was with the Center for Biomedical Image Computing and Analytics, University of Pennsylvania, PA, USA. E-mail: houbo@pennmedicine.upenn.edu*

*Corresponding Author: Dr. Yi He (yihe@cs.odu.edu)*

ings. Once observing new features, both straightforward methods which capture a new subspace from all features, or establish a relationship for each pair of adjacent spaces will cause a huge waste of computations. Instead, our method can always re-capture a new latent representation can be constructed combining existed subspace and new features. Such an adaptive growing subspace can, simultaneously, leverage old knowledge and absorb new information.

Unfortunately, all existing studies suffer from a tradeoff between *onlineness* and *expressiveness*. Specifically, on the one hand, shallow learners (*e.g.*, generalized linear models [39], Hoeffding trees [40]) possess a faster online convergence rate, thanks to their simple model structures with a small number of trainable parameters [41]. However, due to their limited learning capacity, they usually end up with inferior performance when dealing with high-dimensional media streams, of which the feature interplay is often complex.

On the other hand, deep learners (*e.g.*, neural networks [42], [43], deep forests [44], [45]) enjoy a low-dimensional hidden representation to build accurate predictive models on complex raw inputs. Yet, their large number of parameters residing in the entangled model structures invites stochastic updates, leading to a very slow convergence rate. In an online learning context, more error predictions tend to be made before the learners converge to an equilibrium. These additional errors are recognized as *regrets*, where the slower the convergence rate, the larger the learner regrets in a hindsight.

Motivated by this tradeoff, this paper mainly explores one question: *How can we build an online learner that joins the two merits, namely, 1) converges as fast as shallow models to minimize the online regrets and 2) learns latent representations as expressive as deep models from high-dimensional inputs with complex feature relationships.*

Our affirmative answer provides a novel learning paradigm, termed *Online Learning Deep models from Data of Double Streams* (OLD$^3$S). Our key idea is to train an online learner that *automatically* adjusts its learning capacity in accordance with the complexities and temporal variation patterns of input data stream. Specifically, OLD$^3$S is with an over-complete neural architecture [43], [46], [47] and starts from using its shallow layers, approximating a simple classifier to attain fast convergence at initial rounds. Over time, the deeper layers are gradually mobilized, as more samples streaming in requires 1) a highly capable classifier that can learn expressive latent representations and 2) a precise delineation of complex feature interplay. Knowledge reuse is enabled in both aspects i) the shallow-to-deep model switch via representations sharing and ii) the pre-and-post evolving feature spaces via reconstructive mapping and ensemble learning [48]. This benefits our approach by expediting the convergence rate in a temporal continuum, so as to maximize its online efficiency and efficacy when learning from doubly-streaming data.

**Specific contributions of this paper are as follows:**

i) This is the first study to explore the doubly-streaming data mining problem in an online deep learning context, where the high-dimensional data streams with feature space evolution tend to incur a tradeoff between convergence rate and learning capacity. Section 3 manifest the technical challenges from empirical evidence.
ii) A novel OLD$^3$S approach is proposed to tackle the problem, where a modeling architecture with its depth *learned* from data is devised to adapt to minimize the online classification regrets and precisely approximate the feature-wise relationship on-the-fly. Detailed analysis can be found in Section 4.
iii) A theoretical analysis substantiates that OLD$^3$S can provably lead to performance improvement over in two aspects i) online learners with fixed depths and ii) a single classifier without feature space ensembling. Details are presented in Section 5.
iv) Real-world high-dimensional datasets covering domains of machine translation and image classification are employed to benchmark our approach. Results suggest the viability and effectiveness of our proposal, documented in Section 6.

## 2 Related Work

Our learning problem relates to two research threads i) Online Learning with Doubly-Streaming Data and ii) Deep Learning with Adaptive Capacity, with the relationship and differences between the prior study efforts and our proposed approach discussed in this section as follows.

### *Online Learning with Doubly-Streaming Data*

Online learning algorithms were devised for data stream processing [49], [50], where the reality of learning is in an on-the-fly setting hence lifts the memory constraint for data analysis at scale. In addition to allowing data to grow in terms of *volume*, in an orthogonal setting, hoping the *features* describing input data to stay strictly unchanging is unrealistic over long time spans. As a response, the pioneering studies [51]–[55] explored a setting of *incremental* feature learning, allow the arriving data instances to carry different sets of features yet later instances are assumed to include monotonically more features than the earlier ones. Subsequent works that strive to learn *evolving* feature spaces [16]–[24], [27]–[29] further relaxed the monotonicity constraint on the feature dynamics, enable effective learning when later instances stop carrying old features that appeared theretofore. A key technique shared by these methods is to establish a mapping relationship between two feature spaces. As such, once the old features fade away, their information can be reconstructed via the mapping, aiding the weak learner trained on insufficiently few instances carrying new features, join to make highly accurate predictions.

Despite their effectiveness in various settings, these methods all prescribe a *linear* model to fit the mapping, which is unfortunately not capable to deal with complex real data, *e.g.*, images in an evolving spectrum domain, documents written in different languages. We are aware of a very recent work [26] that does not use linear but copula model to fit a non-linear mapping with statistical guarantees. However, this work requires to deem each feature as a copula component, and hence cannot scale up to a high-dimensional space (*e.g.*, images or natural languages). Our proposed OLD$^3$S approach does not suffer this restriction by discovering a latent feature space in which the original data dimension is largely condensed, thereby being generalizable to a wider range of real applications.

***Deep Learning with Adaptive Capacity***

Neural networks have emerged for several decades to approximate underlying functions with arbitrary complexity [56]–[58]. However, their universal approximation capability is grounded on an assumption of an infinitely wide hidden layer, which cannot be satisfied in practical modeling. The advent of Deep Neural Networks sidestepped this issue by imposing a *hierarchical* representation learning procedure [59]–[61], trading in width for depth, so as to fit complex decision functions underlying data. However, this hierarchical design introduces *over-parameterization*, where the large number of learnable parameters request massive rounds of training iterations over huge datasets to converge. Implementing deep learning for online decision-making thus becomes seemingly impossible.

A key question to solve the challenge is how to choose the network depth (representing the entire model capacity) in accordance with the underlying function in an adaptive, automated, and data-agnostic fashion. Huang et al. [62] firstly theorized and implemented the concept of stochastic depth, a training procedure that trains shallow networks and tests with deep networks, randomly dropping a subset of layers to quickly identify key layers. A method of deducing which layers can be trimmed is therefore needed. Larsson et al. [63] later identified a strategy to construct deep networks structured as fractals. This confers the ability to regularize co-adaptation of subpaths, effectively allowing for the isolation of high performing layers within a larger architecture. We can now judge values of groups of layers, making a delineation of value more concrete. Sahoo et al. [43] and He et al. [25] demonstrated a *Hedge Backpropagation* mechanism for online/lifelong deep learning, where the model depth is deemed as a trainable semantic metric, jointly with the layer parameters to decide the function complexity learned from data streams in a dynamic way.

***Passive-Aggressive (PA) Algorithms.*** As the crux for HBP is to adjust the learning capacity of model, astute readers may correlate it with PA algorithms [64] [52] [65], which also possess the capability of adjusting model learning efficacy in accordance with data complexity. We supplement a discussion to clarify the differences between our work and PA algorithms. In particular, PA algorithms enabled more aggressive updates of model parameters, if error predictions are made, and remain unchanged otherwise. HBP training differs from PA algorithms in two aspects. First, PA algorithms focused on adjusting the updating steps to encourage fast convergence, whereas their models are with a *fixed* learning capacity. Second, PA algorithms are mainly tailored for *linear* classifiers, which mostly fail to deal with streaming media data, typically residing in high-dimensional spaces and with complex patterns. As such, HBP and PA algorithms are tackling different optimization objectives, thereby encountering disparate technical challenges.

***Progressive Learning (PL) Algorithms.*** We also note that another recent work termed as Progressive Learning [66] which presents some similarities with our OLD$^3$S. However, there are two fundamental differences between two works. First, our work introduces a problem that the task is described by an evolving feature space with dimensional alterations. In contrast, PL endeavors to grapple with a sequence of tasks with the varying distributions. Second, both methods initialize new models upon receiving new data, yet their approaches to utilizing prior model knowledge differ. Our OLD$^3$S speed up the convergence of the new model by extracting knowledge from old models, while PL freezes prior models, allowing them solely to extract data representations without participating in training updates. Hence, although both PL and OLD$^3$S attempt to leverage old knowledge, PL can not be implemented for our problem.

Unfortunately, all these deep methods fail to take the feature space evolution into account, a factor that can largely affect the non-linearity of the resultant learning function. As a result, they cannot be adapted to learn the doubly-streaming data. To fill the gap, we propose to bring together the two fragmented subfields of online deep learning and doubly-streaming data mining. In particular, we respect that the mapping relationship between the pre-and-post evolving feature spaces can be massively more complex than the previously explored linear models, and must be gauged by a neural approximator that grows its capacity autonomously and adaptively.

## 3 PRELIMINARIES

There are three parts in this section. We formulate the problem in Section 3.1, present the challenges in Section 3.2, and outline the key design ideas in Section 3.3.

### 3.1 Problem Statement

Let $\{(\mathbf{x}_t, y_t) \mid t = 1, 2, \ldots, T\}$ denote an input sequence, where $\mathbf{x}_t$ is the data instance observed at the $t$-th round, accompanied with a ground truth label $y_t \in \{1, 2, \ldots, C\}$. It is worth noting that our online classification problem is formulated in a multi-class regime with in total $C$ class options, which excels our competitors [16], [19], [24], [52] that focus on binary classification only.

In the context of doubly-streaming data, we follow the pioneer [16], consider the set of features describing $\mathbf{x}_t$ to evolve with the following regularity, illustrated in Figure 1.

Instances first observed during timespan $\mathcal{T}_1$ are from the feature space $\mathcal{S}_1$. In the overlapping period $\mathcal{T}_b$, the data distribution in $\mathcal{S}_1$ starts to evolve to a new distribution in $\mathcal{S}_2$, and instances from different spaces but with the same labels can be observed at the same time. When it comes to $\mathcal{T}_2$, only instances from $\mathcal{S}_2$ are observable. Specifically, three timespans are described as:

- In the timespan $t_1 \in \mathcal{T}_1 := \{1, \ldots, T_1\}$, the classifier can only observe the instances described by the feature space $\mathcal{S}_1$ while $\mathcal{S}_2$ is unobservable, *i.e.*, $\mathbf{x}_{t_1} \in \mathcal{S}_1 \subseteq \mathbb{R}^{d_1}$, each of which is a $d_1$-dimensional vector.
- In the timespan $t_b \in \mathcal{T}_b := \{T_1 + 1, \ldots, T_b\}$, the feature space evolves, and the classifier observes the two feature spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ simultaneously, with each data instance being $\mathbf{x}_{t_b} = [\mathbf{x}_{t_b}^{\mathcal{S}_1}, \mathbf{x}_{t_b}^{\mathcal{S}_2}]^\top \in \mathcal{S}_1 \times \mathcal{S}_2 \subseteq \mathbb{R}^{d_1+d_2}$.
- In the timespan $t_2 \in \mathcal{T}_2 := \{T_b + 1, \ldots, T_2\}$, the old feature space $\mathcal{S}_1$ opts out, and the classifier can observe the evolved $\mathcal{S}_2$ only. Each data instance is $\mathbf{x}_{t_2} \in \mathcal{S}_2 \subseteq \mathbb{R}^{d_2}$, a $d_2$-dimensional vector.

Note, such feature space evolving from $\mathcal{S}_1$ to $\mathcal{S}_2$ can be easily generalized to infinitely more spaces (*e.g.*, $\mathcal{S}_2$ to $\mathcal{S}_3$,

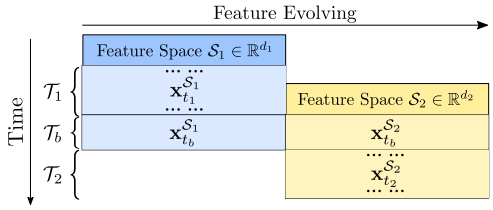



Fig. 1: Illustration of doubly-streaming data. Only in a very short timespan $|\mathcal{T}_b| \ll |\mathcal{T}_1|$ or $|\mathcal{T}_2|$, the samples are described by the two feature spaces concurrently.

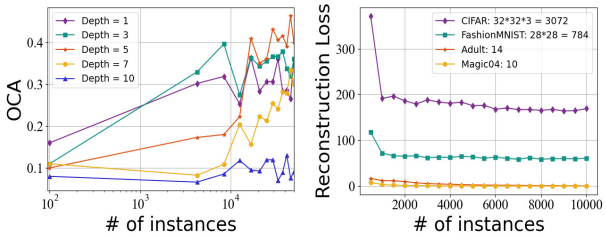



Fig. 2: Two challenges underlie the OLD$^3$S problem. *Left*: The deeper the learning model, the slower the convergence rate. *Right*: The higher the data dimensionality, the more inferior the feature relationship captured by linear mappings.

then $\mathcal{S}_3$ to $\mathcal{S}_4$), wherein all spaces can have disparate properties and semantic meanings and the mapping relationship between any two spaces can be arbitrarily complex. Such dynamism in the doubly-streaming data makes a prefix of learner capacity close to impossible.

At any time instant $t = \{t_1, t_b, t_2\}$, the learner $f_t$ observes $\mathbf{x}_t$ and makes a prediction $\hat{y}_t = f_t(\mathbf{x}_t)$. The true label $y_t$ is revealed thereafter, and an instantaneous loss indicating the discrepancy between $y_t$ and $\hat{y}_t$ is suffered. Based on the loss information, the learner updates to $f_{t+1}$ using first-order [67]–[70] or second-order [71]–[74] oracles, getting prepared for the next round. Our goal is to find a sequence of classifiers $\{f_1, \ldots, f_T\}$ that minimize the *empirical risk* [75] over $T$ rounds: $\min_{f_1,\ldots,f_T} \frac{1}{T}\sum_{t=1}^{T} \ell(y_t, f_t(\mathbf{x}_t))$, where $\ell(\cdot,\cdot)$ denotes the loss metric and often is prescribed as convex in its argument such as square loss or logistic loss.

### 3.2 Opportunities and Challenges

A common practice to enable online learning with doubly-streaming data is to leverage the overlapping timespan $\mathcal{T}_b$ to learn a *reconstructive mapping* $\phi : \mathcal{S}_2 \mapsto \mathcal{S}_1$, such that once the features of $\mathcal{S}_1$ are not observed during $\mathcal{T}_2$, their information can be reproduced, allowing the learner to harvest the old information learned during the $\mathcal{T}_1$ time period for better performance [16], [19], [24], [26].

Let $f_t = \{f_t^{\mathcal{S}_1}, f_t^{\mathcal{S}_2}\}$ denote the learner with $f_t^{\mathcal{S}_1}$ and $f_t^{\mathcal{S}_2}$ being the two classifiers corresponding to the $\mathcal{S}_1$ and $\mathcal{S}_2$ feature spaces, respectively. During $\mathcal{T}_2$, instead of predicting the observed instance as $f_t^{\mathcal{S}_2}(\mathbf{x}_t)$, the learner exploits the unobserved information from $\mathcal{S}_1$ to make prediction as: $f_t(\mathbf{x}_t) = \lambda_1 \cdot f_t^{\mathcal{S}_1}(\tilde{\mathbf{x}}_t) + \lambda_2 \cdot f_t^{\mathcal{S}_2}(\mathbf{x}_t)$, with $\tilde{\mathbf{x}}_t = \phi(\mathbf{x}_t) \in \mathcal{S}_1$ being the reconstructed data vector in the $\mathcal{S}_1$ space. With delicately tailored ensemble parameters $\lambda_1$ and $\lambda_2$, this reconstruction-based method enjoys a provably better prediction performance than using the classifier $f_t^{\mathcal{S}_2}$ only. Unfortunately, this method is not able to scale up to cope with real-world media data streams because of two challenges.

#### Challenge I – Train Deep Models On-The-Fly

The real-world media data carrying non-linear patterns often request deep learners (*e.g.*, neural network models) for effective processing. However, different from linear classifiers which are widely employed in previous studies [16]–[26], it is more difficult to train deep models in an online fashion. The large number of trainable parameters and complex model architectures tend to make deep learners data-hungry and converge slowly. In an online learning context, since each instance requiring an immediate prediction is presented only once, the deep learners tend to *regret* [75], making substantial errors before converging to equilibria. To verify this problem, a simple example reduced from the CIFAR experiment is illustrated in the left panel of Figure 2, where neural network models with different depths are trained on the same task in one-pass.

This example suggests that, as the model depth goes deeper, the learner suffers from a flatter convergence rate. Although such deep learners can end up with high online classification accuracy (OCA), they constantly underperform shallower models before given sufficient instances, thereby regretting largely. Notably, a learner with an improperly ultra-deep architecture (*cf.* depth = 10) may even fail to converge in an online setting. The reason can be possibly attributed to the diminishing feature reuse [62], [63] where the semantic meanings of raw inputs tend to be *washed out* by the layer-by-layer feedforward with massive randomly initialized parameters; No expressive representations can be learned online.

#### Challenge II – Learn Complex Reconstructive Mapping in Short Overlapping Timespans

In practice, an overlapping phase $\mathcal{T}_b$ in which the two feature spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ coexist is very short. Revisit the smart manufacturing example, where we can construct $\mathcal{T}_b$ by predeploying a batch of new sensors before the old sensors expiring their lifespans – a too long $\mathcal{T}_b$ is economically not affordable. This constraint blocks several seemingly plausible methods, *e.g.*, online transfer learning [76], [77], domain adaptation [78], [79], to work well, as they all require a sufficiently long overlapping phase to *align* the features before and after the evolution.

Prior studies [16], [19], [24] have advocated deducing *linear* functions to approximate the mapping relationship $\phi$ between the old and new features in a short $\mathcal{T}_b$, with the objective formulated as: $\min_{\phi} \sum_{t_b=T_1+1}^{T_b} \left\| \phi(\mathbf{x}_{t_b}^{\mathcal{S}_2}) - \mathbf{x}_{t_b}^{\mathcal{S}_1} \right\|_2^2$ where $\phi(\cdot) = \mathbf{W}^\top \cdot$. Unfortunately, this linear reconstructive mapping $\phi$ cannot work for media data streams with nonlinear feature interplay. An empirical evidence is presented in the right panel of Figure 2, in which we observe that, the higher the data dimension, the more complex the mapping relationship between two feature spaces, and hence the larger the reconstruction loss that a linear mapping suffers.

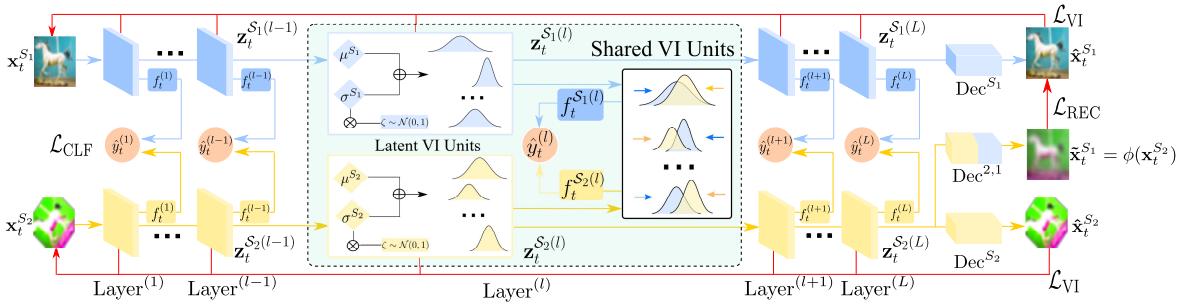



Fig. 3: An architectural illustration of our OLD$^3$S computational network during the overlapping $\mathcal{T}_b$ timespan. Instances $\mathbf{x}_t^{\mathcal{S}_1}$ and $\mathbf{x}_t^{\mathcal{S}_2}$ are encoded by two independent L-layer VAE models as $\mathbf{z}_t^{\mathcal{S}_1}$ and $\mathbf{z}_t^{\mathcal{S}_2}$ and decoded to $\hat{\mathbf{x}}_t^{\mathcal{S}_1}$ and $\hat{\mathbf{x}}_t^{\mathcal{S}_2}$, respectively. In each layer, variational codes $\mathbf{z}_t^{\mathcal{S}_1}$ and $\mathbf{z}_t^{\mathcal{S}_2}$ are forced to align in one shared latent subspace, and are predicted by classifiers $f_t^{\mathcal{S}_1}$ and $f_t^{\mathcal{S}_2}$, respectively. The final prediction $\hat{y}_t$ is the ensemble of results of two classifiers.

### 3.3 Our Thoughts

To overcome the two challenges, our key idea is to discover a set of *shared latent* features that summarize information from the pre-and-post evolving feature spaces $\mathcal{S}_1$ and $\mathcal{S}_2$. Compared with learning the mapping $\phi: \mathcal{S}_2 \mapsto \mathcal{S}_1$ directly in the short $\mathcal{T}_b$, our idea can exploit the long $\mathcal{T}_1$ timespan to learn a latent feature subspace from $\mathcal{S}_1$ independently at first, and then align it with that from $\mathcal{S}_2$ to expedite learning efficiency. Specifically, we employ variational inference [80] to model the underlying distribution of $\mathcal{S}_1$ stream as:

$$Q\left(\mathbf{z}_{t_1}^{\mathcal{S}_1} \mid \mathbf{x}_{t_1},\ t_1 = 1, \ldots, T_1\right) = \prod_{i=1}^{z} \mathcal{N}\left(\mathbf{z}_i^{\mathcal{S}_1} \mid \mu_i^{\mathcal{S}_1}, (\sigma_i^{\mathcal{S}_1})^2\right), \quad (1)$$

where Q indicates the conditional probability density, and $\mathcal{N}$ is a normal (Gaussian) distribution with corresponding mean $\mu_i^{\mathcal{S}_1}$ and variance $(\sigma_i^{\mathcal{S}_1})^2$. Thus, a variational code $\mathbf{z}^{\mathcal{S}_1} \in \mathbb{R}^z$ is drawn from a multivariate Gaussian that surrogates the data instances streaming from the original feature space $\mathcal{S}_1$ [81]. Later in the overlapping $\mathcal{T}_b$ phase, a new variational code $\mathbf{z}^{\mathcal{S}_2} \in \mathbb{R}^z$ is extracted from the $\mathcal{S}_2$ stream, similar as Eq. (1) and omitted for simplicity. The two surrogate Gaussians that approximate the $\mathcal{S}_1$ and $\mathcal{S}_2$ distributions (from which $\mathbf{z}^{\mathcal{S}_1}$ and $\mathbf{z}^{\mathcal{S}_2}$ were drawn) are enforced to be identical, such that they can be deemed as the shared latent subspace that connects the old and new feature spaces. We intermediately reconstruct the $\mathcal{S}_1$ data representations from the shared surrogate statistics.

To make this process online, we propose a neural architectural design which can *learn* the optimal depth from data streams autonomously, starting from shallow and gradually turning to deep if more complex variational feature mapping relationships are required to be approximated. The more accurate this reconstructive mapping is approximated, the better the learner can leverage the old classifier trained on the $\mathcal{S}_1$ stream, and hence the higher the online classification accuracy can be obtained by ensembling the old and new classifiers. The details are in the next Section 4.

## 4 Our Approach

**Overview.** In a nutshell, our proposed OLD$^3$S approach can be conceptually framed in a learning objective as follows.

$$\min_{f_t, \phi} \sum_{\text{HBP}} \Big[ \sum_{t_1, t_b} \Big( \mathcal{L}_{\text{VI}}(\phi) + \mathcal{L}_{\text{REC}}(\phi) \Big) + \sum_{t_b, t_2} \mathcal{L}_{\text{CLF}}(f_t, \phi) \Big].$$

In this section, we scrutinize this learning objective in sequence. The variational inference loss $\mathcal{L}_{\text{VI}}$ and the reconstruction loss $\mathcal{L}_{\text{REC}}$ together determine how the shared latent subspace is learned, presented in Section 4.1. The classification loss $\mathcal{L}_{\text{CLF}}$ synopsizes how the old and new classifiers are ensembled to expedite convergence for better prediction performance in Section 4.2. We end this section by elaborating how this minimization problem is realized by an *elastic* neural network model that automatically adjusts its depth in an online, data-driven fashion in Section 4.3.

### 4.1 Variational Latent Subspace Discovery

To discover the latent subspace $\mathcal{Z}$, we employ the Variational Auto-Encoder (VAE) [81]–[83] to summarize the observed data instances into latent variational codes. As illustrated in Figure 3, two independent VAEs are established, trained by minimizing the loss term:

$$\mathcal{L}_{\text{VI}}^{\{\mathcal{S}_1, \mathcal{S}_2\}} = -\mathbb{E}_{Q(\mathbf{z}_t \mid \mathbf{x}_t)}\big[\log \mathrm{P}(\mathbf{x}_t \mid \mathbf{z}_t)\big] + KL\Big(Q(\mathbf{z}_t \mid \mathbf{x}_t) \parallel \mathrm{P}(\mathbf{z}_t)\Big), \quad (2)$$

where $t \in \mathcal{T}_1 \cup \mathcal{T}_b$ and $t \in \mathcal{T}_b$ for the VAEs on two feature spaces $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively.

**Intuition 1:** The physical meanings of minimizing Eq. (2) are as follows. i) Minimizing the first term equates to maximizing the data generation quality, namely, the likelihood that the original data observations can be decoded from the extracted latent codes. Let the tuple $(\text{Enc}, \text{Dec})$ denote the encoder and the decoder networks in a VAE, the first term encourages $\mathbf{x}_t \approx \text{Dec}(\mathbf{z}_t)$ where $\mathbf{z}_t = \text{Enc}(\mathbf{x}_t)$. ii) The second term gauges the Kullback-Leibler (KL)-divergence [82], [84] between the underlying posterior $Q(\mathbf{z}_t \mid \mathbf{x}_t)$ and the latent marginal $\mathrm{P}(\mathbf{z}_t) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. With the posterior calculated by Eq. (1), for the extracted latent code $\mathbf{z}_t$, we denote its $i$-th entry with $z_i$ and is drawn from a Gaussian with mean $\mu_i$ and variance $\sigma_i^2$. To make the variational inference differentiable, reparameterization is employed as $z_i = \mu_i + \sigma_i \cdot \zeta$ with $\zeta \sim \mathcal{N}(0, 1)$ being normal noises. A reconstruction loss is then imposed to regularize the two independently learned latent spaces, from which a shared latent feature subspace is discovered during the overlapping timespan $\mathcal{T}_b$:

$$\mathcal{L}_{\text{REC}} = \ell\left[\mathbf{x}_{t_b}^{\mathcal{S}_1}, \text{Dec}^{2,1}(\mathbf{z}_{t_b}^{\mathcal{S}_2})\right] + KL\Big(Q(\mathbf{z}_{t_b}^{\mathcal{S}_1} \mid \mathbf{x}_{t_b}^{\mathcal{S}_1}) \parallel Q(\mathbf{z}_{t_b}^{\mathcal{S}_2} \mid \mathbf{x}_{t_b}^{\mathcal{S}_2})\Big). \quad (3)$$

**Intuition 2:** In the first term of Eq. (3), a new decoder network $\mathrm{Dec}^{2,1}(\cdot)$ which takes in the latent code from $\mathcal{S}_2$ to reconstruct the data of $\mathcal{S}_1$ approximates our desired reconstructive mapping $\phi$. The second term gauges the KL-divergence between the posteriors that were independently drawn from different variational distributions. Minimizing this term encourages the different variational distributions – the surrogate Gaussians to have similar probability densities, as conceptually illustrated in the middle panel of Figure 3. We note that this term is asymmetric, where the variational density of $\mathcal{S}_2$ is required to resemble that of $\mathcal{S}_1$ but not the opposite. This makes an intuitive sense as the variational distributions of $\mathcal{S}_1$ have been learned from $\mathcal{T}_1$ over a long time horizon, which is more likely to yield an accurate approximation of the underlying data distribution than that from a much shorter $\mathcal{T}_b$ only.

The two losses in Eqs. (2) and (3) together discover the shared latent feature subspace $\mathcal{Z}$. In the subsequent $\mathcal{T}_2$ timespan in which only the $\mathcal{S}_2$ space can be observed, an arriving instance $\mathbf{x}_{t_2}$ is embedded into $\mathcal{Z}$ by its corresponding VAE as $\mathbf{z}_{t_2} = \mathrm{Enc}(\mathbf{x}_{t_2})$, from which a reconstructed data representation of the $\mathcal{S}_1$ space is decoded, *i.e.*, $\tilde{\mathbf{x}}_{t_2}^{\mathcal{S}_1} := \phi(\mathbf{x}_{t_2}) = \mathrm{Dec}^{2,1}(\mathbf{z}_{t_2})$. Compared with linear models, this VAE architecture tends to learn a complex nonlinear mapping relationship between $\mathcal{S}_1$ and $\mathcal{S}_2$, which is common in high-dimensional media streams. Hence, vanished features are better reconstructed with new features, and information embedded in the old classifier is better harvested to assist the weak new classifier. As a result, after the overlapping period $\mathcal{T}_2$, OLD$^3$S makes less errors in an online fashion.

### 4.2 Online Prediction with Ensembled Learners

Once the old features of $\mathcal{S}_1$ vanish, the learner $f_t$ is not likely to make accurate predictions on the arriving instances by relying on $\mathcal{S}_2$ solely. Let $f_t = \{f_t^{\mathcal{S}_1}, f_t^{\mathcal{S}_2}\}$ denote the learner at the beginning of $\mathcal{T}_b$ when $\mathcal{S}_2$ just emerges. As $\mathcal{T}_b$ is short, the $f_t^{\mathcal{S}_2}$ part of the learner corresponding to the new features of $\mathcal{S}_2$ have been trained with very few instances hence is not likely to converge. Relying on $f_t^{\mathcal{S}_2}$ to predict the instances in $\mathcal{T}_2$ would incur substantial regrets.

To aid, we leverage the old $f_t^{\mathcal{S}_1}$ part that has been trained with a much larger number of instances during $\mathcal{T}_1$. Thanks to the reconstructive mapping $\phi$ approximated by the VAEs in Section 4.1, we can realize an online ensemble classification to yield accurate predictions when $f_t^{\mathcal{S}_2}$ is not ready, defined as follows.

$$\mathcal{L}_{\mathrm{CLF}} := \ell(y_t, \hat{y}_t) = -\sum_{c=1}^{C} y_{t,c} \log(\hat{y}_{t,c}), \quad \forall t \in \mathcal{T}_b \cup \mathcal{T}_2, \tag{4}$$

$$\hat{y}_t = p \cdot f_t^{\mathcal{S}_1}(\tilde{\mathbf{x}}_t^{\mathcal{S}_1}) + (1-p) \cdot f_t^{\mathcal{S}_2}(\mathbf{x}_t), \quad \mathbf{x}_t \in \mathcal{S}_2, \tag{5}$$

where Eq. (4) employs cross-entropy loss function [3] to gauge the multi-class learning loss, with $y_{t,c}$ and $\hat{y}_{t,c}$ being the ground truth and predicted probability that $\mathbf{x}_t$ belongs to the $c$-th class, respectively.

**Intuition 3:** The idea behind Eq. (5) is to let the ensemble coefficient $p \in (0,1)$ decide the impacts of the observed $\mathbf{x}_t$ and its reconstructed version $\tilde{\mathbf{x}}_t^{\mathcal{S}_1}$ in making predictions. At the beginning of $\mathcal{T}_2$ when the feature space just evolved, the old classifier $f_t^{\mathcal{S}_1}$ should be largely helpful with large $p$. Over time, the value of $p$ decays because of two reasons 1) the new classifier $f_t^{\mathcal{S}_2}$ becomes stronger and 2) the old classifier $f_t^{\mathcal{S}_1}$ can be less useful due to the distribution drift. An updating strategy is needed to echo this intuitive process, where the new classifier takes over gradually as the old classifier conveys less discriminative power.

In this work, we update the ensemble coefficient with exponential experts [75], where the empirical risks of using the old and new classifiers to make independent predictions are accumulated as:

$$R_T^{\mathcal{S}_1} = \sum_{t=T_1+1}^{T_2} \ell\big(y_t, f_t^{\mathcal{S}_1}(\tilde{\mathbf{x}}_t^{\mathcal{S}_1})\big), \; R_T^{\mathcal{S}_2} = \sum_{t=T_1+1}^{T_2} \ell\big(y_t, f_t^{\mathcal{S}_2}(\mathbf{x}_t)\big). \tag{6}$$

The smaller the cumulative empirical risk is suffered, the better predictions the classifier makes, and hence the higher its corresponding coefficient is uplifted exponentially. The updating rule is defined as $p = e^{-\eta R_T^{\mathcal{S}_1}} / (e^{-\eta R_T^{\mathcal{S}_1}} + e^{-\eta R_T^{\mathcal{S}_2}})$, where $\eta$ is a tuned parameter with its value assignment discussed in Theorem 2 of Section 5.

### 4.3 Adaptive Model Depth Learning with HBP

With the reconstructive mapping and the ensemble prediction, the information conveyed by the unobserved $\mathcal{S}_1$ can be reaped to better the learning performance. The remaining problem is how to realize the mapping and the classifiers with models of appropriate depths that are most likely to produce the optimal solutions. Unfortunately, fixing such depths beforehand is impossible without prior knowledge of how the data streams evolve in the sample space (*e.g.*, distribution drift that may require classifiers with various discriminant power to avoid overfitting) and the feature space (*e.g.*, a diversity of feature mapping relationships requires VAEs with disparate architectures). As it is unrealistic to rely on human experts to provide such knowledge constantly over long timespans, this problem boils down to the desire of a model architecture that can *learn* the best depth from data autonomously.

To this end, we leverage the Hedge Backpropagation (HBP) [25], [43] mechanism to incorporate the model depth as a learnable semantic that shall be determined in a data-driven manner through optimization. Instead of evaluating the loss based on the output from the last network layer only (as most deep learning models do), the main idea of HBP is to evaluate the losses on *all* the intermediate hidden representations yielded from the network layers from shallow to deep. Specifically, given an overcomplete network with $L$ hidden layers in total, the output of the $l$-th encoder layer of the VAE is recursively denoted as $\mathbf{z}_t^{(l)} = \mathrm{Enc}^{(l)}\big(\mathbf{z}_t^{(l-1)}\big)$, with $\mathbf{z}_t^{(0)} = \mathbf{x}_t$, where $t \in \mathcal{T}_1 \cup \mathcal{T}_b$ and $t \in \mathcal{T}_b \cup \mathcal{T}_2$ for the VAEs corresponds to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. The objective of HBP is defined as follows.

$$\min_{\{\alpha^{(l)}\}_{l=1}^{L}} \sum_{l=1}^{L} \alpha^{(l)} \Bigg[ \sum_{t_1,t_b} \Big( \mathcal{L}_{\mathrm{VI}}^{(l)} + \mathcal{L}_{\mathrm{REC}}^{(l)} \Big) + \sum_{t_b,t_2} \mathcal{L}_{\mathrm{CLF}}^{(l)} \Bigg], \tag{7}$$

where the loss terms $\mathcal{L}_{\mathrm{VI}}^{(l)}$, $\mathcal{L}_{\mathrm{REC}}^{(l)}$, and $\mathcal{L}_{\mathrm{CLF}}^{(l)}$ are evaluated on $\mathbf{z}_t^{(l)}$ at the $l$-th layer as shown in Figure 3. In particular, 1) Evaluated by $\mathcal{L}_{\mathrm{VI}}^{(l)}$ is how well the latent code $\mathbf{z}_t^{(l)}$ can summarize the raw inputs with a surrogate Gaussian via

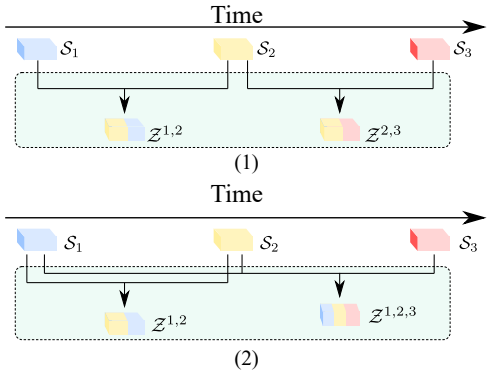



Fig. 4: Two straightforward solutions to build lifelong learners with multiple feature spaces, where $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ are three feature spaces appear in sequence along the time horizon. (1): a latent subspace shared by each new pair of observed feature spaces is learned, namely, $\mathcal{Z}^{1,2}$ for $\mathcal{S}_1$ and $\mathcal{S}_2$, and $\mathcal{Z}^{2,3}$ for $\mathcal{S}_2$ and $\mathcal{S}_3$. (2): a latent subspace is learned for all observed features, namely, $\mathcal{Z}^{1,2,3}$ for $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$.

using Eq. (2); For instances of $\mathcal{S}_1$, it is evaluated over $\mathcal{T}_1$ and $\mathcal{T}_b$ timespans, and for instances of $\mathcal{S}_2$, it is evaluated over $\mathcal{T}_b$ only. 2) Evaluated by $\mathcal{L}_{\text{REC}}^{(l)}$ is how precisely the reconstructive mapping is learned so that the $\mathcal{S}_1$ feature space can be reconstructed from the data instances of $\mathcal{S}_2$ via using Eq. (3); It is only evaluated during the overlapping phase $\mathcal{T}_b$ where $\mathcal{S}_1$ and $\mathcal{S}_2$ coexist. 3) Evaluated by $\mathcal{L}_{\text{CLF}}^{(l)}$ is how accurately the ensemble of both old and new classifiers can make online predictions via using Eq. (4); It is evaluated during $\mathcal{T}_b$ and $\mathcal{T}_2$ as the ensemble prediction is used only if the features of $\mathcal{S}_2$ become observed.

**Intuition 4:** The crux of HBP lies in finding the equilibrium that minimizes the three loss terms in Eq. (7) into a Pareto optimum. To do this, we update the hedge weight $\alpha^{(l)}$ that determines the impact of the $l$-th layer in a boosting fashion [85]: $\alpha_{t+1}^{(l)} \leftarrow \text{Norm}\big(\alpha_t^{(l)} \beta^{\mathcal{L}_{\text{VI+REC+CLF}}^{(l),t}}\big)$, where $\beta \in (0,1)$ is a discounting rate and $\mathcal{L}_{\text{VI+REC+CLF}}^{(l),t}$ accumulates the three losses in Eq. (7) suffered at the $t$-th round. Denoted by $\text{Norm}(\cdot)$ is a normalization function that reweighs each $\alpha^{(l)}$ by the sum of all $L$ layers, ensuring $\alpha^{(l)} \in (0,1)$. The idea is straightforward: the layer of which the output incurs large losses should be penalized and takes a discounted weight in the next round. Otherwise, if a layer is in an *optimal* depth, it approaches the minimizer of Eq. (7) with the incurred losses very small, such that the remaining layers (*i.e.*, those deeper than this hidden layer) cannot identify and learn meaningful gradient directions. Their hedge weights would stay in small values. Hence, when a few instances arrive in at the beginning, only the first several layers are activated with larger weight parameters. The shallow model converges faster than a fixed-depth model. With more instances observed, the requirement of higher learning capacity gradually assigns larger parameters to the deeper layers, which generates better data representations. The elastic model can reduce the number of prediction errors at both early and later periods in an online fashion.

### 4.4 Lifelong Learning from Streaming Feature Spaces

Thus far, we have explored the problem of discovering the variational latent subspace $\mathcal{Z}$ between two feature spaces $\mathcal{S}_1$ and $\mathcal{S}_2$. A natural generalize of it would be to learn multiple disparate feature spaces that evolve and appear in a lifelong manner. As delineated in Section 3.1, given a new feature space $\mathcal{S}_3$ that evolved from $\mathcal{S}_2$, a seemingly plausible solution is to repeat the entire learning procedure detailed in Sections 4.1, 4.2, and 4.3, in order to deduce another shared latent space. To clarify, we denote this newly learned space between $\mathcal{S}_2$ and $\mathcal{S}_3$ as $\mathcal{Z}^{2,3}$, and the previously learned space as $\mathcal{Z}^{1,2}$. Figure 4(1) conceptually illustrates this solution.

Despite straightforward, this solution suffers from two prominent drawbacks. First, it incurs high computational cost. Assume, for showcase, a sequence of evolving feature spaces denoted as $\{\mathcal{S}_n \mid n = 1, 2, \ldots, N\}$; the learning process will be repeated $N-1$ times, and all features appeared in the middle are learned twice (*e.g.*, $\mathcal{S}_2$ has been learned twice for the pairs of spaces ($\mathcal{S}_1$, $\mathcal{S}_2$) and ($\mathcal{S}_2$, $\mathcal{S}_3$) repeatedly). Second, after learning all $N$ spaces, only one shared latent space $\mathcal{Z}^{N-1,N}$ is resultant, whereas the previously learned knowledge is missed out along with the vanished feature spaces – a phenomenon coined as *catastrophic forgetting* in the lifelong/continual learning literature [25], [86], [87]. In our case, if a data instance described by any of the vanished feature spaces (*i.e.*, $\mathcal{S}_1, \ldots, \mathcal{S}_{N-2}$) arrives, the online learner is most likely to make erroneous predictions.

To aid the forgetting issue, one may think to train a model that learns from all observed features once a new feature space emerges, as shown in Figure 4(2). Namely, once $\mathcal{S}_3$ emerges, the instances that are described by $\mathcal{S}_2$ and $\mathcal{S}_3$ are aligned with those instances described by $\mathcal{S}_1$ and $\mathcal{S}_2$ to co-train a shared latent space $\mathcal{Z}^{1,2,3}$, that harmonizes information from all three feature spaces $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$. Alas, this idea is memory intensive, as it requires to store instances that are with all observed feature spaces. As in a doubly-streaming setting the new features constantly emerge without stopping, storing all instances would soon exhaust the memory and make the learning infeasible.

To counter the forgetting issue without bottlenecked by the memory constraint, we draw insight from the recent advances in [88] to tailor a lifelong learning extension of OLD$^3$S, named OLD$^3$S-L. The key idea of OLD$^3$S-L is to insist on using one shared latent space to harmonize all emerging feature spaces. Specifically, each feature space is assigned a VAE model to generate its own corresponding latent subspace firstly. In addition to the first subspace, we deem the recently learned latent space as a regularization term imposed on the newly emerged feature space, enforcing it to integrate the information in both the old and new data representations, as demonstrated in Figure 6. Suppose a new feature space $\mathcal{S}_3$ emerges, and now the learner starts to observe instances described by $\mathcal{S}_2$ and $\mathcal{S}_3$ in a short timespan, namely $\mathbf{x}'_{t_b} \in \mathcal{S}_2 \times \mathcal{S}_3 \subseteq \mathbb{R}^{d_2+d_3}$. The regularizer imposed on Eq. (3) is devised as follows,

$$\Omega(\mathcal{Z}^{1,2}, \mathcal{S}_3) = \ell\left[\mathbf{x}'^{\mathcal{S}_2}_{t_b}, \text{Dec}(\mathbf{z}'^{\mathcal{S}_3}_{t_b})\right] + KL\Big(Q(\mathbf{z}'^{\mathcal{S}_2}_{t_b} \mid \mathbf{x}'^{\mathcal{S}_2}_{t_b}) \parallel Q(\mathbf{z}'^{\mathcal{S}_3}_{t_b} \mid \mathbf{x}'^{\mathcal{S}_3}_{t_b})\Big). \tag{8}$$

where $\text{Dec}(\cdot)$ here refers to a decoder combining knowledge of all three feature spaces, $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$. Then we can reconstruct the data based on the subspace $\mathcal{Z}^{1,2,3}$. The

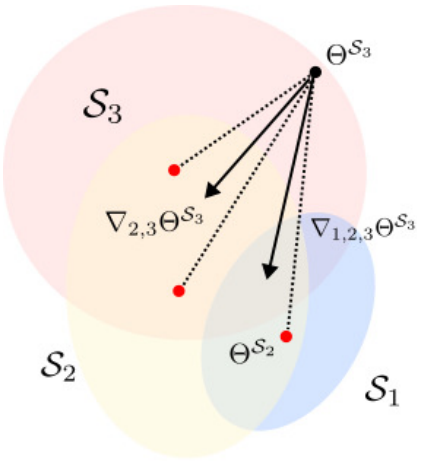



Fig. 5: Illustration of how the new OLD$^3$S-L leverages the vanished feature space ($\mathcal{S}_1$) to regularize its updating step (along a negative gradient direction). The three contours conceptually represent the parameter spaces from which the optimal VAEs corresponding to the feature spaces $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ are searched. The red points are the centers of contours, namely, the optima. $\Theta^{\mathcal{S}_3}$ denotes the (random) initialization of the VAE model learning latent representation of the new feature space $\mathcal{S}_3$. $\nabla_{1,2,3}\Theta^{\mathcal{S}_3}$ and $\nabla_{2,3}\Theta^{\mathcal{S}_3}$ demonstrate the updating steps of OLD$^3$S-L (which retains the knowledge from $\mathcal{S}_1$) and OLD$^3$S (which forgets $\mathcal{S}_1$ and focusing on $\mathcal{S}_2$ and $\mathcal{S}_3$ only), respectively.

process how the third model VAE$^{\mathcal{S}_3}$ generates a shared latent subspace by constraining its parameters is in detail illustrated as Figure 5, where $\Theta^{\mathcal{S}_3}$ is its initial parameter in a parameter space which expands with the appearance of the newly feature space and model.

First, the gradient update on the parameters of VAE$^{\mathcal{S}_3}$ is only constrained by the newly parameter space it causes. Because the first step of a VAE model is to capture a subspace only containing the information of the feature space it belongs to as described in Figure 6. For the straightforward method described in Figure 4(1), to include knowledge of $\mathcal{S}_2$, the initial parameter $\Theta^{\mathcal{S}_3}$ starts to be constrained by the second parameter space as well, and gradually approaches the optimal point where the model could express information of both $\mathcal{S}_2$ and $\mathcal{S}_3$. Since information contained in $\mathcal{S}_1$ is not considered, $\Theta^{\mathcal{S}_3}$ will move away from the space caused by $\mathcal{S}_1$. All above constraints determine the corresponding gradient optimization direction $\nabla_{2,3}\Theta^{\mathcal{S}_3}$ together. Similarly, the second model VAE$^{\mathcal{S}_2}$ can learn its shared latent subspace and find its optimal point represented as $\Theta^{\mathcal{S}_2}$.

Our method OLD$^3$S-L decides its gradient direction $\nabla_{1,2,3}\Theta^{\mathcal{S}_3}$ to harmonize all emerging feature spaces in another way. Specifically, under the same constraint of the newly parameter space, the initial parameter $\Theta^{\mathcal{S}_3}$ is updated to approach $\Theta^{\mathcal{S}_2}$. Since $\Theta^{\mathcal{S}_2}$ is achieved by the constraints of the first two spaces, the constraint information will be transferred back with the approximation of it and parameters of VAE$^{\mathcal{S}_3}$ are constrained by three parameter spaces gradually. The subspace integrate information from $\mathcal{S}_1$ and $\mathcal{S}_2$ brought by $\Theta^{\mathcal{S}_2}$ with itself. Therefore, the initial point $\Theta^{\mathcal{S}_3}$ will be updated along the direction $\nabla_{1,2,3}\Theta^{\mathcal{S}_3}$ to move closer to

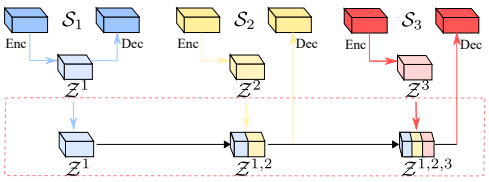



Fig. 6: Illustration of the proposed OLD$^3$S-L, where one latent subspace is learned in an incremental manner to harmonize information from new feature spaces that arrive continuously like a stream. The novelty lies in that it does not require to store all observed instances that were described by the vanished features.

both $\mathcal{S}_1$ and $\mathcal{S}_2$. The optimal parameter of VAE$^{\mathcal{S}_3}$ will arrive in a position where all feature spaces are harmonized.

Note, our OL task possesses different focus over continual learning (CL) in terms of performing predictions on a data sequence. In CL, the overarching goal is to retain knowledge from all seen tasks, where each task contains a data subset with unique distribution. Thus, a CL learner trained later on must be able to make accurate predictions on instances from any previous task anytime. In OL, on the contrary, the prediction is conducted in a one-pass setting, where the learner strives to be accurate on newly arriving instances with minimized regret. As a result, OL learners do not necessarily keep their performance on past data over long time spans. More specifically, our OLD$^3$S-L setting differs from traditional CL in two aspects. First, OLD$^3$S-L and CL tend to deal with disparate data evolving patterns, where CL mainly counters against distribution drift and OLD$^3$S-L copes with feature space evolution. Second, our OLD$^3$S-L focuses on assisting the weak learner initialized for new feature space with knowledge learned from the vanished feature spaces in an ensemble manner. Thus, the shared latent subspace strives to capture knowledge from learned data instances being informative for building predictive models in new feature space. Revisit the example shown in Figure 4(2): we hypothesize that our OLD$^3$S-L can enjoy a better performance on the new $\mathcal{S}_3$ space if $\mathcal{S}_3$ is evolved from $\mathcal{S}_1$ and $\mathcal{S}_2$ on a space continumum hence is correlated with $\mathcal{S}_1$; otherwise if $\mathcal{S}_3$ and $\mathcal{S}_1$ become independent, the OLD$^3$S-L variant boils down to OLD$^3$S as the latent subspace only establishing relationship between $\mathcal{S}_2$ and $\mathcal{S}_3$ would suffice to yield good prediction performance in $\mathcal{S}_3$. We validate this hypothesis with empirical study, with results documented in **Q7** of Section 6.2.

## 5 Theoretical Analysis

We analyze the theoretical properties of the proposed OLD$^3$S approach, aiming to answer two research questions.

**Q1.** *How does the learning performance of our OLD$^3$S learner compare to the learners of arbitrary depths?*

This question naturally arises as, at the first glance, our OLD$^3$S learner uses the outputs of all hidden layers in a weighted combination, wherein several layers may yield less expressive latent codes (*e.g.*, the too shallow or too deep layers) to detriment the learning performance. Although such layers are discounted due to their inferiority, it is

important to know what is the cost of using HBP to learn the optimal model depth. The answer is provided as:

**Theorem 1.** *Over $T$ rounds, our OLD$^3$S suffers cumulative loss:*

$$\mathcal{L}_{\text{OLD}^3\text{S}} \leq C_\beta \cdot \min_{l^\star} \Big\{ \sum_{t=1}^{T} \mathcal{L}^{(l),t}_{\text{VI+REC+CLF}} \Big\}_{l=1}^{L} + \frac{\ln L}{1-\beta}, \tag{9}$$

*where $C_\beta = \ln(1/\beta)/(1-\beta) > 0$ is a monotonically decreasing scalar.*

***Proof:*** Before our analysis, two loss formulas need to be introduced. For $l$-th layer, its cumulative loss over T rounds is defined as: $\mathcal{L}^{(l)}_{\text{VI+REC+CLF}} = \sum_{t=1}^{T} \ell_t^{(l)}$. And the loss is denoted as $\mathcal{L}^{(l)}$ in the follows for simplicity.

Similarly, let $\mathcal{L}_{\text{OLD}^3\text{S}} = \sum_{t=1}^{T} \sum_{l=1}^{L} p_t^{(l)} \ell_t^{(l)} = \sum_{t=1}^{T} \mathbf{p}_t \cdot \ell_t$ as the total cumulative loss suffered by our algorithm. Different from the formal content, a new parameter $p$ is introduced to represent Norm$(\cdot)$, where $\mathbf{p}_t$ is a vector including parameters $p$ of all layers, and allocated by corresponding weight parameter $\boldsymbol{\alpha_t} = [\alpha_t^{(1)}, \ldots, \alpha_t^{(L)}]^\top$. In particular, the relationship between them and the updating rule of parameter $\alpha$ is described as follows.

$$\alpha_{t+1}^{(l)} = \alpha_t^{(l)} \cdot \beta^{\ell_t^{(l)}}, \quad \mathbf{p}_t = \frac{\boldsymbol{\alpha_t}}{\sum_{l=1}^{L} \alpha_t^{(l)}}. \tag{10}$$

To conduct further proof, we here introduce:

**Lemma 1** (Freund and Schapire, 1997 [85]). *$\beta^r \leqslant 1-(1-\beta)$, for $\beta \geqslant 0$ and $r \in [0,1]$.*

Then, combined with Eq. (10) and Lemma 1, this implies

$$\begin{aligned} \sum_{l=1}^{L} \alpha_{t+1}^{(l)} &= \sum_{l=1}^{L} \alpha_t^{(l)} \beta_i^{\ell_t^{(l)}}, \\ &\leqslant \sum_{l=1}^{L} \alpha_t^{(l)} \left(1-(1-\beta)\ell_t^{(l)}\right), \\ &= \Big(\sum_{l=1}^{L} \alpha_t^{(l)}\Big) \left(1-(1-\beta)\mathbf{p}_t \cdot \ell_t\right). \end{aligned} \tag{11}$$

Applying repeatedly for $t = 1, \ldots, T$ yields

$$\begin{aligned} \sum_{l=1}^{L} \alpha_{T+1}^{(l)} &\leqslant \prod_{t=1}^{T} \left(1-(1-\beta)\mathbf{p}_t \cdot \ell_t\right), \\ &\leqslant \exp\Big(-(1-\beta)\sum_{t=1}^{T} \mathbf{p}_t \cdot \ell_t\Big), \\ &= \exp\Big(-(1-\beta)\mathcal{L}_{\text{OLD}^3\text{S}}\Big), \end{aligned}$$

since $1+x \leqslant e^x$ for all $x$ and $\mathcal{L}_{\text{OLD}^3\text{S}} = \sum_{t=1}^{T} \mathbf{p}_t \cdot \ell_t$. Then, we get the follow formula,

$$\begin{aligned} \ln\Big(\sum_{l=1}^{L} \alpha_{T+1}^{(l)}\Big) &\leqslant \ln \exp\left(-(1-\beta)\mathcal{L}_{\text{OLD}^3\text{S}}\right), \\ \mathcal{L}_{\text{OLD}^3\text{S}} &\leqslant \frac{-\ln\left(\sum_{l=1}^{L} \alpha_{T+1}^{(l)}\right)}{1-\beta}. \end{aligned} \tag{12}$$

Next, going back to Eq. (10),

$$\begin{aligned} \alpha_{T+1}^{(l)} &= \alpha_1^{(l)} \prod_{t=1}^{T} \beta^{\ell_t^{(l)}}, \\ &= \alpha_1^{(l)} \beta^{\mathcal{L}^{(l)}}, \end{aligned} \tag{13}$$

and for all layers, we get,

$$\begin{aligned} \sum_{l=1}^{L} \alpha_{T+1}^{(l)} &= \sum_{l=1}^{L} \alpha_1^{(l)} \beta^{\mathcal{L}^{(l)}}, \\ &\geqslant \beta^{\max_l \in L \mathcal{L}^{(l)}} \sum_{l=1}^{L} \alpha_1^{(l)}. \end{aligned} \tag{14}$$

By now, all preparations for analyzing Theorem 1 are complete. Combined Eq. (13) and Eq. (14),

$$\mathcal{L}_{\text{OLD}^3\text{S}} \leqslant \frac{-\ln\left(\sum_{l\in L} \alpha_1^{(l)}\right) - (\ln\beta)\max_{l\in L} \mathcal{L}^{(l)}}{1-\beta}. \tag{15}$$

This is a general bound statement where all layers are be considered. For any $l \in \{1, \ldots, L\}$, we achieve a special case:

$$\mathcal{L}_{\text{OLD}^3\text{S}} \leqslant \frac{-\ln \alpha_1^{(l)} - \mathcal{L}^{(l)} \ln\beta}{1-\beta}. \tag{16}$$

The bound 16 state that our OLD$^3$S only perform a little bit worse than the best $l$-th layer among the sequence. The difference lies in the choice of $\beta$ and the initial weight $\alpha_1^{(l)}$ of each layer. If every weight is set equally such that $\alpha_1^{(l)} = 1/L$, then this bound becomes:

$$\mathcal{L}_{\text{OLD}^3\text{S}} \leqslant \frac{\min_l \mathcal{L}^{(l)} \ln(1/\beta) + \ln L}{1-\beta}. \tag{17}$$

The bound given in Eq. (17) can be written as:

$$\mathcal{L}_{\text{OLD}^3\text{S}} \leq C_\beta \cdot \min_{l^\star} \Big\{ \sum_{t=1}^{T} \mathcal{L}^{(l),t}_{\text{VI+REC+CLF}} \Big\}_{l=1}^{L} + \frac{\ln L}{1-\beta},$$

where $C_\beta = \ln(1/\beta)/(1-\beta) > 0$ as stated in Theorem 1. □

*Remark 1.* A hindsight optimal model of which the *optimal* depth $l^\star$ that yields the least learning loss over $T$ rounds, presented at the RHS of Eq. (5), provides a natural upper bound of our OLD$^3$S model. Theorem 1 suggests that our model is comparable to this optimal model (*cf.* $\lim_{\beta\to 1} C_\beta = 1$, $\ln L/(1-\beta) < 0$). As in practice the optimal $l^\star$ is unknown and can vary according to datasets, it is not realistic to conduct a set of experiments to decide the optimal depth for each dataset. Instead, our HBP method can help the model automatically learn the optimal depth at each round and achieve the comparable cumulative loss. Hence, our OLD$^3$S learner strictly enjoys a lower online learning loss than any neural network models with their depth fixed in ad-hoc, *i.e.*, **Q1** answered.

**Q2.** *How helpful is the ensemble learning method of our OLD$^3$S that reconstructs the old feature space $\mathcal{S}_1$, compared of using $\mathcal{S}_1$ or $\mathcal{S}_2$ independently?*

We derive the risk bound to analyse the asymptotic property of our OLD$^3$S approach. Specifically, in timespans $\mathcal{T}_b$ and $\mathcal{T}_2$ when features from $\mathcal{S}_2$ appear:

**Theorem 2.** *With $\eta = 8\sqrt{1/\ln T}$ and $T = |\mathcal{T}_b \cup \mathcal{T}_2|$, we have*

$$\mathbb{E}_{t\sim T}[\ell(y_t, \hat{y}_t)] \leq \frac{1}{T}\min\{R_T^{\mathcal{S}_1}, R_T^{\mathcal{S}_2}\} + \frac{1}{\sqrt{\ln T}} + \frac{\ln 2}{8T}\sqrt{\ln T}, \tag{18}$$

*where $R_T^{\mathcal{S}_1}$ and $R_T^{\mathcal{S}_2}$ denote the cumulative risks of using independent $\mathcal{S}_1$ and $\mathcal{S}_2$ classifiers over $T$ rounds as defined in Eq. (6).*

***Proof:*** Here, we define $R_T^{\mathcal{S}_1} = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t^{S_1})$, and $R_T^{\mathcal{S}_2} = \sum_{t=1}^{T} \ell(y_t, \hat{y}_t^{S_2})$ with the same parameter $T = |\mathcal{T}_b \cup \mathcal{T}_2|$.

Then, a quantitative variable $Q_T$ is introduced as:

$$Q_T = \exp\left(-\eta R_T^{\mathcal{S}_1}\right) + \exp\left(-\eta R_T^{\mathcal{S}_2}\right), \tag{19}$$

and it is easy to verify that $Q_1 = e^0 + e^0 = 2$ because there is no loss for both classifiers in the first round.

Over $T$ iterations, we have

$$\begin{aligned}
&\ln\left(\frac{Q_T}{Q_1}\right) \\
&= \ln\left(\exp\left(-\eta R_T^{\mathcal{S}_1}\right) + \exp\left(-\eta R_T^{\mathcal{S}_2}\right)\right) - \ln 2, \\
&\geq \ln\left(\max\left\{\exp\left(-\eta R_T^{\mathcal{S}_1}\right), \exp\left(-\eta R_T^{\mathcal{S}_2}\right)\right\}\right) - \ln 2, \\
&= \max\left\{-\eta R_T^{\mathcal{S}_1}, -\eta R_T^{\mathcal{S}_2}\right\} - \ln 2, \\
&= -\eta \min\left\{R_T^{\mathcal{S}_1}, R_T^{\mathcal{S}_2}\right\} - \ln 2.
\end{aligned} \tag{20}$$

At the $T^{\text{th}}$ iteration, we have

$$\begin{aligned}
&\ln\left(\frac{Q_T}{Q_{T-1}}\right) \\
&= \ln\left(\frac{\exp\left(-\eta R_T^{\mathcal{S}_1}\right) + \exp\left(-\eta R_T^{\mathcal{S}_2}\right)}{\exp\left(-\eta R_{T-1}^{\mathcal{S}_1}\right) + \exp\left(-\eta R_{T-1}^{\mathcal{S}_2}\right)}\right), \\
&= \ln\left(\frac{\exp\left(-\eta(R_{T-1}^{\mathcal{S}_1} + \ell_T^{\mathcal{S}_1})\right) + \exp\left(-\eta(R_{T-1}^{\mathcal{S}_2} + \ell_T^{\mathcal{S}_2})\right)}{\exp\left(-\eta R_{T-1}^{\mathcal{S}_1}\right) + \exp\left(-\eta R_{T-1}^{\mathcal{S}_2}\right)}\right), \\
&= \ln\left(p\exp\left(-\eta\ell_T^{\mathcal{S}_1}\right) + (1-p)\exp\left(-\eta\ell_T^{\mathcal{S}_2}\right)\right),
\end{aligned} \tag{21}$$

where $\ell_T^{\mathcal{S}_1} = \ell\left(y_T, \hat{y}_T^{\mathcal{S}_1}\right)$ and $\ell_T^{\mathcal{S}_2} = \ell\left(y_T, \hat{y}_T^{\mathcal{S}_2}\right)$ denote the instantaneous losses suffered by making predictions on $x_T$ and $\tilde{x}_T$ at the $T^{\text{th}}$ iteration, respectively. And,

$$p = \frac{\exp\left(-\eta R_T^{\mathcal{S}_1}\right)}{\exp\left(-\eta R_T^{\mathcal{S}_1}\right) + \exp\left(-\eta R_T^{\mathcal{S}_2}\right)} \in [0, 1]$$

is a defined parameter.

To conduct further proof, we here introduce the Hoeffding Inequality:

**Lemma 2** (Hoeffding Inequality [75])**.** *Let $X$ be a random variable with $a \leq X \leq b$. Then for any $s \in \mathbb{R}$,*

$$\ln \mathbb{E}(e^{sX}) \leq s\mathbb{E}X + \frac{s^2(b-a)^2}{8}.$$

Suppose the instantaneous losses are normalized at each iteration, s.t. $\forall \ell_t^{\mathcal{S}_1}, \ell_t^{\mathcal{S}_2} \in [0, 1]$ can be relaxed to:

$$\begin{aligned}
&\leq -\eta\left(p\ell_T^{\mathcal{S}_1} + (1-p)\ell_T^{\mathcal{S}_2}\right) + \frac{\eta^2}{8}, \\
&\leq -\eta\left[\ell\left(y_T, p\hat{y}_T^{\mathcal{S}_1} + (1-p)\hat{y}_T^{\mathcal{S}_2}\right)\right] + \frac{\eta^2}{8}, \\
&= -\eta\ell\left(y_T, \hat{y}_T\right) + \frac{\eta^2}{8},
\end{aligned} \tag{22}$$

based on the convexity of the loss function $\ell(\cdot)$.

Again, over $T$ iterations, we have:

$$\begin{aligned}
&\ln\left(\frac{Q_T}{Q_{T-1}}\right) + \ln\left(\frac{Q_{T-1}}{Q_{T-2}}\right) + \ldots + \ln\left(\frac{Q_2}{Q_1}\right), \\
&= \ln\left(\frac{Q_T}{Q_{T-1}} \cdot \frac{Q_{T-1}}{Q_{T-2}} \cdot \ldots \cdot \frac{Q_2}{Q_1}\right) = \ln\left(\frac{Q_T}{Q_1}\right), \\
&\leq -\eta\left[\ell\left(y_T, \hat{y}_T\right) + \ell\left(y_{T-1}, \hat{y}_{T-1}\right) \ldots + \ell\left(y_1, \hat{y}_1\right)\right] + \frac{\eta^2}{8}T, \\
&= -\eta\sum_{t=1}^{T}\ell\left(y_t, \hat{y}_t\right) + \frac{\eta^2}{8}T.
\end{aligned} \tag{23}$$

Chaining the inequalities Eq. (20) and Eq. (23) yields:

$$\sum_{t=1}^{T}\ell\left(y_t, \hat{y}_t\right) \leq \min\left\{R_T^{\mathcal{S}_1}, R_T^{\mathcal{S}_2}\right\} + \frac{\eta}{8}T + \frac{\ln 2}{\eta}. \tag{24}$$

Now, the bound is decided by the value of $\eta$. Specifically, with $\eta = 8\sqrt{1/\ln T}$, the upper bound becomes:

$$\sum_{t=1}^{T}\ell\left(y_t, \hat{y}_t\right) \leq \min\left\{R_T^{\mathcal{S}_1}, R_T^{\mathcal{S}_2}\right\} + T/\sqrt{\ln T} + (\ln 2/8)\sqrt{\ln T}. \tag{25}$$

Now, the Theorem 2 holds immediately. □

*Remark 2.* The LHS of Eq. (18) provides the empirical prediction risks of our OLD$^3$S approach, upper-bounded by Theorem 2 with a sub-linear slackness $\Theta(\sqrt{T}/T) = 1/\sqrt{\ln T} + (\ln 2/8T)\sqrt{\ln T}$. We can verify that $\lim_{T\to\infty}\Theta(\sqrt{T}/T) = 0$, which suggests that our OLD$^3$S with ensembling outperforms an independent learner trained on $\mathcal{S}_1$ or $\mathcal{S}_2$, whichever yields smaller risk, *i.e.*, **Q2** answered.

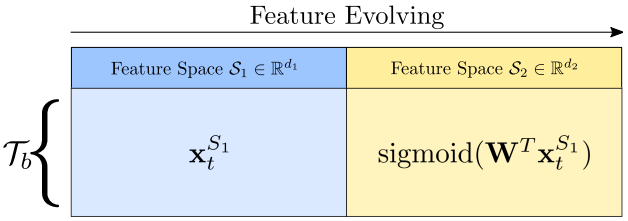



Fig. 7: Illustration of Evolving Features for Tabular Datasets.

TABLE 1: Statistics of the 10 datasets. $|\mathcal{S}_1|$ and $|\mathcal{S}_2|$ are the dimensions of the old and new feature spaces, respectively.

| No. | Dataset | # Samples | $\lvert\mathcal{S}_1\rvert$ | $\lvert\mathcal{S}_2\rvert$ | # Classes |
|---|---|---|---|---|---|
| 1 | magic04 | 36,119 | 10 | 30 | 2 |
| 2 | adult | 61,559 | 14 | 30 | 2 |
| 3 | EN-FR | 34,758 | 21,531 | 24,892 | 6 |
| 4 | EN-IT | 34,758 | 21,531 | 15,506 | 6 |
| 5 | EN-SP | 34,758 | 21,531 | 11,547 | 6 |
| 6 | FR-IT | 49,648 | 24,893 | 15,503 | 6 |
| 7 | FR-SP | 49,648 | 24,893 | 11,547 | 6 |
| 8 | CIFAR | 95,000 | 3072 | 3072 | 10 |
| 9 | Fashion | 114,000 | 784 | 784 | 10 |
| 10 | SVHN | 139,257 | 3072 | 3072 | 10 |

## 6 Experiments

Empirical results are presented to verify the viability and effectiveness of our OLD$^3$S approach. Detailed experimental

setups are elaborated in Section 6.1 and the results and findings are extrapolated in Section 6.2.

## 6.1 Evaluation Setup

### 6.1.1 Dataset Preparation

We benchmark our OLD$^3$S approach on 10 real-world datasets covering three domains to verify its versatility. Statistics of the studied datasets are summarized in Table 1.

• ***UCI Data Science (No. 1-2):*** The two datasets have only one feature space $\mathcal{S}_1$ at first, and we artificially create a new feature space $\mathcal{S}_2 = \text{sigmoid}(\mathbf{W}^\top \mathcal{S}_1)$ with a random Gaussian $\mathbf{W}$ and a nonlinear sigmoid function. Thus, the original feature space evolved with a nonlinear relationship during the overlapping period as shown in Figure 7. The two feature spaces are concatenated as the shape shown in Figure 1 to simulate the doubly-streaming data.

• ***Multilingual Text Categorization (No. 3-7):*** A set of documents are described by four languages including English (EN), French (FR), Italian (IT), and Spanish (SP). By treating each document as a bag of words (features), the vocabulary of each language can be deemed as a feature space. At each time, a document is presented and our model aims to classify it into one of the six categories. To simulate doubly-streaming, the language describing the documents shifts over time, *e.g.*, EN-FR, where the model learned to classify English documents is soon presented with French after a short overlapping $\mathcal{T}_b$ timespan, requiring to approximate the translation relationship between languages. To exacerbate the non-linearity of the mapping between two languages, we apply the sigmoid function on the $\mathcal{S}_2$ space.

• ***Online Image Classification (No. 8-10):*** Images are typical media data of high dimensionality and low information density. To simulate doubly-streaming data, we follow the preprocessing steps suggested by [25], [89] to create an evolved space by transforming the original images with various spectral-mapping, shearing, rescaling, and rotating. Images are presented one at a step, and the model needs to learn the complex pixel transformation online.

### 6.1.2 Dataset Visualization

We visualize data distributions of the first 1,000 instances with label 0 and 1 from $\mathcal{S}_1$ and $\mathcal{S}_2$ in the overlapping period $\mathcal{T}_b$ via T-SNE [90], as shown in Figure 8. Red and blue points indicate Label 0 and 1 in $\mathcal{S}_1$ respectively; Green and yellow are for 0 and 1 in $\mathcal{S}_2$, respectively. We can observe that the evolving data distribution of $\mathcal{S}_2$ deviates slightly from the original distribution of $\mathcal{S}_1$, as shown in Figure 8e and 8f. Some of datasets even display a disjoint distribution for $\mathcal{S}_2$, as shown in Figure 8a and 8b. A model learned from the old feature space $\mathcal{S}_1$ will present poor performance for data with new distribution from $\mathcal{S}_2$. Only for *FashionMNIST* shown in Figure 8j, it seems feasible to learn one classifier for data from both spaces. While only observing instances from $\mathcal{S}_1$, the large margin between red and blue points would allow multiple optimal classifiers for $\mathcal{S}_1$. However, most of these classifiers become sub-optimal for $\mathcal{S}_2$ when the distance between green and yellow instances decreases, thereby leading the poor generalization performance as well. As a result, the models learned from $\mathcal{S}_1$ can not be directly used for data with different distribution from $\mathcal{S}_2$ , without establishing a mapping relationship.

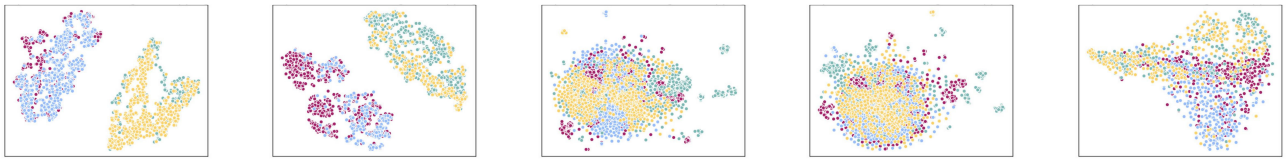
We take the dataset *adult* as an example to show how our method approximates two disjoint distributions in the latent subspace, as shown in Figure 9. Figure 9a illustrates the original data distribution, where red and blue points indicate Label 0 and 1 in $\mathcal{S}_1$, respectively; Green and yellow points indicate Label 0 and 1 in $\mathcal{S}_2$, respectively. We can observe that instances from different feature spaces display a disjoint distribution. As discussed above, the model learned from $\mathcal{S}_1$ cannot be used for $\mathcal{S}_2$ directly. Figure 9b, 9c, 9d, and 9e show the latent distributions of four hidden layers of well-trained VAEs, respectively. As the model goes deeper, we can observe that yellow points are far away from red points and start to approach blue points. This indicates that the model learns better shared subspaces, where the latent representations of instances with Label 1 are indeed approximating to each other, even though they reside in different feature spaces. As shown in Figure 9d, it becomes possible to use one classifier to make accurate predictions for both $\mathcal{S}_1$ and $\mathcal{S}_2$. HBP enables our elastic model to only output the prediction made with the optimal shared representation.

### 6.1.3 Compared Methods

Three state-of-the-art competitors tailored for processing double-streaming data are employed for comparative study, with their main ideas presented as follows. **FOBOS** [91] is a canonic online learning baseline that operates over first-order oracles with a projected subgradient that encourages sparse solutions. To make it work for doubly-streaming data, zeros are padded to the new features and vanished old features. **OLSF** [51] is the first study to tackle an incremental feature space, where new features constantly emerging are carried in all subsequent data instances. OLSF updates the online learners in a passive-aggressive fashion, where the learning coefficients of old features are re-weighed to new features only if these new features convey significant information that changes the decision boundary. **FESL** [16] is the pioneer work to deal with doubly-streaming data, which nevertheless employed linear functions to learn classifiers and to approximate a mapping relationship between feature spaces. A comparison with FESL rationalizes our design of adaptive deep learner and variational feature mapping approximator. For the ablation study, two variants of our OLD$^3$S approach are proposed, named OLD-Linear and OLD-FD. They differ from our original OLD$^3$S design by: 1) **OLD-Linear** employs a linear mapping to approximate the feature mapping relationship and 2) **OLD-FD** trains a deep neural network with a fixed depth. We craft the two variants to necessitate the designs of a non-linear, VI-based feature mapping approximator and the HBP that allows model depth to be learned from data autonomously.

### 6.1.4 Evaluation Metric

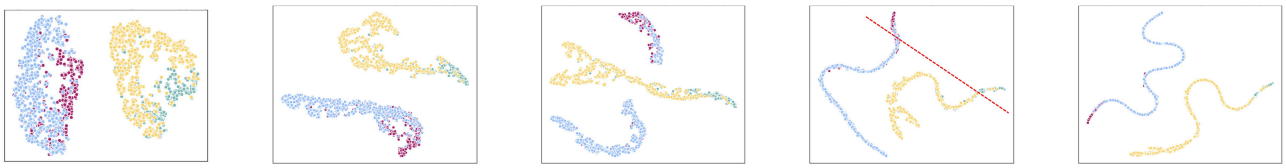
As the traditional classification accuracy is ill-conditioned in online learning, we employ the Online Classification Accuracy (**OCA**) and Averaged Cumulative Regret (**ACR**)



(a) magic04 (b) adult (c) Reuter-EN-FR (d) Reuter-EN-IT (e) CIFAR

(f) SVHN (g) Reuter-EN-SP (h) Reuter-FR-IT (i) Reuter-FR-SP (j) FashionMNIST

Fig. 8: The change of data distributions before and after the evolving. Red and Blue points indicate Label 0 and 1 in $\mathcal{S}_1$, respectively; Green and Yellow points are for Label 0 and 1 in $\mathcal{S}_2$, respectively.



(a) Original (b) Hidden 1 (c) Hidden 2 (d) Hidden 3 (e) Hidden 4

Fig. 9: Disjoint distribution of original data in *adult* dataset is approximated in the shared latent subspace. The latent representations of the third hidden layer (Hidden 3) is chosen to made the prediction by the elastic model. Red and Blue points indicate Label 0 and 1 in $\mathcal{S}_1$, respectively; Green and Yellow points are for Label 0 and 1 in $\mathcal{S}_2$, respectively.

TABLE 2: Comparative results of averaged cumulative regret (ACR $\pm$ mean variance) benchmarked on 10 datasets, where the lower the value, the better the method performs. The best results are bold. The bullet $\bullet$ indicates that our OLD$^3$S approach outperforms the competitors with a statistical significance supported by the *paired t-tests* at 95% confidence level.

| Dataset | FOBOS | OLSF | FESL | OLD-Linear | OLD-FD | OLD$^3$S |
|---|---|---|---|---|---|---|
| magic04 | .119 $\pm$ .022$\bullet$ | .335 $\pm$ .021$\bullet$ | .110 $\pm$ .016$\bullet$ | .075 $\pm$ .018 | .076 $\pm$ .021 | **.052 $\pm$ .017** |
| adult | .076 $\pm$ .064 | .225 $\pm$ .019$\bullet$ | .067 $\pm$ .044 | .055 $\pm$ .017 | .068 $\pm$ .018 | **.049 $\pm$ .019** |
| EN-FR | .326 $\pm$ .064$\bullet$ | .324 $\pm$ .018$\bullet$ | .345 $\pm$ .044$\bullet$ | .168 $\pm$ .030$\bullet$ | .137 $\pm$ .030$\bullet$ | **.068 $\pm$ .025** |
| EN-IT | .318 $\pm$ .060$\bullet$ | .314 $\pm$ .019$\bullet$ | .337 $\pm$ .040$\bullet$ | .197 $\pm$ .028$\bullet$ | .143 $\pm$ .033$\bullet$ | **.083 $\pm$ .024** |
| EN-SP | .302 $\pm$ .060$\bullet$ | .322 $\pm$ .021$\bullet$ | .335 $\pm$ .037$\bullet$ | .197 $\pm$ .036$\bullet$ | .136 $\pm$ .027$\bullet$ | **.077 $\pm$ .024** |
| FR-IT | .278 $\pm$ .047$\bullet$ | .301 $\pm$ .013$\bullet$ | .314 $\pm$ .037$\bullet$ | .195 $\pm$ .031$\bullet$ | .147 $\pm$ .030$\bullet$ | **.084 $\pm$ .026** |
| FR-SP | .272 $\pm$ .046$\bullet$ | .310 $\pm$ .014$\bullet$ | .336 $\pm$ .040$\bullet$ | .201 $\pm$ .029$\bullet$ | .155 $\pm$ .026 | **.102 $\pm$ .027** |
| CIFAR | .468 $\pm$ .017$\bullet$ | .504 $\pm$ .014$\bullet$ | .463 $\pm$ .013$\bullet$ | .166 $\pm$ .032 | .232 $\pm$ .038$\bullet$ | **.150 $\pm$ .030** |
| Fashion | .305 $\pm$ .033$\bullet$ | .294 $\pm$ .016$\bullet$ | .247 $\pm$ .023$\bullet$ | .160 $\pm$ .033$\bullet$ | .123 $\pm$ .019$\bullet$ | **.056 $\pm$ .015** |
| SVHN | .808 $\pm$ .011$\bullet$ | .604 $\pm$ .014$\bullet$ | .806 $\pm$ .011$\bullet$ | .144 $\pm$ .038$\bullet$ | .120 $\pm$ .025 | **.089 $\pm$ .018** |
| w/t/l | 9/1/0 | 10/0/0 | 9/1/0 | 7/3/0 | 6/4/0 | — |

to measure the performance. Specifically, they are defined:

$$\mathrm{OCA}(f_t) = 1 - \frac{1}{B}\sum_{i=t-B}^{t} [\![ y_i \neq f_t(\mathbf{x}_i) ]\!], \quad T = |\mathcal{T}_1 \cup \mathcal{T}_b \cup \mathcal{T}_2|$$

$$\mathrm{ACR} = \frac{1}{T}\sum_{t=1}^{T}\Big[\max_{f*}\mathrm{OCA}(f^*) - \mathrm{OCA}(f_t)\Big].$$

Intuitively, OCA dynamically measures the accuracy of a classifier $f_t$ at the $t$-th round, evaluated at the *most recent* $B$ instances. ACR evaluates how large the online learner regrets comparing to a hindsight optimum $f^*$ by accumulating the OCA differences between $f_t$ and $f^*$ over $T$ rounds. The smaller the value of ACR was, the less largely the learner regrets, and the better the online classification was.

#### 6.1.5 Experiment Setup

Tunable parameters in our paper can be divided into two types: one is updated automatically, and the other needs to be decided and fixed in advance.

- The ensemble coefficients $p$ and $1-p$ in Eq. (5) is used to balance the weights of old and new classifiers in the ensemble result. Initialized values are set as 0.5 to 0.5.

The value of $p$ is updated automatically according to the classifier's cumulative empirical risk defined as Eq. (6). The relative analysis and updating process can be found in Section 4.2.

- The choice of tuned parameter $\eta$ ensures that the ensemble method is better than using old or new classifier independently. We give its optimal value $\eta = 8\sqrt{1/\ln T}$ in Theorem 2 of Section 5, and it is in inverse proportion to the dataset size $T$.
- The hedge weight $\alpha^{(l)}$ of each layer is also updated automatically according to the cumulative loss of corresponding layer classifiers defined as Eq. (7). Besides, the sum of all hedge weights equals to 1 by a normalization function according to *Intuition 4*. Both can be found in Section 4.3.
- The discounting rate $\beta$ is used to control the updating rate of the hedge weight $\alpha^{(l)}$. When the value of $\beta$ approximates 1, our model is comparable to this optimal mode with fixed depth, discussed in Theorem 1 of Section 5. We did a grid search within the range of [0.3, .35, 0.5, .55, 0.7, .75, 0.9, .95], and 0.9 is chosen because of its best performance among most datasets.
- The optimizers are selected as SGD and Adam for datasets UCI Data No. 1-2, and all other datasets No. 3 - 10, respectively. The learning rate is set as 0.001 for both optimizers, and two coefficients in Adam used for computing the averages of gradient and its square is set as 0.9 and 0.999.
- We employ the convolutional neural network as our model backbone for imagery datasets (CIFAR, SVHN, and FashionMNIST). For other high-dimensional inputs, we build the over-complete network with multilayer perceptron with six hidden layers. We use ReLU as the activation function.

To achieve a robust result, each experiment is repeated 10 rounds while keeping all tunable parameters consistent. The average results are presented in Table 2 and Figure 10. At all rounds, the model is initialized in the same way. To enable cross validation, we randomly shuffle the data input sequences for different rounds.

### 6.2 Results and Findings

We present the experimental results in this section aiming to answer research questions (**Q3** – **Q8**).

**Q3.** *How does our OLD$^3$S compare to the state-of-the-arts?*

From the comparative results presented in Table 2 and Figure 10, we make three observations as follows. *First*, our OLD$^3$S achieves the best ACR performance. This result rationalizes our proposal of learning deep learners with complex feature relationships, as the competitors mainly relying on linear models manifest inferior performances.

*Second*, our OLD$^3$S outperforms FOBOS by 73% on average. In addition, FOBOS suffers the largest performance drop in terms of OCA when the old features become unobserved, as shown in Figures 10a, 10b, 10c, 10g, 10h, and 10i. Particularly for Reuter, as shown in Table 5, re-training FOBOS while encountering new data lead to the decrease by an average of 52%. This is because that FOBOS does not correlate the old and new feature spaces thus can be equated to initializing a new learner for the newly emerged features. Our approach excels as we learned the feature correlation to boost the learning performance on the new features, and then enjoys a much smoother learning curve while the feature space evolves.

*Third*, compared to OLSF, our approach wins by 77% on average. The reason can be attributed to that OLSF is tailored for dealing with an incrementally increasing feature space only, and does not possess the mechanism to handle the fading away features. The learned knowledge of the old feature space is hence wasted. Our approach aids the situation by learning a reconstructive mapping between the two feature spaces, letting the learner enjoy the information conveyed by the old and unobservable features, thereby attaining better ACR and sharper OCA curves with the time.

**Q4.** *How helpful is the deep learner enabled by the VI mapping?*

The comparison among four algorithms: FOBOS, FESL, our OLD$^3$S approach and its OLD-Linear variant amounts to the answer. *First*, our OLD$^3$S outperforms FESL and OLD-Linear by ratios of 74% and 44% on average, respectively. This performance gap indicates the non-linear mapping relationship between feature spaces must be respected, as FESL and OLD-Linear both employed linear functions to approximate the reconstructive mapping. *Second*, more significant OCA drops are observed from OLD-Linear in Figures 10b, 10c,and 10e. This result suggests that the low-dimensional latent space resulted from the variational encoding does not suffice to simplify the complex feature reconstruction relationships to an extent that they can be approximated by linear functions.

*Third*, we observe that FESL may even underperformed FOBOS in terms of ACR, despite that FESL suffers a smaller performance drop of OCA overtime. This observation advocates that FESL learned the feature relationship at a certain level, but the linearity of the mapping function does suffice to fully capture the complex feature interactions, such that the linear reconstruction of old features is helpful at the beginning of $\mathcal{T}_2$ (smaller OCA drop) but soon becomes less useful overtime (slower learning rate), and eventually becomes *noises* which negatively affect the prediction accuracy, ending up with inferiority to FOBOS. In other words, it is better to initialize a new learner than trying to reconstruct old and unobservable features inaccurately with an insufficiently capable linear mapping.

**Q5.** *In which cases does an adaptive learning capacity excel?*

A comparison between our OLD$^3$S with the OLD-FD variant answers this question. We observe that 1) OLD$^3$S excels and significantly outperforms OLD-FD in six settings 2) OLD$^3$S converges faster with steeper OCA curves in all settings, especially for Reuter datasets. These two observations validate the *tightness* of Theorem 1 in the sense that, although OLD-FD may end up with higher OCA with increasingly more arriving instances (*e.g.*, Figures 10d and 10g), its slower convergence rate incurs larger online prediction errors before the parameters are readily trained. This makes the usage of HBP to expedite the online learning efficiency become a better choice. In addition, from Figures 10d and 10e, we observe that OLD-FD learns slower as the learning task becomes more difficult. (The objects in CIFAR impose more complex visual concepts than the street-view house numbers in SVHN, where the hindsight

TABLE 3: Runtime of six methods on ten datasets.

| Dataset | FOBOS | FESL | OLSF | OLD-Linear | OLD-FD | OLD$^3$S |
|---|---|---|---|---|---|---|
| magic04 | 24 | 65 | 17 | 230 | 160 | 264 |
| adult | 49 | 136 | 30 | 384 | 266 | 331 |
| EN-FR | 30 | 100 | 102 | 417 | 237 | 336 |
| EN-IT | 30 | 99 | 96 | 412 | 231 | 329 |
| EN-SP | 30 | 98 | 94 | 414 | 235 | 327 |
| FR-IT | 43 | 142 | 170 | 586 | 332 | 466 |
| FR-SP | 42 | 140 | 164 | 585 | 337 | 468 |
| CIFAR | 25 | 1820 | 18 | 2820 | 2932 | 3083 |
| Fashion | 24 | 286 | 16 | 3372 | 4588 | 4653 |
| SVHN | 54 | 2298 | 21 | 4543 | 4310 | 4236 |

TABLE 4: Results of extra errors made by two variants on ten datasets.

| Dataset | magic04 | adult | EN-FR | EN-IT | EN-SP | FR-IT | FR-SP | CIFAR | Fashion | SVHN |
|---|---|---|---|---|---|---|---|---|---|---|
| OLD-Linear | 863 | -1146 | 6982 | 7726 | 8000 | 11460 | 10496 | 1291 | 2504 | 5764 |
| OLD-FD | 882 | 1232 | 4826 | 4270 | 4180 | 6426 | 5916 | 7427 | 1778 | 3783 |

TABLE 5: OCA of FOBOS and OLD$^3$S pre-and-post observing new features on ten datasets.

| Dataset | magic04 | adult | EN-FR | EN-IT | EN-SP | FR-IT | FR-SP | CIFAR | Fashion | SVHN |
|---|---|---|---|---|---|---|---|---|---|---|
| FOBOS (pre) | .790 | .782 | .746 | .760 | .756 | .774 | .770 | .228 | .660 | .141 |
| FOBOS (post) | .644 | .760 | .226 | .280 | .270 | .210 | .216 | .204 | .575 | .132 |
| OLD$^3$S (pre) | .864 | .832 | .838 | .840 | .808 | .824 | .800 | .595 | .870 | .859 |
| OLD$^3$S (post) | .822 | .766 | .796 | .802 | .810 | .828 | .836 | .611 | .824 | .854 |

optimal OCAs in CIFAR and SVHN are 72.7% and 93.3%, respectively). Our OLD$^3$S is invariant to the inherent complexity of the datasets and manifests a fast online learning rate. This finding advocates the adaptive model capacity of our OLD$^3$S is generalizable to more learning tasks, without requiring prior knowledge of the underlying distribution or learning complexity of the doubly-streaming data of interest.

**Q6.** *What is the tradeoff between runtime complexity and algorithm efficacy?*

To answer this question, we conducted the experiment to compare the runtime between our OLD$^3$S and five other

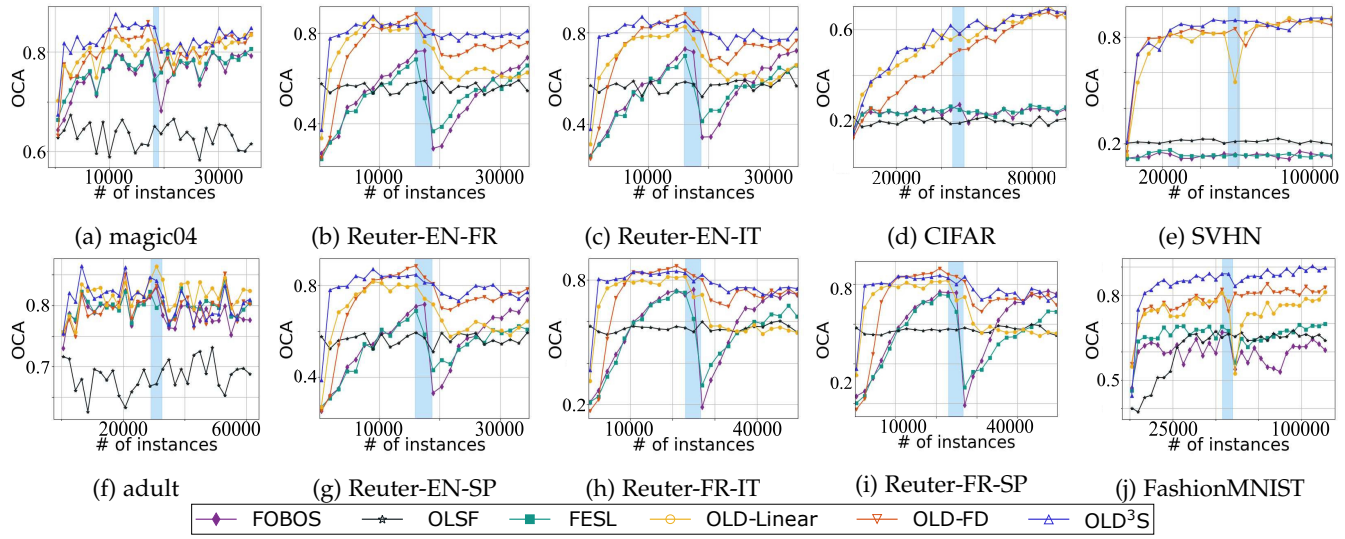
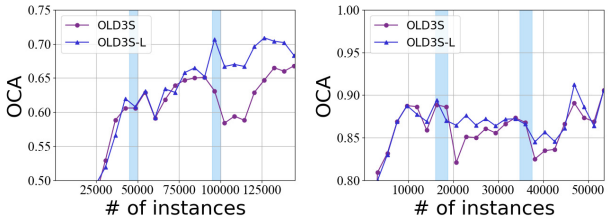



Fig. 10: The trends of OCA of six methods on 10 datasets in the doubly-streaming setting. The blue-shadowed areas indicate the overlapping $\mathcal{T}_b$ timespans.

TABLE 6: Average number of errors made by OLD$^3$S-L and OLD$^3$S on six datasets. The best results are bold. OLD$^3$S-L is the lifelong learning extension of OLD$^3$S.

| Alg. | EN-FR-IT | EN-IT-SP | EN-SP-FR |
|---|---|---|---|
| OLD$^3$S | $15576 \pm 130$ | $15726 \pm 125$ | $15784 \pm 125$ |
| OLD$^3$S-L | $\mathbf{15058 \pm 115}$ | $\mathbf{15224 \pm 130}$ | $\mathbf{15092 \pm 120}$ |
| **Alg.** | **CIFAR** | **Fashion** | **SVHN** |
| OLD$^3$S | $62141 \pm 320$ | $27825 \pm 200$ | $35795 \pm 280$ |
| OLD$^3$S-L | $\mathbf{58753 \pm 240}$ | $\mathbf{26451 \pm 170}$ | $\mathbf{34270 \pm 250}$ |

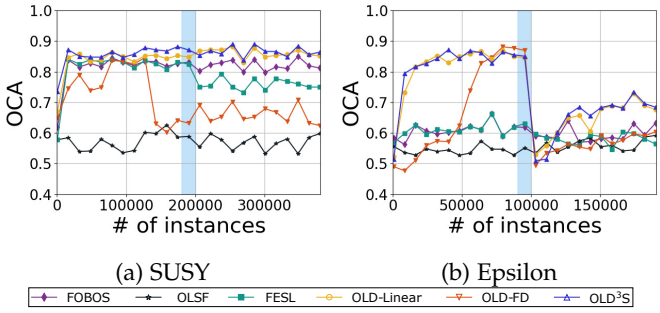



(a) CIFAR (b) Reuter-EN-FR-IT

Fig. 11: The trends of OCA of two methods on two datasets in the doubly-streaming setting. The two blue-shadowed areas indicate two overlapping $\mathcal{T}_b$ timespans.

TABLE 7: Statistics of SUSY and Epsilon. $|\mathcal{S}_1|$ and $|\mathcal{S}_2|$ are the dimensions of the old and new feature spaces.

| No. | Dataset | # Samples | $\lvert\mathcal{S}_1\rvert$ | $\lvert\mathcal{S}_2\rvert$ | # Classes |
|---|---|---|---|---|---|
| 1 | SUSY | 200,000 | 50 | 200 | 2 |
| 2 | Epsilon | 100,000 | 2,000 | 3,000 | 2 |




(a) SUSY (b) Epsilon

Fig. 12: The trends of OCA of six methods on SUSY and Epsilon in the doubly-streaming setting. The blue-shadowed areas indicate the overlapping $\mathcal{T}_b$ timespans.

competitors. We benchmark all experiments on virtual machine, configured as 4 $\times$ Intel(R) Xeon(R) Gold 6148 CPU, one Nvidia V100 GPU, and 16GB RAM. From results presented in Table 3, we can draw two observations. First, the runtime of all linear models FOBOS, OLSF, and FESL are faster than deep models, e.g. OLD$^3$S and its two variants OLD-FD and OLD-Linear on all datasets. However, this is at the cost of their worst performance, which can be found in Table 2 and Figure 10. Second, when comparing three deep models, OLD$^3$S 1) describes a nonlinear relationship yet is faster than OLD-Linear on all *Reuter* datasets, 2) calculates and updates the hedge parameters iteratively yet is faster than OLD-FD on the *SVHN* dataset. This indicates that OLD$^3$S enjoys a comparable runtime performance with its two deep variants while presenting the best accuracy.

To further illustrate the advantage of OLD$^3$S over its two variants, we conducted one more experiment to compare OLD$^3$S with OLD-Linear and OLD-FD to present how many less errors OLD$^3$S made. Results can be found in Table 4, from which we can draw two observations. First, OLD$^3$S always makes less errors than its two variants in addition to OLD-Linear in dataset *adult*. With the help of two mechanisms, OLD$^3$S present better performance than OLD-Linear and OLD-FD in general. Second, while only implementing one mechanism, the cases where either OLD-Linear or OLD-FD prevails may vary across different datasets. For example, thanks to HBP, OLD-Linear makes 6136 fewer errors than OLD-FD in *CIFAR* even if a linear relationship is approximated in a nonlinear image dataset. This observation indicates both mechanisms we proposed are indispensible, and can jointly improve the model performance without sustaining complexity overhead.

**Q7.** *What are the gains by learning lifelong streaming feature spaces instead of repeating the entire learning procedure?*

To answer this question, we conduct experiments implementing OLD$^3$S and OLD$^3$S-L on six datasets. Experimental results are shown in Table 6 where OLD$^3$S-L make less errors on all datasets, from 502 on Reuter-EN-IT-SP to 3388 on CIFAR. We can conclude that OLD$^3$S-L leveraging knowledge learned from all vanished feature spaces has the better performance than OLD$^3$S which only learns mapping from two consecutive spaces.

An interesting observation from Table 6 is that the performance of OLD$^3$S-L on imagery datasets is better than that on Reuter datasets in general. Two examples, the trends of OCA in CIFAR and Reuter-EN-FR-IT, are shown in Figure 11. We can observe the accuracy gap after the second overlapping period on CIFAR is more evident than that on Reuter-EN-FR-IT. The possible reason is that, although knowledge from two previous spaces is distilled by our lifelong method OLD$^3$S-L in both datasets, the evolving relationship between FR and IT is relatively independent from the relationship between EN and FR. Translating FR into IT requires little EN knowledge, and the mapping relationship learned to translate EN to FR is less helpful for tha learned to translate FR to IT. In CIFAR, the second mapping mapping relationship $\mathcal{S}_2 \mapsto \mathcal{S}_3$ is intentionally constructed upon the mapping of $\mathcal{S}_1 \mapsto \mathcal{S}_2$. Specifically, images after the first complex pixel transformation are then processed with a different transformation. Hence, the evolving relationship between $\mathcal{S}_2$ and $\mathcal{S}_3$ inherits from, thus is highly correlated to the previous relationship between $\mathcal{S}_1$ and $\mathcal{S}_2$. Taking advantage of knowledge of the first relationship will help capture the second relationship.

Theses experimental results can validate our hypothesis that the lifelong variant OLD$^3$S-L outperforms the naive repeating method OLD$^3$S if $\mathcal{S}_3$ is evolved from $\mathcal{S}_1$ and $\mathcal{S}_2$ on a space continuum.

**Q8.** *Does the method OLD$^3$S retain high performance on larger scale datasets?*

TABLE 8: Comparative results of averaged cumulative regret (ACR ± mean variance) benchmarked on SUSY and Epsilon, where the lower the value, the better the method performs. The best results are bold.

| Dataset | FOBOS | OLSF | FESL | OLD-Linear | OLD-FD | OLD$^3$S |
|---|---|---|---|---|---|---|
| SUSY | $.075 \pm .024$ | $.319 \pm .024$ | $.105 \pm .026$ | $.043 \pm .021$ | $.193 \pm .036$ | $\mathbf{.031 \pm .019}$ |
| Epsilon | $.279 \pm .025$ | $.329 \pm .021$ | $.284 \pm .024$ | $.149 \pm .028$ | $.257 \pm .027$ | $\mathbf{.144 \pm .027}$ |

To answer this question, we conducted experiments on two dataset *SUSY* and *Epsilon* with larger scale. The statistics are presented in Table 7 experimental results are shown in Table 2 and Figure 10. We can observe that our OLD$^3$S still presents the best performance than all other competitors with the the lowest ACR value with 3.1% and 14.4% respectively, from Table 2. Besides, compared with its two variants, OLD$^3$S always holds the faster convergence rate than OLD-FD. As shown in Figure 12a, OLD-FD shows a sudden drop on the dataset *SUSY* after observing about 150,000 instances in multiple rounds of experiments. This fixed-depth deep model performs even worse than linear models after the overlapping $\mathcal{T}_b$. We extrapolate that this SUSY dataset has a highly nonlinear evolving relationship, making a data-demand OLD-FD suffer from low convergence rate. The slow convergence trend also appears in the dataset Epsilon, as shown in Figure 12b. OLD-FD tends to converge after observing nearly 75,000 instances, while OLD-Linear and OLD$^3$S only needs about 10,000 instances. The significant declines of OLD-Linear and OLD$^3$S on *Epsilon* after $\mathcal{T}_b$ indicate the failure of leveraging old knowledge. To wit, we can observe that OLD$^3$S needs about 10,000 instances to converge while the overlapping period only contains 10,000 instances. As a result, a deep VAE model fails to capture a latent subspace of $\mathcal{S}_1$ and force it to approximate that of $\mathcal{S}_2$ in a short overlapping period.

## 7 Conclusion

This paper proposed a new online learning paradigm, named OLD$^3$S, which enables a deep learner to make on-the-fly decisions on data streams with a constantly evolving feature space. The key idea is to establish a mapping relationship between the old and new features, such that once the old features vanish, they are reconstructed from the new features, allowing the learner to harvest both old and new feature information to make accurate online predictions via ensembling. To realize this idea, the crux lies in the harmonization of model onlineness and expressiveness. To respect the high dimensionality and complex feature interplay in the real-world data streams, our OLD$^3$S approach discovered a shared latent subspace using variational approximation, which can encode arbitrarily expressive mapping functions for feature reconstruction. Meanwhile, as the real-time nature of data streams biases shallow models, (which converge faster hence regret less when learning just begins), our approach enjoyed an optimal depth *learned* from data, starting from shallow and gradually becoming deep if more complex patterns are required to be captured in an online fashion. Theoretical results indicated that our approach can provably benefit from an adaptively learned model depth and an online ensemble prediction. Theoretical and comparative studies evidenced the viability of our approach and its superiority over the state-of-the-art competitors.

## Acknowledgement

This work is supported by the National Science Foundation (NSF) under Grants IIS-2245946 and IIS-2236578, the National Natural Science Foundation of China (NSFC) under grant 62176070, and the Commonwealth Cyber Initiative.

## References

[1] Y. Gong and W. Xu, *Machine learning for multimedia content analysis*. Springer Science & Business Media, 2007, vol. 30.
[2] L. Jiang, "Web-scale multimedia search for internet video content," in *WWW*, 2016, pp. 311–316.
[3] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao, "Learning in high-dimensional multimedia data: the state of the art," *Multimedia Systems*, vol. 23, no. 3, pp. 303–313, 2017.
[4] A.-M. Tousch, S. Herbin, and J.-Y. Audibert, "Semantic hierarchies for image annotation: A survey," *Pattern Recognition*, vol. 45, no. 1, pp. 333–345, 2012.
[5] B. Thomee and M. S. Lew, "Interactive search in image retrieval: a survey," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 2, pp. 71–86, 2012.
[6] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–30, 2018.
[7] E. Ferrara, "The history of digital spam," *Communications of the ACM*, vol. 62, no. 8, pp. 82–91, 2019.
[8] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *SIGIR*, 2017, pp. 335–344.
[9] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and M. Zhou, "A deep latent factor model for high-dimensional and sparse matrices in recommender systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 7, pp. 4285–4296, 2019.
[10] D. Wu, Q. He, X. Luo, M. Shang, Y. He, and G. Wang, "A posterior-neighborhood-regularized latent factor model for highly accurate web service qos prediction," *IEEE Transactions on Services Computing*, 2019.
[11] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and X. Wu, "A data-characteristic-aware latent factor model for web services qos prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
[12] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.
[13] G. Marcus, "The next decade in AI: four steps towards robust artificial intelligence," *arXiv preprint arXiv:2002.06177*, 2020.
[14] O. Bousquet and A. Elisseeff, "Stability and generalization," *The Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.
[15] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, *Generalization in deep learning*. Cambridge university press, 2021.
[16] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Learning with feature evolvable streams," in *NeurIPS*, vol. 30, 2017.
[17] C. Hou and Z.-H. Zhou, "One-pass learning with incremental and decremental features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2776–2792, 2017.
[18] E. Beyazit, J. Alagurajah, and X. Wu, "Online learning from data streams with varying feature spaces," in *AAAI*, vol. 33, no. 01, 2019, pp. 3232–3239.

[19] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Online learning from capricious data streams: a generative approach," in *IJCAI*, 2019, pp. 2491–2497.

[20] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, "Learning with feature and distribution evolvable streams," in *ICML*. PMLR, 2020, pp. 11 317–11 327.

[21] B.-J. Hou, Y.-H. Yan, P. Zhao, and Z.-H. Zhou, "Storage fit learning with feature evolvable streams," in *AAAI*, 2021.

[22] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Prediction with unpredictable feature evolution," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2021.

[23] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Toward mining capricious data streams: A generative approach," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 3, pp. 1228–1240, 2021.

[24] Y. He, X. Yuan, S. Chen, and X. Wu, "Online learning in variable feature spaces under incomplete supervision," in *AAAI*, vol. 35, no. 5, 2021, pp. 4106–4114.

[25] Y. He, S. Chen, B. Wu, X. Yuan, and X. Wu, "Unsupervised lifelong learning with curricula," in *WWW*, 2021, pp. 3534–3545.

[26] Y. He, J. Dong, B.-J. Hou, Y. Wang, and F. Wang, "Online learning in variable feature spaces with mixed data," in *ICDM*. IEEE, 2021, pp. 181–190.

[27] C. Schreckenberger, Y. He, S. Lüdtke, C. Bartelt, and H. Stuckenschmidt, "Online random feature forests for learning in varying feature spaces," in *AAAI*, vol. 37, no. 4, 2023, pp. 4587–4595.

[28] D. Wu, S. Zhuo, Y. Wang, Z. Chen, and Y. He, "Online semi-supervised learning with mix-typed streaming features," in *AAAI*, vol. 37, no. 4, 2023, pp. 4720–4728.

[29] Y. He, C. Schreckenberger, H. Stuckenschmidt, and X. Wu, "Towards utilitarian online learning–a review of online algorithms in open feature space," in *IJCAI*, 2023.

[30] Q. Zhang, J. Wu, P. Zhang, G. Long, I. W. Tsang, and C. Zhang, "Inferring latent network from cascade data for dynamic social recommendation," in *ICDM*. IEEE, 2016, pp. 669–678.

[31] Y. Xiao, B. Liu, J. Yin, and Z. Hao, "A multiple-instance stream learning framework for adaptive document categorization," *Knowledge-Based Systems*, vol. 120, pp. 198–210, 2017.

[32] A. Cano and B. Krawczyk, "Kappa updated ensemble for drifting data stream mining," *Machine Learning*, vol. 109, no. 1, pp. 175–218, 2020.

[33] Y.-F. Li, Y. Gao, G. Ayoade, H. Tao, L. Khan, and B. Thuraisingham, "Multistream classification for cyber threat data with heterogeneous feature space," in *WWW*, 2019, pp. 2992–2998.

[34] X. Nie, R. Gao, R. Wang, and D. Xiang, "Online multiview deep forest for remote sensing image classification via data fusion," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 8, pp. 1456–1460, 2020.

[35] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[36] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[37] L. Zhang, "Online learning in changing environments." in *IJCAI*, 2020, pp. 5178–5182.

[38] Y. Huang, C. Zhang, J. Yella, S. Petrov, X. Qian, Y. Tang, X. Zhu, and S. Bom, "Grassnet: Graph soft sensing neural networks," in *IEEE International Conference on Big Data*. IEEE, 2021, pp. 746–756.

[39] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *ICML*, 2003, pp. 928–936.

[40] C. Schreckenberger, T. Glockner, H. Stuckenschmidt, and C. Bartelt, "Restructuring of hoeffding trees for trapezoidal data streams," in *ICDM*. IEEE, 2020, pp. 416–423.

[41] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[43] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online deep learning: Learning deep neural networks on the fly," in *IJCAI*, 7 2018, pp. 2660–2666.

[44] Z.-H. Zhou and J. Feng, "Deep forest," *National Science Review*, vol. 6, no. 1, pp. 74–86, 2019.

[45] J. Ren, B. Hou, and Y. Jiang, "Deep forest for multiple instance learning," *Journal of Computer Research and Development*, vol. 56, no. 8, p. 1670, 2019.

[46] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[47] F. van Beers, "Hedge backpropagation in convolutional neural networks," 2021.

[48] Z.-H. Zhou, "Ensemble learning," in *Machine learning*. Springer, 2021, pp. 181–210.

[49] C. C. Aggarwal, *Data streams: models and algorithms*. Springer, 2007, vol. 31.

[50] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[51] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu, "Towards mining trapezoidal data streams," in *ICDM*. IEEE, 2015, pp. 1111–1116.

[52] ——, "Online learning from trapezoidal data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2709–2723, 2016.

[53] C. Hou, L.-L. Zeng, and D. Hu, "Safe classification with augmented features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2176–2192, 2018.

[54] E. Beyazit, M. Hosseini, A. Maida, and X. Wu, "Learning simplified decision boundaries from trapezoidal data streams," in *ICANN*. Springer, 2018, pp. 508–517.

[55] D. Wu, Y. He, X. Luo, and M. Zhou, "A latent factor analysis-based approach to online sparse streaming feature selection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[56] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.

[57] Y. Lu and J. Lu, "A universal approximation theorem of deep neural networks for expressing probability distributions," *NeurIPS*, vol. 33, pp. 3094–3105, 2020.

[58] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[59] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[60] T.-H. Lin and H. Kung, "Stable and efficient representation learning with nonnegativity constraints," in *ICML*. PMLR, 2014, pp. 1323–1331.

[61] H. Mhaskar, Q. Liao, and T. Poggio, "When and why are deep networks better than shallow ones?" in *AAAI*, vol. 31, no. 1, 2017.

[62] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *ECCV*. Springer, 2016, pp. 646–661.

[63] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," in *ICLR*, 2017.

[64] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, no. 19, pp. 551–585, 2006. [Online]. Available: http://jmlr.org/papers/v7/crammer06a.html

[65] J. Lu, P. Zhao, and S. Hoi, "Online passive aggressive active learning and its applications," in *Asian Conference on Machine Learning*. PMLR, 2015, pp. 266–282.

[66] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[67] N. Srebro, K. Sridharan, and A. Tewari, "On the universality of online mirror descent," *NeurIPS*, vol. 24, 2011.

[68] B. McMahan, "Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization," in *AISTATS*. JMLR Workshop and Conference Proceedings, 2011, pp. 525–533.

[69] F. Orabona, K. Crammer, and N. Cesa-Bianchi, "A generalized online mirror descent with applications to classification and regression," *Machine Learning*, vol. 99, no. 3, pp. 411–435, 2015.

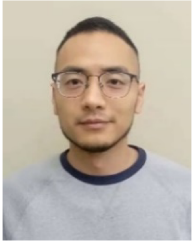
[70] H. Fang, N. Harvey, V. Portella, and M. Friedlander, "Online mirror descent and dual averaging: keeping pace in the dynamic case," in *ICML*. PMLR, 2020, pp. 3008–3017.

[71] A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire, "Algorithms for portfolio management based on the newton method," in *ICML*, 2006, pp. 9–16.

[72] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[73] N. N. Schraudolph, J. Yu, and S. Günter, "A stochastic quasi-newton method for online convex optimization," in *AISTATS*. PMLR, 2007, pp. 436–443.
[74] Y. Ye, L. Lei, and C. Ju, "Hones: a fast and tuning-free homotopy method for online newton step," in *AISTATS*. PMLR, 2018, pp. 2008–2017.
[75] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.
[76] P. Zhao, S. C. Hoi, J. Wang, and B. Li, "Online transfer learning," *Artificial intelligence*, vol. 216, pp. 76–102, 2014.
[77] Q. Wu, H. Wu, X. Zhou, M. Tan, Y. Xu, Y. Yan, and T. Hao, "Online transfer learning with multiple homogeneous or heterogeneous sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1494–1507, 2017.
[78] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *CVPR*. IEEE, 2011, pp. 577–584.
[79] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, "Online learning of object representations by appearance space feature alignment," in *ICRA*. IEEE, 2020, pp. 10 473–10 479.
[80] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
[81] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *ICLR*, 2014.
[82] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *ICML*. PMLR, 2015, pp. 1613–1622.
[83] L. Galke, B. Franke, T. Zielke, and A. Scherp, "Lifelong learning of graph neural networks for open-world node classification," in *IJCNN*. IEEE, 2021, pp. 1–8.
[84] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[85] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
[86] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.
[87] V. V. Ramasesh, A. Lewkowycz, and E. Dyer, "Effect of scale on catastrophic forgetting in neural networks," in *ICLR*, 2021.
[88] C. Shen, X. Wang, J. Song, L. Sun, and M. Song, "Amalgamating knowledge towards comprehensive classification," in *AAAI*, 2019.
[89] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online meta-learning," in *ICML*, 2019, pp. 1920–1930.
[90] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
[91] J. C. Duchi and Y. Singer, "Efficient learning using forward-backward splitting." in *NeurIPS*, vol. 22, 2009, pp. 495–503.

**Heng Lian** (Student Member, IEEE) received an M.S. in Computer Science from the University of Nottingham and a B.E. from DaLian Maritime University (China) in 2019 and 2017, respectively. Heng Lian is a Ph.D. student of Computer Science at Old Dominion University. His research interest lies in machine learning and specifically in online machine learning and data stream analytics. He is a recipient of Best Graduate Researcher Award from ODU at 2023.



**Di Wu** (Member, IEEE) received the Ph.D. degree from the Chongqing Institute of Green and Intelligent Technology (CIGIT), Chinese Academy of Sciences (CAS), China in 2019. Currently, he is a Professor for the College of Computer and Information Science, Southwest University, Chongqing, China. He has over 75 publications, including 19 IEEE/ACM TRANSACTIONS papers and several conference papers of AAAI, IEEE ICDM, WWW, IJCAI, etc. He is serving as an Associate Editor for the journals of Neurocomputing and Frontiers in Neurorobotics. His research interests include data mining and machine learning. His homepage: wudi1989.github.io/Homepage/.

**Bo-Jian Hou** received the BS and PhD degrees in computer science from Nanjing University, China, in 2014 and 2020, respectively. He is currently a postdoctoral researcher of Division of Informatics in the Department of Biostatistics, Epidemiology and Informatics at the Perelman School of Medicine in the University of Pennsylvania. His research interests include trustworthy AI such as interpretability, fairness and robustness and data mining such as biomedical data analysis and survival analysis.

**Jian Wu** (Member, IEEE) received his Ph.D. degree from the Pennsylvania State University in 2011. He received a BS degree from the University of Science and Technology of China in 2004. Dr. Wu is an assistant professor of Computer Science at Old Dominion University. His research interest includes natural language processing, natural language understanding, scholarly big data, information retrieval, and the science of science. He won the best reviewer at the ACM/IEEE-CS Joint Conference on Digital Libraries in 2018. He shared the British Computer Society Award 2021 for the Best Open Source Project with Dr. C. Lee Giles.

**Yi He** (Member, IEEE) received his Ph.D. degree in computer science from the University of Louisiana at Lafayette and a B.E. from the Harbin Institute of Technology (China) in 2020 and 2013, respectively. He is an assistant professor of Computer Science at Old Dominion University. He published more than 45 technical articles in journals and conference proceedings, including 13 IEEE/ACM TRANSACTIONS papers and premier venues including AAAI, IJCAI, WWW, ICDM, SDM, etc. He served as Workshop Co-Chair of IEEE International Conference on Data Mining (ICDM) 2023 and Registration Chair of IEEE ICDM 2022. He is a recipient of the IEEE TCII Volunteer Award 2022 and the NSF CRII Award 2023.