Online Deep Learning from Doubly-Streaming Data

Heng Lian
hliao001@odu.edu
Old Dominion University
Norfolk, Virginia, USA

John Scovil Atwood
jatwo002@odu.edu
Old Dominion University
Norfolk, Virginia, USA

Bo-Jian Hou
houbo@pennmedicine.upenn.edu
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Jian Wu
j1wu@odu.edu
Old Dominion University
Norfolk, Virginia, USA

Yi He∗
yihe@cs.odu.edu
Old Dominion University
Norfolk, Virginia, USA

1 INTRODUCTION

Machine learning has become a fundamental building block in many cyber infrastructures, providing an automated scalable apparatus to analyze the high-dimensional data streams (e.g., images, texts, videos) pervading all corners of the Internet [22, 23, 39]. Examples include multimedia retrieval [63, 64], online speech analytics [16, 18], recommender systems [11, 13, 66, 68, 69], 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 [50]. 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 [8, 39]. 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 [4, 25–32, 34, 77] and domain applications [9, 45, 53, 71, 74]. 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 performance over time [21, 47, 73].

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 [36], 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 [31]. Tangibly, as the new and old sensors (i.e., features) often differ in terms of amount, version, metric, and position, a new classifier needs to be initialized. Yet, this new classifier...
may stay weak and error-prone before the training samples carry-
ing these new features grows to a sufficiently large volume. Mean-
while, the old classifier becomes unusable with the unobserved fea-
tures, leading to substantial waste of the data collection and train-
ing effort. A relationship between the pre-and-post evolving fea-
ture spaces must be established, so that the old features can be re-
constructed 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 [26, 28–30].

Unfortunately, all existing studies suffer from a tradeoff between
onlineness and expressiveness. Specifically, on the one hand, shallow
learners (e.g., generalized linear models [81], Hoeffding trees [59])
possess a faster online convergence rate, thanks to their simple
model structures with a small number of trainable parameters [61].
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 [43, 57],
deep forests [56, 80]) enjoy a low-dimensional hidden representa-
tion 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 pre-
dictions tend to be made before the learners converge to an equi-
librium. 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 ques-
tion: How can we build an online learner that joins the two merits,
namely, 1) converges as fast as shallow models to minimize the on-
line regrets and 2) learns latent representations as expressive as deep
models from high-dimensional inputs with complex feature relation-
ships.

Our affirmative answer provides a novel learning paradigm, termed
**Online Learning Deep models from Data of Double Streams (OLD3S).**
Our key idea is to train an online learner that automatically adjusts
its learning capacity in accordance with the complexities and tem-
poral variation patterns of input data stream. Specifically, OLD3S
is with an over-complete neural architecture [44, 57, 65] and starts
from using its shallow layers, approximating a simple classifier to
attain fast convergence at initial rounds. Over time, the deeper lay-
ers are gradually mobilized, as more samples streaming in requires
1) a highly capable classifier that can learn expressive latent rep-
resentations and 2) a precise delineation of complex feature inter-
play. Knowledge reuse is enabled in both 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 [79]. This benefits our approach by expediting the con-
vergence 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 summarized 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 evolu-
tion tend to incur a tradeoff between convergence rate and
learning capacity. The technical challenges are manifested
from empirical evidence in Section 3.

ii) A novel OLD3S 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. Details are in Section 4.

iii) 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 5.

2 RELATED WORK

**Online Learning with Doubly-Streaming Data.** Online learn-
ing algorithms were devised for data stream processing [2, 60],
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 al-
lowing data to grow in terms of volume, in an orthogonal setting,
hoping the features describing input data to stay strictly unchang-
ing is unrealistic over long time spans. As a response, the pioneer-
ing studies [5, 33, 67, 75, 76] explored a setting of incremental fea-
ture learning, allow the arriving data instances to carry different
sets of features yet later instances are assumed to include monotono-
ically more features than the earlier ones. Subsequent works that
strive to learn evolving feature spaces [4, 27–32, 34, 77] further re-
axed 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 meth-
ods is to establish a mapping relationship between the old and new
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 fea-
tures, 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 evolv-
ing 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 guaran-
tees. 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 OLD3S 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 [12, 48, 49]. However, their universal ap-
proximation capability is grounded on an assumption of an infini-
tely wide hidden layer, which cannot be satisfied in practical mod-
eling. The advent of Deep Neural Networks sidestepped this issue
by imposing a hierarchical representation learning procedure [3,
46, 52], trading in width for depth, so as to fit complex decision
functions underlying data. However, this hierarchical design intro-
duces over-parameterization, where the large number of learnable
parameters request massive rounds of training iterations over huge
datasets to converge. Online decision-making using deep learning
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. [35] 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. Larson et al. [42] later identified a strategy to construct deep networks structured as fractals. This confers the ability to regularize work depth (representing the entire model capacity) in accordance with the underlying function in an adaptive, automated, and data-adaptively.

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

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 \( \{(x_t, y_t) \mid t = 1, 2, \ldots, T\} \) denote an input sequence, where \( x_t \) 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 [27, 29, 31, 76] that focus on binary classification only.

In the context of doubly-streaming data, we follow the pioneer [31], consider the set of features describing \( x_t \) to evolve with the following regularity, illustrated in Figure 1. Specifically,

- In the span \( t_1 \in \mathcal{T}_1 := \{1, \ldots, t_1\} \), the classifier observes the instances described by the feature space \( S_1 \), i.e., \( x_{t_1} \in S_1 \subseteq \mathbb{R}^{d_1} \), each of which is a \( d_1 \)-dimensional vector.
- In the span \( t_2 \in \mathcal{T}_2 := \{t_1 + 1, \ldots, t_2\} \), the feature space evolves, and the classifier observes the two feature spaces \( S_1 \) and \( S_2 \) simultaneously, with each data instance being \( x_{t_2} = [x_{t_2}^{S_1}, x_{t_2}^{S_2}]^\top \in S_1 \times S_2 \subseteq \mathbb{R}^{d_1 + d_2} \).
- In the span \( t_3 \in \mathcal{T}_3 := \{t_2 + 1, \ldots, t_3\} \), the old space \( S_1 \) is closed out, and the classifier observes the evolved \( S_2 \) only. Each data instance is \( x_{t_3} \in S_2 \subseteq \mathbb{R}^{d_2} \), a \( d_2 \)-dimensional vector.

Note, such feature space evolving from \( S_1 \) to \( S_2 \) to \( S_3 \) can be easily generalized to infinitely more spaces (e.g., \( S_2 \) to \( S_3 \), then \( S_3 \) to arbitrary \( S_4 \)) throughout all timespan \( |\mathcal{T}_1| \ll |\mathcal{T}_2| \) or \( |\mathcal{T}_3| \).

\( S_2 \), 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 \( t = \{t_1, t_2, t_3\} \), the learner \( f_t \) observes \( x_t \) and makes a prediction \( \hat{y}_t = f_t(x_t) \). 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 [11, 51, 54, 62] or second-order [1, 24, 58, 72] oracles, getting prepared for the next round. Our goal is to find a sequence of classifiers \( \{f_1, \ldots, f_T\} \) that minimizes the empirical risk over \( T \) rounds: \( \min_{f_1, \ldots, f_T} \sum_{t=1}^T \ell(y_t, f_t(x_t)) \), 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 \( \mathcal{T}_2 \) to learn a reconstructive mapping \( \phi : S_2 \rightarrow S_1 \), such that once the features of \( S_1 \) 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 [26, 27, 29, 31].

Let \( f_1 = (f_1^{S_1}, f_1^{S_2}) \) denote the learner with \( f_1^{S_1} \) and \( f_1^{S_2} \) being the two classifiers corresponding to the \( S_1 \) and \( S_2 \) feature spaces, respectively. During \( \mathcal{T}_2 \), instead of predicting the observed instance as \( f_2^{S_2}(x_t) \), the learner exploits the unobserved information from \( S_1 \) to make prediction as: \( f_2(x_t) = \lambda_1 \cdot f_1^{S_1}(x_t) + \lambda_2 \cdot f_1^{S_2}(x_t) \), with \( \lambda_1 = \phi(x_t) \in S_1 \) being the reconstructed data vector in the \( S_1 \) space. With delicately tailored ensemble parameters \( \lambda_1 \) and \( \lambda_2 \), this reconstruction-based learning method enjoys a provably better prediction performance than using the classifier \( f_2^{S_2} \) only.

Unfortunately, this method does not scale up to cope with real-world media data streams because of two challenges as follows.

**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, 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, as each instance requiring immediate prediction is presented only once, the deep learners tend to regret [10], making substantial errors before converging to equilibria. To verify this, a simple example reduced from the CIFAR
The deeper the learning model, the slower the convergence rate. As they all require a sufficiently long overlapping phase expiring their lifespans—a too long timespan to learn a latent feature subspace that connects the old and new feature spaces. As such, we immediately reconstruct the $S_1$ data representations from the shared surrogate statistics. To make this process online, we propose a neural architectural design which learns the optimal model 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 $S_1$ stream, and hence the higher the online classification accuracy can be obtained by ensembling the old and new classifiers. The details are presented 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:

$$
\min_{\phi} \sum_{t} \left( \mathcal{L}_{\text{VI}}(\phi) + \mathcal{L}_{\text{REC}}(\phi) \right) + \sum_{t_1, t_2} \mathcal{L}_{\text{CLF}}(f_t, \phi).
$$

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 $\mathcal{L}_{\text{CLF}}$ synergizes 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) [7, 20, 40] 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:

$$
\mathcal{L}_{\text{VI}}^{(S_1, S_2)} = -\mathbb{E}_{Q(z_t | x_t)} \left[ \log P(x_t | z_t) + \text{KL} \left( \mu_{S_1} \left( \sigma_{S_1}^{-2} \right) \right) \right], \quad (2)
$$

where $t \in T_1 \cup T_2$, and $t \in T_b$ for the VAEs on $S_1$ and $S_2$, 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 $x_t \approx \text{Dec}(z_t)$ where $z_t = \text{Enc}(x_t)$. ii) The second term gauges the Kullback-Leibler
Figure 3: An architectural illustration of our OLD^3S computational network during the overlapping $T_b$ timespan.

(4.1) Online Prediction with Ensembled Learners

Once the old features of $S_1$ vanish, the learner $f_t$ is not likely to make accurate predictions on the arriving instances by relying on $S_2$ solely. Let $f_t = (f_S^{S_1}, f_S^{S_2})$ denote the learner at the beginning of $T_b$ when $S_2$ just emerges. As $T_b$ is short, the $f_S^{S_2}$ part of the learner corresponding to the new features of $S_2$ have been trained with very few instances hence is not likely to converge. Relying on $f_S^{S_2}$ to predict the instances in $T_2$ would incur substantial regrets.

To aid, we leverage the old $f_S^{S_1}$ part that has been trained with a much larger number of instances during $T_1$. 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 $f_S^{S_2}$ is not ready, defined as follows.

$$L_{CLP} := \ell(y_t, \hat{y}_t) = -\sum_{c=1}^{C} y_t,c \log(\hat{y}_t,c), \quad \forall t \in T_b \cup T_2,$$  

where Eq. (4) employs cross-entropy [22] to gauge the multi-class learning loss, with $y_t,c$ and $\hat{y}_t,c$ being the true and predicted probability that $x_t$ 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 $x_t$ and its reconstructed version $\tilde{x}_t^{S_1}$ in making predictions. At the beginning of $T_2$ when the feature space just evolved, the old classifier $f_S^{S_1}$ should be largely helpful with large $p$. Over time, the value of $p$ decays because of two reasons 1) the new classifier $f_S^{S_2}$ becomes stronger and 2) the old classifier $f_S^{S_1}$ can be less useful due to the distribution drift. An updating strategy needs to be designed 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 [10], where the empirical risks of using the old and new classifiers to make independent predictions are accumulated as:

$$R_T^{S_1} = \sum_{t=T_1+1}^{T_2} \ell(y_t, f_S^{S_1}(\tilde{x}_t^{S_1})), \quad R_T^{S_2} = \sum_{t=T_1+1}^{T_2} \ell(y_t, f_S^{S_2}(x_t)).$$

The smaller the cumulative empirical risk is suffered, the better the classifier is, and hence the higher its corresponding coefficient is uplifted exponentially. The updating rule is defined as $p = e^{-\eta R_T^{S_1}} / (e^{-\eta R_T^{S_1}} + e^{-\eta R_T^{S_2}}),$ where $\eta$ is a tuned parameter.

4.3 Adaptive Model Depth Learning with HBP

With the reconstructive mapping and the ensemble prediction, the information conveyed by the unobserved $S_1$ can be reap 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) \cite{25,57} 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 \( z_{t}^{(l)} = \text{Enc}(z_{t}^{(l-1)}) \), with \( z_{t}^{(0)} = x_{t} \), where \( t \in T_{1} \cup T_{2} \) and \( t \in T_{1} \cup T_{2} \) for the VAEs correspond to \( S_{1} \) and \( S_{2} \), respectively. The objective of HBP is defined as follows.

\[
\min_{\{\alpha^{(l)}\}_{l=1}^{L}} \left\{ \sum_{l_{1}, l_{2}} \alpha_{l_{1}, l_{2}} \left( L_{V1}^{(l_{1})} + L_{REC}^{(l_{1})} + \sum_{l_{0}} L_{CLF}^{(l_{0})} \right) \right\},
\]

where the loss terms \( L_{V1}^{(l)} \), \( L_{REC}^{(l)} \), and \( L_{CLF}^{(l)} \) are evaluated on \( z_{t}^{(l)} \) at the \( l \)-th layer as shown in Figure 3. In particular, 1) Evaluated by \( L_{V1}^{(l)} \) is how well the latent code \( z_{t}^{(l)} \) can summarize the raw inputs with a surrogate Gaussian via using Eq. (2); for instances of \( S_{1} \), it is evaluated over \( T_{1} \) and \( T_{2} \) timespans, and for instances of \( S_{2} \), it is evaluated over \( T_{2} \) only. 2) Evaluated by \( L_{REC}^{(l)} \) is how precisely the reconstructive mapping is learned so that the \( S_{1} \) feature space can be reconstructed from the data instances of \( S_{2} \) via using Eq. (3); it is only evaluated during the overlapping phase \( T_{1} \) where \( S_{1} \) and \( S_{2} \) coexist. 3) Evaluated by \( L_{CLF}^{(l)} \) is how accurately the ensemble of both old and new classifiers can make online predictions via using Eq. (4); it is evaluated during \( T_{2} \) and \( T_{1} \) as the ensemble prediction is used only if the features of \( S_{2} \) 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 \cite{19}: \( \alpha_{l_{1}, l_{2}}^{(l)} \leftarrow \text{Norm}(\alpha_{l_{1}, l_{2}}^{(l)} \beta L_{V1-REC-CLF}^{(l)} \), where \( \beta \in (0, 1) \) is a discounting rate and \( L_{V1-REC-CLF}^{(l)} \) accumulates the three losses in Eq. (7) suffered at the \( l \)-th round. Denoted by \( \text{Norm}() \) 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.

### 5 EXPERIMENTS

Empirical results are presented to verify the viability and effectiveness of our OLD3S approach. We elaborate the experimental setups in Section 5.1 and extrapolate the results and findings in Section 5.2.

| Table 1: Statistics of the 10 datasets. \( |S_{1}| \) and \( |S_{2}| \) are the dimensions of the old and new feature spaces, respectively. |
| No. | Dataset  | # Samples | \( |S_{1}| \) | \( |S_{2}| \) | # 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      |

### 5.1 Evaluation Setup

#### 5.1.1 Dataset Preparation

We benchmark our OLD3S 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 one feature space \( S_{1} \) at first, and we artificially create a new feature space \( S_{2} = \text{sigmoid}(W^{T}S_{1}) \) with a random Gaussian \( W \) and a nonlinear sigmoid function. The two feature spaces are concatenated as the shape 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 documents after a short overlapping \( T_{1} \) 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 \( S_{2} \) feature 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 \cite{17,25} 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.

#### 5.1.2 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** \cite{14} 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** \cite{75} 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-weighted to new features only if these new features...
convey significant information that changes the decision boundary.

- **FESL** [31] 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.

### 5.1.3 Ablation Variants
For the ablation study, two variants of our OLD3S approach are proposed, named OLD-Linear and OLD-FD. They differ from our original OLD3S design by: 1) **OLD-Linear** employed 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.

### 5.1.4 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) to measure the performance. Specifically, they are defined:

\[
\text{OCA}(f_t) = 1 - \frac{1}{B} \sum_{i=1}^{B} \delta[y_i \neq f_t(x_i)], \quad T = \lvert T_1 \cup T_2 \cup T_3 \rvert 
\]
\[
\text{ACR} = \frac{1}{T} \sum_{i=1}^{T} \max_{f^*} \text{OCA}(f^*) - \text{OCA}(f_t) \right]. 
\]

Intuitively, OCA dynamically measures the accuracy of a classifier \(f_t\) 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, and the better the online classification was performed.

### 5.2 Results and Findings
We present the experimental results in Table 2 and Figure 4, aiming to answer three research questions (Q1 – Q3) as follows.

**Q1. How does our OLD3S approach compare to the state-of-the-arts?**

From the comparative results presented in Table 2, we make three observations as follows. **First**, our OLD3S 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 OLD3S outperforms FOBOS by 69% on average. In addition, FOBOS suffers the largest performance drop in terms of OCA when the old features become unobserved, as shown in Figures 4a, 4b, and 4c. 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 as soon as 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, 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 along the time horizon.

**Q2. How helpful is the deep learner enabled by the VI mapping?**

The comparison among FOBOS, FESL, our OLD3S approach and its OLD-Linear variant amounts to the answer. **First**, our OLD3S outperforms FESL and OLD-Linear by ratios of 69% and 44% on average, respectively. This performance gap indicates that 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 4b, 4c, and 4e. This result suggests that the low-dimensional latent space resulting 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 underperform 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 \(T_2\) (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 features inaccurately with an insufficiently capable linear mapping.

**Q3. In which cases does an adaptive learning capacity excel?**

A comparison between our OLD3S with the OLD-FD variant answers this question. We observe that 1) OLD3S excels and significantly outperforms OLD-FD in six settings 2) OLD3S converges faster with steeper OCA curves in all settings. These two observations validate the tightness of HBP in the sense that, although OLD-FD may end up with higher OCA with increasingly more arriving data instances (e.g., Figures 4a and 4d), its slower convergence rate incurs larger online prediction errors before the network parameters are readily trained. This necessitates the usage of HBP to expedite the online learning efficiency.

In addition, from Figures 4d and 4c, 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 numbers in SVHN, where the hindsight optimal OCAs in CIFAR and SVHN are 72.7% and 93.3%, respectively). Our OLD3S 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 OLD3S is generalizable to more learning tasks, without requiring prior knowledge of the underlying distribution or learning complexity of the doubly-streaming data of interest.

### 6 Conclusion
This paper proposed a new online learning paradigm, named OLD3S, which enables a deep learner to make on-the-fly decisions on data streams with a constantly evolving feature space. The key idea
Table 2: Comparative results of averaged cumulative regret (ACR ± mean variance) benchmarked on 10 datasets, where the lower the value, the better the method performs. The best results are bold. The bullet • indicates that our OLD$^S$ 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$^S$ |
|------------|-------------|------------|------------|------------|--------|---------|
| magic04    | .119 ± .022*| .335 ± .021*| .110 ± .016*| .075 ± .018| .076 ± .021| .052 ± .017|
| adult      | .076 ± .064| .225 ± .019*| .067 ± .044| .055 ± .017| .068 ± .018| .049 ± .019|
| EN-FR      | .326 ± .064*| .324 ± .018*| .345 ± .044*| .168 ± .030*| .137 ± .030*| .068 ± .025|
| EN-IT      | .318 ± .060*| .314 ± .019*| .337 ± .040*| .197 ± .028*| .143 ± .033*| .083 ± .024|
| EN-SP      | .302 ± .060*| .322 ± .021*| .335 ± .037*| .197 ± .036*| .136 ± .027*| .077 ± .024|
| FR-IT      | .278 ± .047*| .301 ± .013*| .314 ± .037*| .195 ± .031*| .147 ± .030*| .084 ± .026|
| FR-SP      | .272 ± .046*| .310 ± .014*| .336 ± .040*| .201 ± .020*| .155 ± .026| .102 ± .027|
| CIFAR      | .468 ± .017*| .504 ± .013*| .463 ± .013*| .166 ± .032| .232 ± .038*| .150 ± .030|
| Fashion    | .305 ± .053*| .294 ± .016*| .247 ± .023*| .160 ± .033*| .123 ± .019*| .056 ± .015|
| SVHN       | .808 ± .011| .604 ± .011*| .806 ± .011*| .144 ± .038*| .120 ± .025| .089 ± .018|
| w/t/l      | 9/1/0       | 10/0/0     | 9/1/0       | 7/3/0       | 6/4/0   | —       |

Figure 4: The trends of OCA of six methods on five datasets in the doubly-streaming setting. The blue-shadowed areas indicate the overlapping $T_b$ timespans. Due to the space limitation, complete results are deferred to the supplementary file.

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$^S$ 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, 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. Comparative studies evidenced the viability of our approach and its superiority over the state-of-the-art competitors.

REFERENCES

[1] Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E Schapire. 2006. Algorithms for portfolio management based on the newton method. In ICML. 9–16.
[2] Charu C Aggarwal. 2007. Data streams: models and algorithms. Vol. 31. Springer.
[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35, 8 (2013), 1798–1828.
[4] Ege Beyazit, Jeerthathi Agarajah, and Xindong Wu. 2019. Online learning from data streams with varying feature spaces. In AAM, Vol. 33. 3232–3239.
[5] Ege Beyazit, Matin Hosseini, Anthony Maida, and Xindong Wu. 2018. Learning simplified decision boundaries from trapezoidal data streams. In ICANN. Springer, 508–517.
[6] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. Journal of the American statistical Association 112, 518 (2017), 859–877.
[7] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2013. Weight uncertainty in neural network. In Springer, 508–517.
[8] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. The Journal of Machine Learning Research 2 (2002), 499–526.
[9] Tianping Chen and Hong Chen. 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks 6, 4 (1995), 911–917.
[10] Jingyan Chen, Hanwang Zhang, Xiangnan He, Ligang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In SIGIR. 335–344.
[11] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 2020. Meta-Learning. In ICML. PMLR, 3008–3017.
[12] Yingying Chen and Hong Chen. 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks 6, 4 (1995), 911–917.
[13] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2020. Recommendation to dynamical systems. IEEE Transactions on Neural Networks and Learning Systems 31, 12 (2020), 616–628.
[14] John C Duchi and Yoram Singer. 2009. Efficient Learning using Forward-Backward Splitting. In NeurIPS, Vol. 22. 495–503.
[15] Emiliano Ferrara. 2019. The history of digital spam. Commun. ACM 62, 8 (2019), 82–91.
[16] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 2019. Online Meta-Learning. In ICML. 1929–1939.
[79] Zhi-Hua Zhou. 2021. Ensemble learning. In *Machine learning*. Springer, 181–210.

[80] Zhi-Hua Zhou and Ji Feng. 2019. Deep forest. *National Science Review* 6, 1 (2019), 74–86.

[81] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*. 928–936.