Journal of Machine Learning Research 26 (2025) 1-59 Submitted 3/24; Revised 1/25; Published 1/25

Supervised Learning with Evolving Tasks and Performance Guarantees

Verónica Álvarez VALVAREZ@BCAMATH.ORG
Basque Center for Applied Mathematics (BCAM)
Bilbao 48009, Spain

Santiago Mazuelas SMAZUELAS@BCAMATH.ORG
Basque Center for Applied Mathematics (BCAM)
IKERBASQUE-Basque Foundation for Science
Bilbao 48009, Spain

Jose A. Lozano JLOZANO@BCAMATH.ORG
Intelligent Systems Group, University of the Basque Country UPV/EHU
Basque Center for Applied Mathematics (BCAM)
San Sebastián, Spain, Bilbao 48009, Spain

Editor: Pierre Alquier

Abstract

Multiple supervised learning scenarios are composed by a sequence of classification tasks. For instance, multi-task learning and continual learning aim to learn a sequence of tasks that is either fixed or grows over time. Existing techniques for learning tasks that are in a sequence are tailored to specific scenarios, lacking adaptability to others. In addition, most of existing techniques consider situations in which the order of the tasks in the sequence is not relevant. However, it is common that tasks in a sequence are evolving in the sense that consecutive tasks often have a higher similarity. This paper presents a learning methodology that is applicable to multiple supervised learning scenarios and adapts to evolving tasks. Differently from existing techniques, we provide computable tight performance guarantees and analytically characterize the increase in the effective sample size. Experiments on benchmark datasets show the performance improvement of the proposed methodology in multiple scenarios and the reliability of the presented performance guarantees.

Keywords: Evolving tasks, Performance guarantees, Minimax risk classification, Supervised classification, Distribution shift

1. Introduction

There are multiple supervised learning scenarios composed by a sequence of tasks (classification problems). These scenarios mainly differ in the specific source tasks that provide information for learning, and the target tasks aimed to be learned (see Figure 1). For instance, in multi-source domain adaptation (MDA), the goal is to learn a target task by leveraging information from all the tasks in the sequence (e.g., Mansour et al. (2009); Li et al. (2022, 2023); Yang et al. (2024)); while in multi-task learning (MTL), the goal is to learn simultaneously the whole sequence by leveraging information from all the tasks (e.g., Zhang and Yang (2018); Lin et al. (2020); Bengio et al. (2009); Pentina et al. (2015); Hu

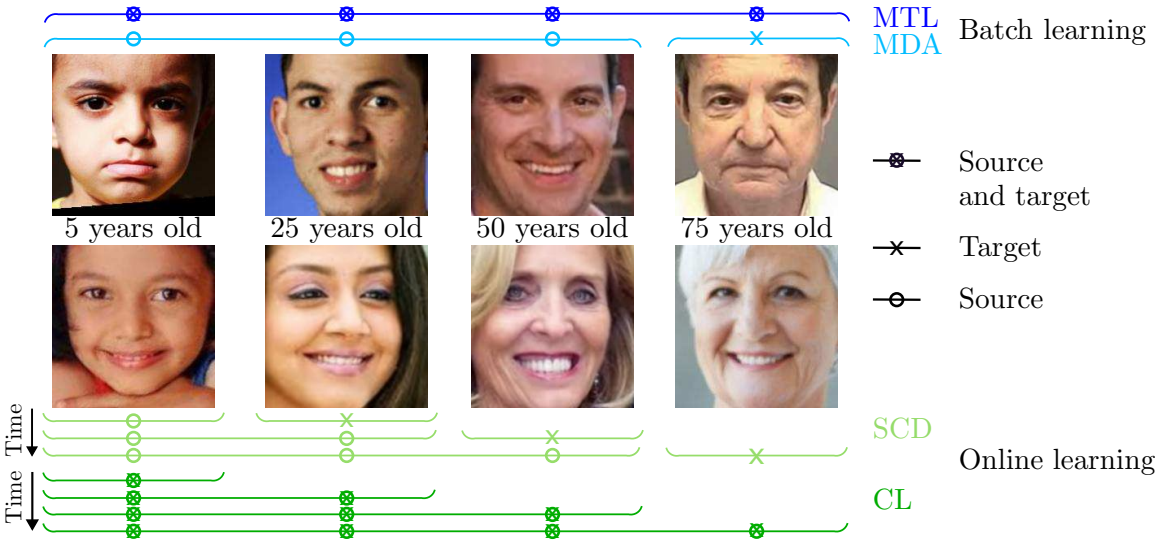


Figure 1: Multiple supervised learning scenarios are composed by a sequence of tasks. Tasks are commonly evolving in the sense that consecutive tasks often have a higher similarity (e.g., gender classification in pictures of people with similar ages). The proposed methodology is applicable to multiple supervised learning scenarios and adapts to evolving tasks.

et al. (2024); Knight and Duan (2024)). Curriculum learning is a specific framework within the broader category of MTL in which tasks are ordered by difficulty, starting with simpler tasks and gradually progressing to more complex ones (Pentina et al., 2015; Weinshall et al., 2018; Bengio et al., 2009; Bell and Lawrence, 2022).

In addition to the batch learning scenarios, there are also online learning scenarios where the sequence of tasks grows over time. For instance, in supervised classification under concept drift (SCD), the goal is to learn at each time step the last task in the sequence by leveraging information from the preceding tasks of the current sequence (e.g., Elwell and Polikar (2011); Brzezinski and Stefanowski (2013); Guo et al. (2024); Fedeli et al. (2023)); while in continual learning (CL), the goal is to learn at each time step the current sequence by leveraging information from all the tasks (e.g., Ruvolo and Eaton (2013); Henning et al. (2021); Thapa and Li (2024); Wang et al. (2024)).

Learning tasks that are in a sequence holds promise to significantly improve performance by leveraging information from different tasks (Ruvolo and Eaton 2013; Lopez-Paz and Ranzato 2017; Chen and Liu 2018; Wang et al. 2024). Such transfer of information can enable accurate classification even in cases where each task has a reduced sample size (e.g., less than 100 samples), thus significantly increasing the effective sample size (ESS) of each task. However, this transfer is hindered by the use of information from tasks characterized by different underlying distributions (e.g., negative transfer and catastrophic forgetting) (Henning et al. 2021; Kirkpatrick et al. 2017; Hurtado et al. 2021; Chen et al. 2023; Evron et al. 2022).

Related work: Existing techniques for learning tasks that are in a sequence are tailored to one specific supervised learning scenario and are not suitable for others. For instance, methods designed for batch learning scenarios utilize samples from all the tasks in the

sequence (Evgeniou and Pontil 2004; Zhang et al. 2015; Knight and Duan 2024; Hu et al. 2024) and are not suitable for online learning scenarios. In addition, methods focused on one target task cannot effectively learn the whole sequence of tasks. Existing techniques designed for each supervised learning scenario follow different strategies. Common MDA methods weight the source samples to minimize the distribution discrepancy between the source and target tasks (Mansour et al. 2009; Bai et al. 2022; Yang et al. 2024). MTL methods usually learn shared representations using samples from all the tasks (Tripuraneni et al., 2020; Zhang and Yang, 2021). Most of existing methods for SCD learn the last task by slightly updating the classification rule from the preceding task (Elwell and Polikar, 2011; Lu et al., 2019). Methods for CL often learn parameters using a pool of stored samples from all the preceding tasks together with the samples from the last task (Lopez-Paz and Ranzato, 2017; Henning et al., 2021).

Most of existing techniques consider situations in which the order of the tasks in the sequence is not relevant (Baxter, 2000; Maurer et al., 2016; Denevi et al., 2019). However, tasks in a sequence are commonly evolving in the sense that consecutive tasks often have a higher similarity (Bartlett, 1992; Hanneke and Yang, 2019; Mazzetto and Upfal, 2023; Pentina and Lampert, 2015). For instance, in the problem of classifying face images of different ages (Zhang et al., 2017), the similarity between consecutive tasks (face images of consecutive ages) is significantly higher (see Figure 1). The existing techniques that adapt to evolving tasks account for a scalar rate of change by using a learning rate (Orabona et al., 2008; Shen et al., 2019), weight factor (Pavlidis et al., 2011; Bai et al., 2022), or window size (Bifet and Gavalda 2007; Pentina and Lampert 2015; Mazzetto and Upfal 2023). Specifically, a slow/fast rate of change is tackled by using a low/high learning rate, weight factor, or smaller/larger window size. More sophisticated techniques designed for SCD account for a time-varying scalar rate of change by using time-varying learning rates (Shen et al., 2019), weight factors (Pavlidis et al., 2011), or window sizes (Bifet and Gavalda 2007; Xie et al. 2023). However, in common practical scenarios, the tasks' changes cannot be adequately addressed accounting only for a scalar rate of change. Such inadequacy is due to the fact that tasks' changes are commonly multidimensional, i.e., different statistical characteristics describing the tasks often change in a different manner.

Most of conventional techniques do not provide performance guarantees since tasks are characterized by different underlying distributions. Existing techniques that provide computable performance guarantees are not designed for evolving tasks (Amit and Meir, 2018; Farid and Majumdar, 2021). On the other hand, existing techniques designed for evolving tasks provide performance guarantees in terms of distributions discrepancies that cannot be computed in practice (Long, 1999; Mohri and Medina, 2012; Pentina and Lampert, 2015).

Our contribution: This paper presents a learning methodology that is applicable to multiple supervised learning scenarios and provides computable tight performance guarantees in terms of error probabilities. In addition, the proposed methodology adapts to evolving tasks accounting for multidimensional changes between consecutive tasks. Specifically, the main contributions in the paper are as follows.

- We establish a learning methodology for evolving tasks based on minimax risk classifiers (MRCs). Such methodology is applicable to scenarios such as MDA, SCD, MTL, and CL.

- We develop learning techniques that provide multidimensional adaptation to evolving tasks by estimating multiple statistical characteristics of the evolving underlying distributions.
- We show that the proposed methodology can provide computable tight performance guarantees for evolving tasks and increase the ESS of each task using information from other tasks.
- We numerically quantify the performance improvement of the proposed methodology in multiple scenarios with reduced sample sizes. In addition, we assess the reliability of the performance guarantees presented.

The rest of the paper is organized as follows. Section 2 briefly describes the problem formulation and MRCs that use linear combinations of general feature mappings. Section 3 presents the learning methodology for evolving tasks and describes the performance guarantees. Section 4 proposes sequential techniques for efficient learning classification rules for all target tasks and Section 5 analytically characterizes the ESS increase. We describe the methods' implementation in scenarios with evolving tasks in Section 6. Then, Section 8 provides multiple numerical results and Section 9 draws the conclusions.

This paper extends the conference papers Álvarez et al. (2022, 2023) by establishing a general methodology with computable performance guarantees applicable to multiple supervised learning scenarios such as MDA, MTL, SCD, and CL. Álvarez et al. (2022) proposed techniques for SCD, while Álvarez et al. (2023) proposed techniques for CL. The additional results presented in this paper are a general learning methodology that is applicable to multiple machine learning scenarios. We present an extension to MDA, MTL, and CL of the results in Álvarez et al. (2022) for SCD that provide multidimensional adaptation to time changes with computable performance guarantees. In addition, we present an extension to MDA, MTL, and SCD of the results in Álvarez et al. (2023) for CL that characterize the increase of the ESS. Furthermore, we assess the evolving assumption and the multidimensional changes in real datasets and present several algorithmic extensions including techniques that account for high-order time dependences among tasks.

2. Preliminaries

This section describes the main notations used in the paper and multiple supervised learning scenarios composed by a sequence of tasks. Then, we briefly describe MRC methods that minimize the worst-case error probability over an uncertainty set. For the readers' convenience, we provide in Table 1 a list with the main notions used in the paper and their corresponding notations.

Notation Calligraphic letters represent sets; bold lowercase letters represent vectors; bold uppercase letters represent matrices; $\mathbf{I}$ and $\mathbb{I}\{\cdot\}$ denote the identity matrix and the indicator function, respectively; $(\cdot)_+$ denotes the positive part of its argument; $\text{sign}(\cdot)$ denotes the vector given by the signs of the argument components; $\mathbf{e}_i$ denotes the i-th vector in a standard basis, $\|\cdot\|_1$ and $\|\cdot\|_\infty$ denote the 1-norm and the infinity norm of its argument, respectively; $\preceq$ and $\succeq$ denote vector inequalities; and $\mathbb{E}_{\text{p}}\{\cdot\}$ and $\mathbb{V}\text{ar}_{\text{p}}\{\cdot\}$ denote the component-wise expectation and the component-wise variance of its argument with respect

Table 1: Main notations used in the paper

Notation	Meaning
$\Delta(\mathcal{X} \times \mathcal{Y})$	set of probability distributions over instances $\mathcal{X}$ and labels $\mathcal{Y}$
$\mathrm{T}(\mathcal{X}, \mathcal{Y})$	set of classification rules from instances $\mathcal{X}$ to labels $\mathcal{Y}$
$\mathrm{h}(y\|x)$	probability assigned by classification rule h to label y for instance x
$\ell(\mathrm{h}, \mathrm{p})$	expected 0-1 loss of classification rule h w.r.t. distribution p
$R(\mathrm{h})$	risk (error probability) of classification rule h
$\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$	feature mapping with dimension m
M	bound of the feature mapping ($\\|\Phi(x, y)\\|_\infty \leq M$)
$\mathcal{U}$	uncertainty set of distributions given by expectations' constraints as in (1)
p_j	underlying distribution that characterizes the j-th task
D_j	sample set of the j-th task
n_j	sample size of D_j
Φ_j	random variable given by the feature mapping of samples from the j-th task
$\boldsymbol{\sigma}_j$	estimate of standard deviation of the j-th task
$\boldsymbol{\tau}_j^\infty$	expectation of feature mapping corresponding to the j-th task
$\boldsymbol{w}_j = \boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_{j-1}^\infty$	change between tasks
$\boldsymbol{d}_j$	estimate of the expected quadratic change between the j-th task and the $j-1$-th task
$\mathcal{U}_j^\infty$	uncertainty set of the j-th task given by the true expectation
R_j^∞	minimum worst-case error probability of the j-th task
$\boldsymbol{\mu}_j^\infty$	classifier parameter of the j-th task corresponding with $\mathcal{U}_j^\infty$
$\boldsymbol{\mu}_j$	classifier parameter of the j-th task
h_j	classification rule using single-task learning
h_j^j	classification rule using forward learning
h_j^k	classification rule using forward and backward learning in a sequence of k tasks
$\mathcal{U}_j, \mathcal{U}_j^j, \mathcal{U}_j^k$	uncertainty sets using single-task, forward, and forward and backward learning
$R(\mathcal{U}_j)$, $R(\mathcal{U}_j^j)$, $R(\mathcal{U}_j^k)$	minimax risks using single-task, forward, and forward and backward learning
$\boldsymbol{\tau}_j$, $\boldsymbol{\tau}_j^j$, $\boldsymbol{\tau}_j^k$	mean vector using single-task, forward, and forward and backward learning
$\boldsymbol{\lambda}_j$, $\boldsymbol{\lambda}_j^j$, $\boldsymbol{\lambda}_j^k$	confidence vectors using single-task, forward, and forward and backward learning
$\boldsymbol{s}_j$, $\boldsymbol{s}_j^j$, $\boldsymbol{s}_j^k$	MSE vectors using single-task, forward, and forward and backward learning
$\boldsymbol{\eta}_j^j, \boldsymbol{\eta}_j^k$	smoothing gains using forward, and forward and backward learning
n_j^j, n_j^k	ESSs using forward, and forward and backward learning

to distribution p. For a vector $\mathbf{v}$, $v^{(i)}$ and $\mathbf{v}^\top$ denote the i-th component and the transpose of $\mathbf{v}$, respectively. In addition, non-linear operators acting on vectors denote component-wise operations. For instance, $|\mathbf{v}|$ and $\mathbf{v}^2$ denote the vector formed by the absolute value and the square of each component, respectively, and for a vector $\mathbf{u}$, $\mathbf{v}/\mathbf{u}$ denotes the vector formed by the component-wise division.

2.1 Problem Formulation

Let $\mathcal{X}$ be a set of instances and $\mathcal{Y}$ be a set of labels with $\mathcal{Y}$ represented by $\{1, 2, \ldots, |\mathcal{Y}|\}$, we denote by $\Delta(\mathcal{X} \times \mathcal{Y})$ the set of probability distributions over $\mathcal{X} \times \mathcal{Y}$. In addition, we

denote by $\mathrm{T}(\mathcal{X}, \mathcal{Y})$ the set of classification rules and by $\mathrm{h}(y|x)$ the probability with which classification rule $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ assigns label $y \in \mathcal{Y}$ to instance $x \in \mathcal{X}$ ($\mathrm{h}(y|x) \in \{0, 1\}$ for deterministic classification rules). Supervised classification techniques use a sample set $D = \{(x_i, y_i)\}_{i=1}^{n}$ formed by n i.i.d. samples from distribution $\mathrm{p}^* \in \Delta(\mathcal{X} \times \mathcal{Y})$ to find a classification rule $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ with small expected loss $\ell(\mathrm{h}, \mathrm{p}^*)$. In the following, we utilize the 0-1-loss so that the expected loss of the classification rule h with respect to the underlying distribution is the error probability of such rule h that is denoted by $R(\mathrm{h})$.

In the addressed setting, there are multiple sample sets $D_1, D_2, \ldots$ where each sample set D_j is composed by n_j i.i.d. instance-label pairs. Sample sets $D_1, D_2, \ldots$ correspond to different classification tasks characterized by underlying distributions $\mathrm{p}_1, \mathrm{p}_2, \ldots$. Learning methods use the available sample sets to obtain classification rules for the target tasks. Such setting describes multiple supervised learning scenarios that mainly differ in the specific target tasks they aim to learn and the source tasks from which they acquire information for learning (see Fig. 1). For instance, MDA methods use sample sets $D_1, D_2, \ldots, D_{k-1}$ from $k-1$ source tasks to obtain a classification rule h_k for the k-th target task; while MTL methods use k sample sets $D_1, D_2, \ldots, D_k$ to obtain classification rules $\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_k$ so all the tasks in the sequence are both source and target. In addition to these batch learning scenarios, the above setting also describes online learning scenarios where sample sets $D_1, D_2, \ldots$ corresponding with different tasks arrive over time. For instance, SCD methods use, at each step k, sample sets $D_1, D_2, \ldots, D_{k-1}$ from $k-1$ source tasks to obtain a classification rule h_k for the k-th target task; while CL methods use, at each step k, sample sets $D_1, D_2, \ldots, D_k$ to obtain classification rules $\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_k$ so all the tasks in the current sequence are both source and target. In this paper, we establish a general methodology for learning tasks in a sequence that can be used in all the above mentioned scenarios.

Most existing techniques are designed for situations where tasks' similarities do not depend on the order of the task in the sequence (Baxter, 2000; Maurer et al., 2016; Denevi et al., 2019). These settings are usually mathematically modeled assuming that the tasks' distributions

$$\mathrm{p}_1, \mathrm{p}_2, \ldots \text{ are independent and identically distributed.} \tag{i.i.d.-A}$$

In particular, each distribution p_j for $j = 1, 2, \ldots$ is sampled from a fixed task meta-distribution (Baxter, 2000; Maurer et al., 2016; Denevi et al., 2019). In this paper, we develop techniques designed for evolving tasks where consecutive tasks are significantly more similar. These settings can be mathematically modeled assuming that the tasks' distributions

$$\mathrm{p}_1, \mathrm{p}_2, \ldots \text{ form a random walk with independent and zero-mean increments.} \tag{Evo-A}$$

In this case, each distribution p_j for $j = 1, 2, \ldots$ is sampled from a task meta-distribution that is continuously evolving over tasks. Under the (Evo-A) assumption, we have that $\mathrm{p}_{j+1} = \mathrm{p}_j + \varepsilon_{j+1}$ where ε_{j+1} for $j = 1, 2, \ldots$ are independent and zero-mean. The (Evo-A) assumption is similar to the "changing task environments" assumption used in Pentina and Lampert (2015) and the distribution change between consecutive tasks in Bartlett (1992); Hanneke and Yang (2019). However, these papers assume that the distribution slightly changes between consecutive tasks, while we assume that the distribution evolves as a random walk with independent increments. Section 3 below further shows how the assumption

used in the paper can describe common real datasets better than the conventional (i.i.d.-A) assumption.

2.2 Minimax risk classifiers

MRCs are based on robust risk minimiztion (RRM) also known as distributionally robust learning (Farnia and Tse; Fathony et al., 2016). RRM-based methods differ in the uncertainty set considered. For instance, Shafieezadeh-Abadeh et al. (2019) utilize uncertainty sets determined by Wasserstein distances and Duchi and Namkoong (2019) utilize uncertainty sets determined by f-divergences. However, these uncertainty sets are not easy to obtain in scenarios with varying distributions. MRCs (Mazuelas et al., 2020, 2023) determine uncertainty sets using expectation estimates that can be effectively obtained in the addressed settings. In particular, these uncertainty sets are especially suitable for learning from a sequence of tasks because they can effectively leverage samples from multiple tasks characterized by different distributions.

The uncertainty set used in the proposed methods is given by constraints on the expectation of a feature mapping $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$ as

$$\mathcal{U} = \{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : |\mathbb{E}_{\mathrm{p}}\{\Phi(x, y)\} - \boldsymbol{\tau}| \preceq \boldsymbol{\lambda}\} \tag{1}$$

where $|\cdot|$ denotes the vector formed by the absolute value of each component in the argument, $\boldsymbol{\tau}$ denotes the vector of expectation estimates corresponding with the feature mapping Φ, and $\boldsymbol{\lambda} \succeq \mathbf{0}$ is a confidence vector that accounts for inaccuracies in the estimate. Feature mappings are vector-valued functions over $\mathcal{X} \times \mathcal{Y}$. For instance, such mappings can be defined by multiple features over instances together with a one-hot encoding of labels, as follows (Mohri et al., 2018)

$$\Phi(x, y) = \mathbf{e}_y \otimes \Psi(x) = [\mathbb{I}\{y = 1\}\Psi(x)^\top, \mathbb{I}\{y = 2\}\Psi(x)^\top, ..., \mathbb{I}\{y = |\mathcal{Y}|\}\Psi(x)^\top]^\top \tag{2}$$

where $\mathbf{e}_y$ is the y-th vector in the standard basis of $\mathbb{R}^{|\mathcal{Y}|}$, $\otimes$ denotes the Kronecker product, and the map $\Psi : \mathcal{X} \to \mathbb{R}^q$ represents instances as real vectors. The feature mapping Φ represents each instance-label pair (x, y) by an m-dimensional real vector with $m = |\mathcal{Y}|q$, so that $\Phi(x, y)$ is composed by $|\mathcal{Y}|$ q-dimensional blocks with values $\Psi(x)$ in the block corresponding to y and zero otherwise. General types of scalar features can be used for MRC learning including those given by thresholds/decision stumps (Lebanon and Lafferty, 2001), the last layer in a neural network (NN) (Bengio et al., 2013), and random features corresponding to a reproducing kernel Hilbert space (Rahimi and Recht, 2007).

Given the uncertainty set $\mathcal{U}$, MRC rules minimize the worst-case error probability and are solutions of the optimization problem

$$\min_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \max_{\mathrm{p} \in \mathcal{U}} \ell(\mathrm{h}, \mathrm{p}) \tag{3}$$

where $\ell(\mathrm{h}, \mathrm{p})$ denotes the expected loss of classification rule h for distribution p. In the following, we utilize the 0-1-loss so that $\ell(\mathrm{h}, \mathrm{p}) = \mathbb{E}_{\mathrm{p}}\{1 - \mathrm{h}(y|x)\}$ and the expected loss with respect to the underlying distribution becomes the error probability of the classification rule. In addition, the optimal value of (3) is denoted by $R(\mathcal{U})$ and referred to as the minimax risk for uncertainty set $\mathcal{U}$.

The MRC rule h assigns label $\hat{y} \in \mathcal{Y}$ to instance $x \in \mathcal{X}$ with probability given by linear-affine combinations of the components of the feature mapping as (see eq. (11) in Mazuelas et al. (2023))

$$\mathrm{h}(\hat{y}|x) = \begin{cases} \left(\Phi(x,\hat{y})^\top \boldsymbol{\mu}^* - \varphi(\boldsymbol{\mu}^*)\right)_+ / c_x & \text{if } c_x \neq 0 \\ 1/|\mathcal{Y}| & \text{if } c_x = 0 \end{cases} \tag{4}$$

with

$$\varphi(\boldsymbol{\mu}^*) = \max_{x\in\mathcal{X},\mathcal{C}\subseteq\mathcal{Y}} \Big(\sum_{y\in\mathcal{C}} \Phi(x,y)^\top \boldsymbol{\mu}^* - 1 \Big)/|\mathcal{C}|, \;\; c_x = \sum_{y\in\mathcal{Y}} \Big(\Phi(x,y)^\top \boldsymbol{\mu}^* - \varphi(\boldsymbol{\mu}^*) \Big)_+ .$$

The vector parameter $\boldsymbol{\mu}^*$ is the solution of the convex optimization problem (see eq. (6) in Mazuelas et al. (2023))

$$\min_{\boldsymbol{\mu}} \; 1 - \boldsymbol{\tau}^\top \boldsymbol{\mu} + \varphi(\boldsymbol{\mu}) + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \tag{5}$$

given by the Fenchel-Lagrange dual of (3) (Mazuelas et al., 2020, 2022); and parameters $\boldsymbol{\mu}^*$ correspond to the Lagrange multipliers of constraints in (1). Note that the label that maximizes the probability in (4) is given by $\hat{y} \in \arg\max_{y\in\mathcal{Y}} \Phi(x,y)^\top \boldsymbol{\mu}^*$. The deterministic classification rule h^{d} that assigns such label $\hat{y}$ to instance x will be referred in the following as deterministic MRC. MRCs are specifically designed for classification problems. The extension of MRCs to regression problems is not straightforward because MRCs do not consider a parametric space of possible rules.

The baseline approach of single-task learning obtains a classification rule h_j for each j-th task leveraging information only from the sample set $D_j = \{(x_{j,i}, y_{j,i})\}_{i=1}^{n_j}$ given by n_j i.i.d. samples from distribution p_j. For single-task learning, MRCs can obtain mean and confidence vectors as (Mazuelas et al., 2020)

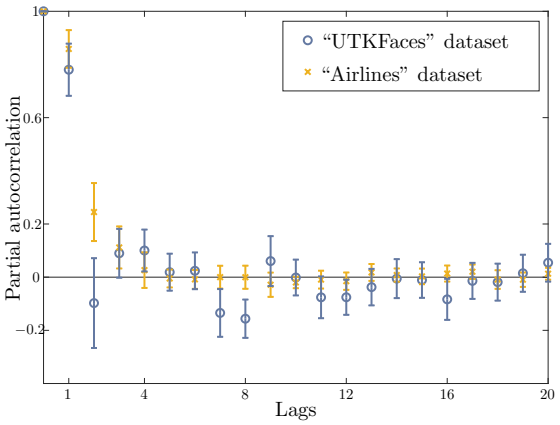
$$\boldsymbol{\tau}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \Phi\left(x_{j,i}, y_{j,i}\right), \quad \boldsymbol{s}_j = \frac{\boldsymbol{\sigma}_j^2}{n_j}, \quad \boldsymbol{\lambda}_j = \lambda_0 \sqrt{\boldsymbol{s}_j} \;\; \in \;\; \mathbb{R}^m \tag{6}$$

where $\lambda_0 > 0$ controls the regularization strength in the methods proposed. The mean vector $\boldsymbol{\tau}_j$ corresponds to the sample mean and provides an unbiased estimator of the feature mapping expectation $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x,y)\}$. The vector $\boldsymbol{\sigma}_j^2$ is given by an estimate of the feature mapping variance. Specifically, the i-th component of vector $\boldsymbol{\sigma}_j^2$ is an estimate of the variance of the i-th component of the feature mapping, denoted by $\mathbb{V}\mathrm{ar}_{\mathrm{p}_j}\{\Phi^{(i)}(x,y)\}$. In particular, if $\boldsymbol{\sigma}_j^2$ is given by the sample variance the vector $\boldsymbol{s}_j$ in (6) corresponds to the mean squared errors (MSEs) of the mean vector $\boldsymbol{\tau}_j$.

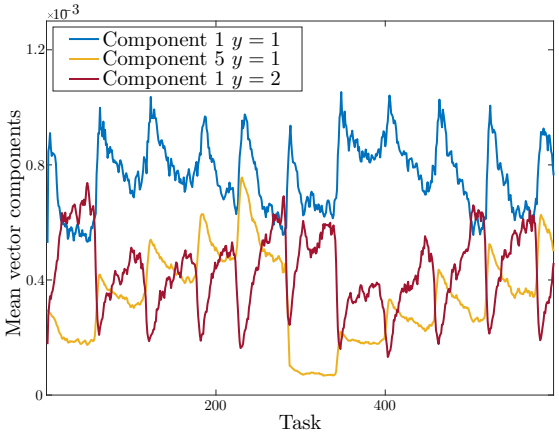
In the following sections, we describe techniques that obtain the mean and MSE vectors in scenarios that are composed by a sequence of tasks. Once such vectors are obtained, the proposed methodology obtains the classifier parameter $\boldsymbol{\mu}_j$ for each j-th task solving the convex optimization problem in (5) that can be efficiently addressed using conventional methods (Nesterov and Shikhman, 2015; Tao et al., 2019).

3. Learning evolving tasks

This section assesses the evolving assumption in Section 2 and presents the methodology for learning evolving tasks.



(a) Averaged partial autocorrelation of mean vectors components +/- their standard deviations shows the better adequacy of the (Evo-A) assumption for evolving tasks.



(b) Mean vector components using "Airlines" dataset shows the multidimensional changes in tasks' distributions.

Figure 2: In common practical scenarios, tasks are evolving and tasks' changes are multidimensional.

3.1 Evolving tasks

In the following, we describe the main aspects of evolving tasks and the inadequacy of conventional modeling assumptions. In particular, we analyze the similarities among underlying distributions p_j, for $j = 1, 2, \ldots$, by assessing the similarities among the corresponding vectors formed by the expectations of a feature mapping. For each j-th task, such a mean vector $\boldsymbol{\tau}_j^{\infty} = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x, y)\}$ represents the statistical characteristics of the underlying distribution p_j as measured by the feature mapping $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^m$.

For a sequence of evolving tasks, the similarity between two tasks depends on their distance in the sequence. In particular, the similarity between p_{j+t} and p_j is often higher for small t, and consecutive tasks ($t = 1$) are significantly more similar. This similarity between consecutive tasks can be better described by the (Evo-A) assumption in Section 2.1 instead of the usual (i.i.d.-A) assumption. The (Evo-A) assumption considers independent changes between consecutive distributions, while the (i.i.d.-A) assumption considers independent distributions. Note that if the tasks' distributions satisfy (Evo-A), the difference between the distributions of the j-th and the $(j + t)$-th tasks has zero-mean and variance $\mathbb{V}\mathrm{ar}\{\mathrm{p}_{j+t} - \mathrm{p}_j\} = \sum_{i=1}^{t} \mathbb{V}\mathrm{ar}\{\mathrm{p}_{j+i} - \mathrm{p}_{j+i-1}\}$ that increases with t. On the other hand, if the tasks' distributions satisfy (i.i.d.-A), the difference between the distributions of the j-th and the $(j + t)$-th tasks has zero-mean and variance $\mathbb{V}\mathrm{ar}\{\mathrm{p}_{j+t} - \mathrm{p}_j\} = \mathbb{V}\mathrm{ar}\{\mathrm{p}_{j+1} - \mathrm{p}_j\} = 2\mathbb{V}\mathrm{ar}\{\mathrm{p}_1\}$ that does not depend on t.

The better adequacy of the (Evo-A) assumption for evolving tasks can be assessed in real datasets by using partial autocorrelations, as shown in Figure 2a. This figure shows the averaged partial autocorrelation of the mean vector components +/- their standard deviations for different lags using "Airlines" and "UTKFaces" datasets (see dataset characteristics in Table 4 of Appendix I). The "Airlines" dataset is composed by tasks corresponding to airplane delays of a specific time period and the "UTKFaces" dataset is composed by tasks

corresponding to the classification of face images of a specific age (see also Fig. 1). The partial autocorrelation is a usual measure of the correlation between observations of a process (see Section 4 in Cowpertwait and Metcalfe (2009)). For instance, the partial autocorrelation indicates that a process is a random walk if there are significant correlations at the first lag, followed by much smaller correlations; while the partial autocorrelation indicates that the observations are i.i.d. if the correlations are not significant at any lag. Figure 2a shows that the partial autocorrelation of the sequence of mean vectors are clearly larger than zero at lag 1. Hence, the figure shows that the (Evo-A) assumption better describes the tasks' distributions than the (i.i.d.-A) assumption in real datasets.

Figure 2a also shows that higher-order dependences among tasks can be adequate for certain datasets. For instance, the figure shows that the partial autocorrelations of the mean vector components are larger than zero at lag 2 for "Airlines" dataset. Accounting for higher-order dependences can better capture the similarities among tasks in certain datasets. Most of the results in the paper are described for the simple case that accounts for first order similarities between consecutive tasks via the above discussed (Evo-A) assumption. Section 7 shows extensions of the proposed methodology accounting for higher-order dependences.

Figure 2b shows that the changes in tasks' distributions are often multidimensional, that is, different statistical characteristics of the underlying distribution often change in a different manner. This figure shows 3 mean vector components (component 1 for $y = 1$, component 5 for $y = 1$, and component 1 for $y = 2$) for different tasks using "Airlines" dataset. Such figure illustrates a clearly different change in each mean vector component that reflects multidimensional tasks' changes. For instance, the expected values of the first instance component for class $y = 2$ (component 1 $y = 2$) often increases in cases where other components decrease. In addition, the expected values of the fifth component for class $y = 1$ (component 5 $y = 1$) exhibit slower changes than those for component 1 with $y = 1$. Existing methods for evolving tasks account for a scalar rate of change that cannot capture the multidimensional tasks' changes. In the following, we propose techniques that account for the change in each component in the mean vector by using a vector $\boldsymbol{d}_j$ that assesses the expected quadratic change between consecutive tasks.

3.2 Learning methodology

Figure 3 depicts the flow diagram of the proposed learning methodology that is applicable to multiple supervised learning scenarios and provides computable tight performance guarantees.

For a sequence of k tasks, the proposed learning methodology obtains an MRC rule for each j-th task leveraging information from the k tasks and accounting for the change between consecutive tasks. Specifically, for each j-th task, the proposed methods obtain an uncertainty set $\mathcal{U}_j^k$ as in (1) by using uncertainty sets, $\mathcal{U}_{j-1}^k$, $\mathcal{U}_{j+1}^k$, for adjacent tasks and the expected quadratic change between consecutive tasks $\boldsymbol{d}_j$, $\boldsymbol{d}_{j+1}$ (see Fig. 3). Then, given the uncertainty set $\mathcal{U}_j^k$, we obtain the classification rule h_j^k and the minimax risk $R(\mathcal{U}_j^k)$ solving the convex optimization problem (5). For instance, in online learning scenarios such as SCD, at each step k an uncertainty set $\mathcal{U}_k^k$ for each k-th task can be obtained by using $\mathcal{U}_{k-1}^{k-1}$ and $\boldsymbol{d}_k$; then such uncertainty set is used to obtain the classification rule h_k^k and the

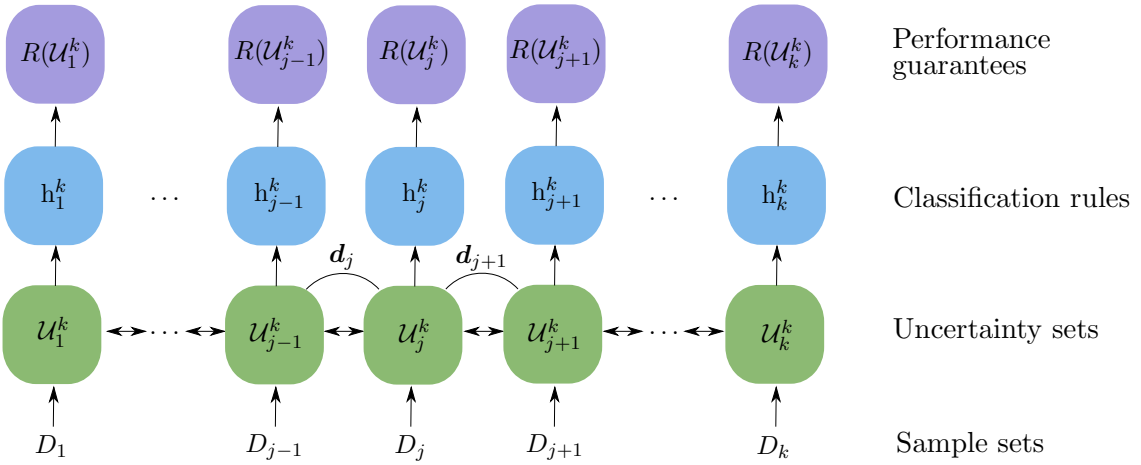


Figure 3: The proposed methodology obtains an uncertainty set $\mathcal{U}_j^k$ for each j-th task using the sample set D_j, the adjacent uncertainty sets $\mathcal{U}_{j-1}^k, \mathcal{U}_{j+1}^k$, and the change between consecutive tasks $\boldsymbol{d}_j, \boldsymbol{d}_{j+1}$. Then, the uncertainty set is used to obtain the classification rule h_j^k together with the minimax risk $R(\mathcal{U}_j^k)$ that directly gives performance guarantees.

minimax risk $R(\mathcal{U}_k^k)$. Section 4 describes techniques that recursively obtain uncertainty sets for each task accounting for multidimensional tasks' changes and Section 6 describes techniques to solve the convex optimization problem in (5) based on subgradient methods (Nesterov and Shikhman, 2015; Tao et al., 2019).

The proposed methodology provides computable tight performance guarantees for the error probability of each j-th task by assessing the minimax risks. Let $R(\mathrm{h}_j^k)$ denote the error probability of the classification rule h_j^k that is determined by the parameter $\boldsymbol{\mu}_j^k$. Then, using the bounds of Theorem 7 in Mazuelas et al. (2022) in the addressed setting, we have that

$$R(\mathrm{h}_j^k) \leq R(\mathcal{U}_j^k) + \left(|\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_j^k| - \boldsymbol{\lambda}_j^k \right)^\top |\boldsymbol{\mu}_j^k| \tag{7}$$

where $R(\mathcal{U}_j^k)$ and $\mathcal{U}_j^k$ denote the minimax risk and the uncertainty set given as in (1) by mean and confidence vectors $\boldsymbol{\tau}_j^k$ and $\boldsymbol{\lambda}_j^k$. The above inequality provides computable tight bounds for error probabilities given by the minimax risk. Specifically, the minimax risk $R(\mathcal{U}_j^k)$ directly provides a bound if the error in the mean vector estimate satisfies $|\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_j^k| \preceq \boldsymbol{\lambda}_j^k$. In other cases, the minimax risk $R(\mathcal{U}_j^k)$ still provides approximate bounds as long as the difference $|\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_j^k| - \boldsymbol{\lambda}_j^k$ is not substantial. Bound in (7) is tight in the sense that it would coincide with the MRC error if the underlying distribution is the worst-case distribution in $\mathcal{U}_j^k$ (the distribution in $\mathcal{U}_j^k$ with the largest Bayes risk). Section 8 shows that the presented bounds can also provide tight performance guarantees in practice.

The learning methodology is applicable to multiple supervised learning scenarios including online (SCD and CL) and batch learning scenarios (MDA and MTL). The proposed techniques effectively use information from all the tasks in the sequence to obtain a classification rule for the last task in SCD and MDA or a sequence of classification rules in MTL

and CL. Section 6 describes the implementation of the proposed methodology in multiple scenarios and describes techniques accounting for higher-order dependences among tasks.

4. Determining uncertainty sets from sample sets of evolving tasks

This section presents techniques to recursively obtain uncertainty sets as in (1) accounting for the evolution of the tasks' distribution and leveraging information from the tasks in the sequence. Specifically, the proposed techniques recursively obtain mean and MSE vectors for each j-th task leveraging information from preceding tasks (forward learning) and, reciprocally, obtain mean and MSE for preceding tasks leveraging information from the succeeding tasks in the sequence (backward learning).

4.1 Forward learning

This section presents the recursions that allow us to leverage information from preceding tasks. For a sequence of k tasks, the proposed techniques first obtain for each j-th task with $j = \{1, 2, \ldots, k\}$ forward mean and MSE vectors leveraging information up to the j-th task. Let $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ denote the mean and MSE vectors for forward learning corresponding to the j-th task. The following recursions allow us to obtain $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j \in \mathbb{R}^m$ for each j-th task using those for the preceding task $\boldsymbol{\tau}_{j-1}^{j-1}, \boldsymbol{s}_{j-1}^{j-1}$ together with a smoothing gain $\boldsymbol{\eta}_j^j$ as follows

$$\boldsymbol{\tau}_j^j = \boldsymbol{\tau}_{j-1}^{j-1} + \boldsymbol{\eta}_j^j \left(\boldsymbol{\tau}_j - \boldsymbol{\tau}_{j-1}^{j-1} \right) \tag{8}$$

$$\boldsymbol{s}_j^j = \boldsymbol{\eta}_j^j \boldsymbol{s}_j \tag{9}$$

$$\boldsymbol{\eta}_j^j = \frac{\boldsymbol{s}_{j-1}^{j-1} + \boldsymbol{d}_j}{\boldsymbol{s}_j + \boldsymbol{s}_{j-1}^{j-1} + \boldsymbol{d}_j} \tag{10}$$

where non-linear operators acting on vectors of the same dimension denote component-wise operations, the vectors $\mathrm{t}\boldsymbol{\tau}_j, \boldsymbol{s}_j$ are given by (6) and $\boldsymbol{\tau}_1^1 = \boldsymbol{\tau}_1$, $\boldsymbol{s}_1^1 = \boldsymbol{s}_1$. Mean and MSE vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ are obtained leveraging information up to the j-th task from sample sets $D_1, D_2, \ldots, D_j$. Specifically, for each j-th task, the vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ are obtained by acquiring information from sample set D_j through mean and MSE vectors $\boldsymbol{\tau}_j$ and $\boldsymbol{s}_j$ and retaining information from sample sets $D_1, D_2, \ldots, D_{j-1}$ through mean and MSE vectors $\boldsymbol{\tau}_{j-1}^{j-1}$ and $\boldsymbol{s}_{j-1}^{j-1}$.

Recursions in (8)-(9) can be applied to multiple supervised learning scenarios as detailed in Section 6. For instance, the proposed methodology applied to CL obtains forward mean vector $\boldsymbol{\tau}_k^k$ for the k-th task leveraging information from preceding tasks using the forward mean vector $\boldsymbol{\tau}_{k-1}^{k-1}$, the sample average $\boldsymbol{\tau}_k$, and the expected quadratic change between consecutive tasks $\boldsymbol{d}_k$. Specifically, the mean vector $\boldsymbol{\tau}_k^k$ is obtained by adding a correction to the sample average $\boldsymbol{\tau}_k$. This correction is proportional to the difference between $\boldsymbol{\tau}_k$ and $\boldsymbol{\tau}_{k-1}^{k-1}$ multiplied by the smoothing gain $\boldsymbol{\eta}_k^k$ that depends on the MSE vectors $\boldsymbol{s}_k, \boldsymbol{s}_{k-1}^{k-1}$ and the expected quadratic change $\boldsymbol{d}_k$. In particular, if $\boldsymbol{s}_k \ll \boldsymbol{s}_{k-1}^{k-1} + \boldsymbol{d}_k$, the mean vector is given by the sample average as in single-task learning, and if $\boldsymbol{s}_k \gg \boldsymbol{s}_{k-1}^{k-1} + \boldsymbol{d}_k$, the mean vector is given by that of the preceding task.

The result below shows that using exact values for variances and expected quadratic changes, the previous recursions obtain the unbiased linear estimates of the mean vector $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x,y)\}$ with minimum MSE.

Theorem 1 *Let $\boldsymbol{\sigma}_j^2$ and $\boldsymbol{d}_j$ in recursions* (8)-(10) *be the actual variance and the actual expected quadratic change between tasks, that is $\boldsymbol{\sigma}_j^2 = [\mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(1)}(x,y)\}, \mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(2)}(x,y)\}, \ldots, \mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(m)}(x,y)\}]^\top$ and $\boldsymbol{d}_j = \mathbb{E}\{\boldsymbol{w}_j^2\} = \mathbb{E}\{(\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_{j-1}^\infty)^2\}$. If the tasks' distributions satisfy* (Evo-A)*, then we have that $\boldsymbol{\tau}_j^j$ given by* (8) *is the unbiased linear estimator of the mean vector $\boldsymbol{\tau}_j^\infty$ based on $D_1, D_2, \ldots, D_j$ that has the minimum MSE, and $\boldsymbol{s}_j^j$ given by* (9) *is its MSE.*

Proof *The mean vectors evolve over tasks through the linear dynamical system (state-space model with white noise processes)*

$$\boldsymbol{\tau}_j^\infty = \boldsymbol{\tau}_{j-1}^\infty + \boldsymbol{w}_j \tag{11}$$

where vectors $\boldsymbol{w}_j$ for $j = 2, 3, \ldots$ are independent and zero-mean. Equation (11) *is obtained because the sequence of probability distribution satisfy the* (Evo-A) *assumption, that is $\mathrm{p}_j = \mathrm{p}_{j-1} + \varepsilon_j$ where ε_j for $j = 2, 3, \ldots$ are independent and zero-mean. In addition, each state variable $\boldsymbol{\tau}_j^\infty$ is observed at each step j through $\boldsymbol{\tau}_j$ so that we have*

$$\boldsymbol{\tau}_j = \boldsymbol{\tau}_j^\infty + \boldsymbol{v}_j \tag{12}$$

where vectors $\boldsymbol{v}_j$ for $j \in \{2, 3, \ldots, k\}$ are independent and zero-mean because $\boldsymbol{\tau}_j$ is the sample average of i.i.d. samples. For the above dynamical systems, the Kalman filter recursions provide the unbiased linear estimator with minimum MSE based on samples corresponding to preceding steps $D_1, D_2, \ldots, D_j$. Then, equations (8) *and* (9) *are obtained after some algebra from the Kalman filter recursions (see details in Appendix A).* ■

The above theorem shows that equations in (8)-(10) enable to recursively obtain the mean vector estimate for each task as well as its MSE vector leveraging information from preceding tasks.

The vectors $\boldsymbol{d}_j$ and $\boldsymbol{\sigma}_j^2$ in the above recursions can be estimated for each j-th task using sample sets. As described in Section 3, the vector $\boldsymbol{d}_j$ assesses the expected quadratic change between consecutive tasks $\mathbb{E}\{\boldsymbol{w}_j^2\} = \mathbb{E}\{(\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_{j-1}^\infty)^2\}$. Such expectation can be estimated by the sample average of the most recent samples as

$$\boldsymbol{d}_j = \frac{1}{W}\sum_{l=1}^{W}\left(\boldsymbol{\tau}_{j_l} - \boldsymbol{\tau}_{j_{l-1}}\right)^2 \tag{13}$$

where $j_0, j_1, \ldots, j_W$ are the $W+1$ closest indexes to j and sample average $\boldsymbol{\tau}_j$ is given by (6). The vector $\boldsymbol{\sigma}_j^2$ assesses the variance of the samples of the j-th task $\mathbb{V}\mathrm{ar}_{\mathrm{p}_j}\{\Phi(x,y)\}$. Such variance can be estimated by the sample variance of sample set D_j. As shown in Section 5 below, the usage of approximated values for $\boldsymbol{d}_j$ and $\boldsymbol{\sigma}_j^2$ does not significantly affect the performance of the methods proposed.

4.2 Forward and backward learning

This section presents the recursions that allow us to obtain mean and MSE vectors for each task leveraging information from preceding and succeeding tasks. The proposed techniques obtain for each j-th task with $j = \{1, 2, \ldots, k\}$ forward and backward mean and MSE vectors leveraging information from the k tasks in the sequence. Let $\boldsymbol{\tau}_j^k$ and $\boldsymbol{s}_j^k$ denote the mean and MSE vectors for forward and backward learning corresponding to the j-th task. The following recursions allow us to obtain $\boldsymbol{\tau}_j^k, \boldsymbol{s}_j^k \in \mathbb{R}^m$ using forward mean and MSE vectors $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j$ together with a smoothing gain $\boldsymbol{\eta}_j^k$ as follows

$$\boldsymbol{\tau}_j^k = \boldsymbol{\tau}_{j+1}^k + \boldsymbol{\eta}_j^k \left(\boldsymbol{\tau}_j^j - \boldsymbol{\tau}_{j+1}^k \right) \tag{14}$$

$$\boldsymbol{s}_j^k = \boldsymbol{s}_{j+1}^k + \boldsymbol{\eta}_j^k (\boldsymbol{s}_j^j - 2\boldsymbol{s}_{j+1}^k + \boldsymbol{\eta}_j^k \boldsymbol{s}_{j+1}^k) \tag{15}$$

$$\boldsymbol{\eta}_j^k = \frac{\boldsymbol{d}_{j+1}}{\boldsymbol{s}_j^j + \boldsymbol{d}_{j+1}} \tag{16}$$

for $j = k-1, k-2, \ldots, 1$ with $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j$ given by (8)-(9). Mean and MSE vectors $\boldsymbol{\tau}_j^k, \boldsymbol{s}_j^k$ are obtained leveraging information from all the sample sets $D_1, D_2, \ldots, D_k$. Specifically, for each j-th task, the vectors $\boldsymbol{\tau}_j^k, \boldsymbol{s}_j^k$ are obtained acquiring information from sample sets $D_{j+1}, D_{j+2}, \ldots, D_k$ through mean and MSE vectors $\boldsymbol{\tau}_{j+1}^k, \boldsymbol{s}_{j+1}^k$ and retaining information from sample sets $D_1, D_2, \ldots, D_j$ through mean and MSE vectors $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j$.

Similarly as the forward learning recursions in (8)-(9), forward and backward learning recursions in (14)-(15) can be applied to multiple supervised learning scenarios as detailed in Section (6). At each step k, the proposed methodology applied to CL obtains the mean and MSE vectors for the j-th task by retaining information from preceding tasks and acquiring information from the new task. First, we obtain mean and MSE vectors $\boldsymbol{\tau}_k^k$ and $\boldsymbol{s}_k^k$, in that case forward and backward learning recursions in (14)-(15) coincide with forward learning recursions in (8)-(9). Then, we obtain mean and MSE vectors $\boldsymbol{\tau}_j^k$ and $\boldsymbol{s}_j^k$ that retain information from preceding tasks through mean and MSE vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ and acquire information from the k-th task through mean and MSE vectors $\boldsymbol{\tau}_{j+1}^k$ and $\boldsymbol{s}_{j+1}^k$.

Recursions in (14)-(15) obtain the mean vector $\boldsymbol{\tau}_j^k$ by adding a correction to the mean vector of the succeeding task $\boldsymbol{\tau}_{j+1}^k$ obtained for forward and backward learning. This correction is proportional to the difference between $\boldsymbol{\tau}_j^j$ and $\boldsymbol{\tau}_{j+1}^k$ multiplied by the smoothing gain $\boldsymbol{\eta}_j^k$ that depends on the MSE vector $\boldsymbol{s}_j^j$ and the expected quadratic change $\boldsymbol{d}_{j+1}$. In particular, if $\boldsymbol{s}_j^j \gg \boldsymbol{d}_{j+1}$, the mean vector is given by that of the corresponding task for forward learning, while if $\boldsymbol{s}_j^j \ll \boldsymbol{d}_{j+1}$, the mean vector is given by that of the next task for backward learning.

The result below shows that using exact values for variances and expected quadratic changes, the previous recursions obtain the unbiased linear estimates of the mean vector $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x, y)\}$ with minimum MSE.

Theorem 2 *Let* $\boldsymbol{\sigma}_j^2$ *and* $\boldsymbol{d}_j$ *in recursions* (14)-(16) *be the actual variance and the actual expected quadratic change between tasks, that is* $\boldsymbol{\sigma}_j^2 = [\mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(1)}(x, y)\}, \mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(2)}(x, y)\}, \ldots, \mathbb{V}ar_{\mathrm{p}_j}\{\Phi^{(m)}(x, y)\}]^\top$ *and*

$\boldsymbol{d}_j = \mathbb{E}\{\boldsymbol{w}_j^2\} = \mathbb{E}\{(\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_{j-1}^\infty)^2\}$. *If the tasks' distributions satisfy* (Evo-A)*, then we have that* $\boldsymbol{\tau}_j^k$ *given by* (14) *is the unbiased linear estimator of the mean vector* $\boldsymbol{\tau}_j^\infty$ *based on* $D_1, D_2, \ldots, D_j, \ldots, D_k$ *that has the minimum MSE, and* $\boldsymbol{s}_j^k$ *given by* (15) *is its MSE.*

Proof *Since the sequence of probability distribution satisfy the* (Evo-A) *assumption, we have that the mean vectors evolve over tasks through the linear dynamical system in* (11)-(12) *where vectors* $\boldsymbol{w}_j$ *and* $\boldsymbol{v}_j$ *for* $j \in \{2, 3, \ldots, k\}$ *are independent and zero-mean. For such systems, the Rauch-Tung-Striebel recursions provide the unbiased linear estimator with minimum MSE based on samples sets* $D_1, D_2, \ldots, D_k$*. Then, equations* (8) *and* (9) *are obtained from the Rauch-Tung-Striebel recursions (see details in Appendix A).* ■

The above theorem shows that equations in (14)-(16) enable to recursively obtain the mean vector estimate for each task as well as its MSE vector leveraging all the available information.

Recursions in (8)-(9) and recursions in (14)-(15) adapt to multidimensional tasks' changes by accounting for the change of all the mean vector components. Specifically, for each j-th task and each i-th component for $i = 1, 2, \ldots, m$, the i-th component of the mean vector estimate is updated by using the corresponding component of the expected quadratic change, the estimate for the consecutive task, and the most recent sample average. Such update accounts for the specific evolution of each i-th component of the mean vector through the recursions in equations (8) and (14) and smoothing gains ($\boldsymbol{\eta}_j^j$, $\boldsymbol{\eta}_j^k$) in equations (10) and (16). In particular, updates for mean vector components with a low smoothing gain slightly change the estimate for the consecutive task (previous task in (8) and next task in (14)), while those updates for components and tasks with a high smoothing gain increase the relevance of the information of the corresponding task (sample average in (8) and forward mean vector in (14)).

5. Effective sample sizes and performance guarantees

This section analytically characterizes the ESS increase obtained by using both forward and backward learning. Then, we compare the ESS of the proposed methodology with conventional techniques that adapt to evolving tasks.

The ESS commonly quantifies the performance improvement of an algorithm in terms of the number of samples the baseline method would require to achieve the same performance. In the baseline approach of single-task learning, MRCs provide bounds for the minimax risk in terms of the smallest minimax risk as described in Mazuelas et al. (2020, 2022, 2023). The smallest minimax risk corresponds with the ideal case of knowing mean vectors exactly, that is, the minimax risk corresponding with the uncertainty set

$$\mathcal{U}_j^\infty = \{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : \mathbb{E}_\mathrm{p}\{\Phi(x, y)\} = \boldsymbol{\tau}_j^\infty\}$$

given by the expectation $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x, y)\}$. The minimum worst-case error probability over distributions in $\mathcal{U}_j^\infty$ is given by

$$R_j^\infty = \min_{\boldsymbol{\mu}} 1 - \boldsymbol{\tau}_j^{\infty\top}\boldsymbol{\mu} + \varphi(\boldsymbol{\mu}) = 1 - \boldsymbol{\tau}_j^{\infty\top}\boldsymbol{\mu}_j^\infty + \varphi(\boldsymbol{\mu}_j^\infty)$$

corresponding with the rule given by parameters $\boldsymbol{\mu}_j^\infty$. Such classification rule is referred to as optimal minimax rule because for any uncertainty set $\mathcal{U}_j^k$ given by (1) that contains the underlying distribution, we have that $\mathcal{U}_j^\infty \subseteq \mathcal{U}_j^k$ and hence $R_j^\infty \leq R(\mathcal{U}_j)$.

The baseline approach of single-task learning obtains uncertainty sets given by (1) using the mean and confidence vector provided by (6). In that case, with probability at least $1-\delta$ we have that

$$R(\mathcal{U}_j) \leq R_j^\infty + \frac{M(\kappa\sqrt{2\log(2m/\delta)} + \lambda_0)}{\sqrt{n_j}} \|\boldsymbol{\mu}_j^\infty\|_1 \tag{17}$$

where M is a bound for the feature mapping, i.e., $\|\Phi(x,y)\|_\infty \leq M$ for any $(x,y) \in \mathcal{X} \times \mathcal{Y}$. Coefficient $\kappa > 0$ is a constant that describes how the bounds are affected by the inaccuracies of the values for $\boldsymbol{\sigma}_j$. This coefficient shows the relationship between the values used for $\boldsymbol{\sigma}_j$ and the sub-Gaussian parameters of Φ_j as $subG(\Phi_j) \preceq \kappa\boldsymbol{\sigma}_j$ where Φ_j denotes the random variable given by the feature mapping of samples from the j-th task, and $subG(\mathbf{z})$ denotes the vector composed by the sub-Gaussian parameters of each component of random vector $\mathbf{z}$ (i.e., $\mathbb{E}\{e^{t(\mathbf{z}-\mathbb{E}\{\mathbf{z}\})}\} \preceq e^{subG(\mathbf{z})^2 t^2/2}$). The sub-Gaussian condition is necessary to obtain high-probability bounds and is satisfied with wide generality. In particular, in this paper we consider feature mappings that are bounded, which ensures that variable Φ_j is also bounded and hence sub-Gaussian (see e.g., Wainwright (2019)). Inequality (17) is obtained using the bounds in Mazuelas et al. (2023) together with the Chernoff bound (see e.g., equation (2.9) in Section 2.1.2 in Wainwright (2019)) for sample averages of sub-Gaussian variables.

5.1 Effective sample sizes using forward learning

The following result provides bounds for the minimax risk for each task using forward learning. Specifically, the theorem below provides bounds for the minimax risk of each j-th task with respect to the smallest minimax risk using forward mean and MSE vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ obtained as in (8) and (9), respectively.

Theorem 3 *Let $\Phi(x,y)$ be a feature mapping bounded by M, and $\mathcal{U}_j^j$ be the uncertainty set given by* (1) *using the mean and confidence vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{\lambda}_j^j = \lambda_0\sqrt{\boldsymbol{s}_j^j}$ provided by* (8) *and* (9)*. If the vectors $\boldsymbol{\sigma}_j^2$ and $\boldsymbol{d}_j$ utilized in recursions* (8)-(10) *satisfy that $subG(\Phi_j) \preceq \kappa\boldsymbol{\sigma}_j$ and $subG(\boldsymbol{w}_j) \preceq \kappa\sqrt{\boldsymbol{d}_j}$ for $\kappa > 0$ and $j = 1, 2, \ldots, k$. Then, under the evolving task assumption in* (Evo-A)*, with probability at least $1-\delta$ we have that*

$$R(\mathcal{U}_j^j) \leq R_j^\infty + \frac{M(\kappa\sqrt{2\log(2m/\delta)} + \lambda_0)}{\sqrt{n_j^j}} \left\|\boldsymbol{\mu}_j^\infty\right\|_1 \tag{18}$$

with

$$n_j^j \geq n_j + n_{j-1}^{j-1} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + \|\boldsymbol{d}_j\|_\infty n_{j-1}^{j-1}} \tag{19}$$

for $j = 2, 3, \ldots k$ and $n_1^1 = n_1$.

Proof *See Appendix B.* ■

Theorem 3 shows that the value n_j^j in (18) is the ESS of the proposed methods since the bound in (18) coincides with that of single-task learning in (17) if the sample size for the j-th task is n_j^j. In particular, single-task learning would require $n_j^j > n_j$ samples to achieve the same performance bound than forward learning using n_j samples. As shown in (19), the ESS of each task is obtained by adding a fraction of the ESS for the preceding task to the sample size. In particular, if $\boldsymbol{d}_j$ is large, the ESS is given by the sample size, while if $\boldsymbol{d}_j$ is small, the ESS is given by the sum of the sample size and the ESS of the preceding task.

The bound in (18) shows that recursions in (8) do not need to use very accurate values for $\boldsymbol{\sigma}_j$ and $\boldsymbol{d}_j$. Specifically, the coefficient κ in (18) can be taken to be small as long as the values used for $\boldsymbol{\sigma}_j$ and $\boldsymbol{d}_j$ are not much lower than the sub-Gaussian parameters of Φ_j and $\boldsymbol{w}_j$, respectively. In particular, κ is smaller than the maximum of $M/\min_{j,i}\{\sigma_j^{(i)}\}$ and $2M/\min_{j,i}\{\sqrt{d_j^{(i)}}\}$ due to the bound for the sub-Gaussian parameter of bounded random variables (see e.g., Wainwright (2019)).

The above theorem shows the ESS obtained by forward learning techniques in terms of the ESS for the preceding task. The following result allows us to directly quantify the ESS in terms of the sample size, the number of tasks, and the expected quadratic change.

Theorem 4 *Let d, $\boldsymbol{\sigma}_j$, and n be such that $d \geq \|\boldsymbol{d}_j\|_\infty$, $\|\boldsymbol{\sigma}_j^2\|_\infty \leq 1$, and $n \leq n_j$ for $j = 1, 2, \ldots, k$. Then, we have that the ESS in* (18) *satisfies*

$$n_j^j \geq n\left(1 + \left(\frac{1+\alpha}{\alpha}\right)\frac{(1+\alpha)^{2j-2}-1}{(1+\alpha)^{2j-1}+1}\right) \quad \textit{with} \quad \alpha = \frac{2}{\sqrt{1+\frac{4}{nd}}-1}. \tag{20}$$

In particular, for $j > 1$, we have that

$$\begin{aligned} n_j^j &\geq n\left(1 + \frac{j-1}{3}\right) && \textit{if} && nd < \frac{1}{j^2} \\ n_j^j &\geq n\left(1 + \frac{1}{5\sqrt{nd}}\right) && \textit{if} && \frac{1}{j^2} \leq nd < 1 \\ n_j^j &\geq n\left(1 + \frac{1}{3nd}\right) && \textit{if} && nd \geq 1. \end{aligned}$$

Proof *See Appendix C.* ■

The above theorem characterizes the increase in the ESS using forward learning in comparison with single-task learning ($\alpha > 0$ and $(1+\alpha)^{2j-2} > 1$ for $j > 1$). For $j = 1$, the ESS equals the sample size n because forward learning coincides with single-task learning. For $j > 1$, the ESS is significantly larger using forward learning in comparison with single-task learning for reduced samples sizes n and small tasks' expected quadratic change d (α significantly larger than 0). In addition, the ESS in (20) grows monotonically with the number of tasks j and becomes proportional to j when the expected quadratic change is smaller than $1/(j^2 n)$. Figure 4 further illustrates the increase in ESS with respect to the sample size due to forward learning in terms of the number of tasks, the sample size, and the expected quadratic change between consecutive tasks. This figure describes the increase

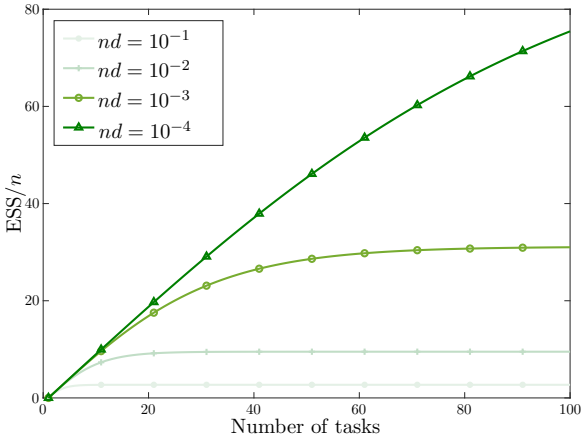


Figure 4: The ESS provided by forward learning significantly increases with the number of tasks especially for small values for the sample size n and the expected quadratic change d.

in ESS with respect to the sample size for $nd = 10^{-4}$, 10^{-3}, 10^{-2}, and 10^{-1}, and a varying number of tasks. In particular, the figure shows that the ESS of the proposed methodology significantly increases with the number of tasks especially for small sample sizes.

Existing methods for learning a sequence of tasks provide comparable performance bounds that decrease with the number of tasks j and the sample size n (Denevi et al., 2019; Pentina and Lampert, 2015; Denevi et al., 2019; Khodak et al., 2019; Balcan et al., 2019). For instance, in scenarios where tasks follow the (i.i.d.-A) assumption the bounds depend on n and j as $\mathcal{O}(\frac{1}{\sqrt{j}}) + \mathcal{O}(\frac{D_{\text{Var}}}{\sqrt{n}})$ in Denevi et al. (2019), where D_{Var} is given by the "relatedness among tasks" and quantifies the tasks variability. In scenarios where tasks are evolving, the bounds depend on n an j as $\mathcal{O}(\frac{1}{\sqrt{j}}) + \mathcal{O}(\frac{D_{KL}}{\sqrt{n}})$ in Pentina and Lampert (2015), where D_{KL} is the KL-divergence describing the difference between consecutive tasks. As shown in Theorems 3 and 4, the presented bounds depend on n and j as $\mathcal{O}(\frac{1}{\sqrt{\text{ESS}(n,d,j)}})$, where d is the expected quadratic change between consecutive tasks. Similarly to previous works, our bounds decrease when n and j increase and increase when d increases, as shown in (20) and (22). In particular, if $nd < \frac{1}{j^2}$, Theorem 4 shows that the proposed methodology provides bounds that depend on n and j as $\mathcal{O}(\frac{1}{\sqrt{nj}})$. Differently from other techniques, the proposed methodology provides bounds that decrease to zero increasing the number of samples for any number of tasks j. In particular, as shown in Theorem 4, we have that $\text{ESS}(n, d, j) = \Omega(n)$ for any d and j. The next section, shows how the usage of backward learning results in even larger ESSs.

5.2 Effective sample sizes using forward and backward learning

The following result provides bounds for the minimax risk for each task using forward and backward learning

Theorem 5 *Let n_j^j and κ be as in Theorem 3 for any $j \in \{1, 2, ..., k\}$. Then, under the evolving task assumption in* (Evo-A)*, with probability at least $1 - \delta$ we have that*

$$R(\mathcal{U}_j^k) \leq R_j^\infty + \frac{M(\kappa\sqrt{2\log(2m/\delta)} + \lambda_0)}{\sqrt{n_j^k}} \left\|\boldsymbol{\mu}_j^\infty\right\|_1 \tag{21}$$

with

$$n_j^k \geq n_j^j + n_{j+1}^{-k} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + n_{j+1}^{-k}\|\boldsymbol{d}_{j+1}\|_\infty}$$

for $j < k$, where n_{j+1}^{-k} satisfies $n_k^{-k} = n_k$ and $n_j^{-k} \geq n_j + n_{j+1}^{-k} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + n_{j+1}^{-k}\|\boldsymbol{d}_{j+1}\|_\infty}$.

Proof *See Appendix D.* ■

Theorem 5 shows that the value n_j^k in (21) is the ESS of the proposed methods since the bound in (21) coincides with that of single-task learning in (17) if the sample size for the j-th task is n_j^k. That is, n_j^k is the ESS of the proposed methodology using forward and backward learning since single-task learning would require $n_j^k > n_j$ samples to achieve the same performance than forward and backward learning using n_j samples. In addition, Theorem 5 shows that the methods proposed can increase the ESS of each task by leveraging information from all the tasks in the sequence. In particular, the bounds provided by inequality (21) are lower than those in Theorem 3. The ESS of each j-th task is obtained by adding a fraction of the ESS for the next task to the ESS of the corresponding task leveraging information from preceding tasks. In particular, if $\boldsymbol{d}_j$ is large, the ESS is given by that leveraging information from j tasks, while if $\boldsymbol{d}_j$ is small, the ESS is given by the sum of the ESS leveraging information from j tasks and the ESS of the next task.

The above theorem shows the increase in ESS in terms of the ESS for consecutive tasks. The following result allows to directly quantify the ESS in terms of the sample size and the expected quadratic change.

Theorem 6 *Let d, $\boldsymbol{\sigma}_j$, and n be such that $d \geq \|\boldsymbol{d}_j\|_\infty$, $\|\boldsymbol{\sigma}_j^2\|_\infty \leq 1$, and $n \leq n_j$ for $j = 1, 2, \ldots, k$. For any $j \in \{1, 2, \ldots, k\}$, we have that the ESS in* (21) *satisfies*

$$n_j^k \geq n\left(1 + \left(\frac{1+\alpha}{\alpha}\right)\frac{(1+\alpha)^{2j-2} - 1}{(1+\alpha)^{2j-1} + 1} + \left(\frac{1+\alpha}{\alpha}\right)\frac{(1+\alpha)^{2(k-j)} - 1}{(1+\alpha)^{2(k-j)+1} + 1}\right) \tag{22}$$

with $\alpha = \frac{2}{\sqrt{1+\frac{4}{nd}}-1}$. In particular, for $j > 1$, we have that

$$\begin{aligned}
n_j^k &\geq n_j^j + n\frac{j(k-j)}{j+2(k-j)} \geq n\left(1 + \frac{j-1}{3} + \frac{j(k-j)}{j+2(k-j)}\right) && \text{if} && nd < \frac{1}{j^2} \\
n_j^k &\geq n_j^j + \frac{1}{5}\sqrt{\frac{n}{d}} && \geq n\left(1 + \frac{2}{5\sqrt{nd}}\right) && \text{if} && \frac{1}{j^2} \leq nd < 1 \\
n_j^k &\geq n_j^j + \frac{1}{3d} && \geq n\left(1 + \frac{2}{3nd}\right) && \text{if} && nd \geq 1
\end{aligned}$$

where n_j^j satisfies (20)*.*

Proof *See Appendix E.* ■

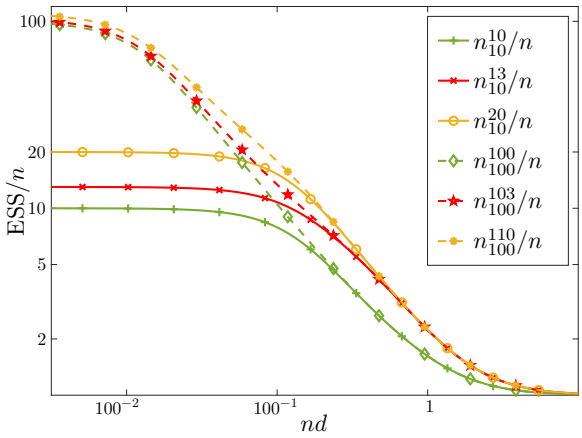


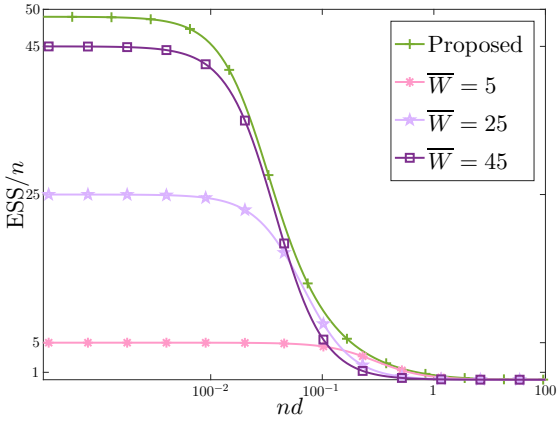
Figure 5: The ESS provided by forward and backward learning (red and orange lines) significantly increases in comparison with forward learning (green line) especially for small values for the sample size n and the expected quadratic change d.

The above theorem characterizes the increase in the ESS using forward and backward learning in comparison with single-task learning and forward learning. For $j = k$, the ESS equals the ESS using forward learning in Theorem 4 because forward and backward learning coincides with forward learning. For $j < k$, the ESS is significantly higher for reduced sample sizes and small tasks' expected quadratic change. In addition, the ESS in (22) grows monotonically with the number of preceding tasks (j) and with the number of succeeding tasks ($k - j$). It becomes proportional to the total number of tasks k when the expected quadratic change is smaller than $1/(j^2 n)$ and $j \geq k/2$. Figure 5 further illustrates the increase in ESS due to forward and backward learning in comparison with forward learning. As the figure shows, the increase in the ESS can be classified into three regimes depending on the sample size n and the expected quadratic change d, as described in Theorems 4 and 6. The ESS is only marginally larger than the sample size for sizeble values of nd (large samples sizes or drastic changes in the distribution); the ESS quickly increases when nd becomes small (reduced sample sizes and moderate changes in the distribution); and the ESS becomes proportional to the total number of tasks k if nd is rather small (very small sample sizes and very slow changes in the distribution).

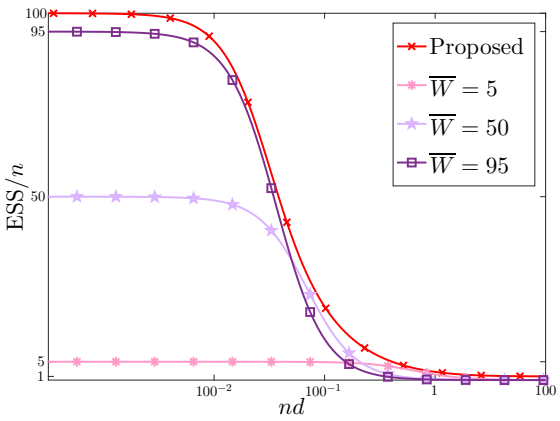
The next section numerically shows the increase in the ESS due to forward and backward learning in comparison with conventional techniques designed for evolving tasks.

5.3 Numerically assessment of the ESSs

This section compares the ESS of the proposed methodology with the baseline approach that utilizes sliding windows. One difference between the sliding window approach and the proposed methodology is that sliding windows abruptly discard older data and assign the same relevance to all the tasks in the window. On the other hand, smoothing strategies, such as the one used in this paper, increase the relevance of the most recent tasks. These strategies have been shown to allow for a more gradual adaptation to temporal changes in



(a) ESS of the $j = 50$-th task leveraging information from preceding tasks.



(b) For a sequence of $k = 100$ tasks, ESS of the $j = 50$-th task leveraging information from preceding and succeeding tasks.

Figure 6: The ESS provided by the proposed methodology is grater than the ESS of the baseline approach that utilizes sliding windows for multiple windows sizes.

the data (Muth, 1960). In addition, the methods proposed can provide a multidimensional adaptation to changes in the sequence of tasks. For instance, the smoothing gains $\boldsymbol{\eta}_j^j$ and $\boldsymbol{\eta}_j^k$ in (10) and (16) are vectors that allow us to estimate each component of the mean vectors in a different manner.

Figure 6 shows the ESS increase obtained by the proposed methodology in comparison with the baseline approach that utilizes sliding windows. Techniques based on sliding windows are commonly used to adapt to evolving tasks (Zhang et al., 2016; Tahmasbi et al., 2021; Mazzetto and Upfal, 2023). Such techniques obtain classification rules for each task using the sample sets corresponding with the $\overline{W}$ closest tasks. Large values of window size adapt to gradual changes in the tasks' distributions, while small values of window size adapt to abrupt changes in the tasks' distributions. Appendix F shows the expressions corresponding to the above ESSs obtained using sliding windows.

Figure 6a illustrates the increase in the ESS due to forward learning in comparison with techniques based on sliding windows of preceding tasks, and Figure 6b illustrates the increase in ESS due to forward and backward learning in comparison with techniques based on sliding windows of preceding and succeeding tasks. Specifically, Figure 6a and Figure 6b show the ESS of the proposed methods and sliding windows with $\overline{W} = 5, 25$, and 45 leveraging information from preceding tasks, and $\overline{W} = 5, 50$, and 95 leveraging information from the same number of preceding and succeeding tasks. Small values of window size yield to low ESS when nd decreases, while large values of window size yield to low ESS when nd increases. The ESS provided by the proposed methodology is greater than that provided by techniques based on sliding windows for different number of tasks, sample sizes, and changes between consecutive tasks.

6. Implementation in multiple scenarios with evolving tasks

This section describes the implementation and the computational and memory complexity of the proposed methodology applied to batch learning scenarios and online learning scenarios.

Multiple supervised learning paradigms obtain classification rules leveraging information only from preceding tasks. For instance, MDA methods often obtain classification rules without samples from the target task (Stojanov et al., 2020; Li et al., 2022), while SCD methods obtain classification rules before receiving samples from the target task (Elwell and Polikar, 2011; Brzezinski and Stefanowski, 2013). In these cases, we obtain mean and MSE vectors $\boldsymbol{\tau}_j^{j-1}$ and $\boldsymbol{s}_j^{j-1}$ for each j-th task using information from sample sets $D_1, D_2, \ldots, D_{j-1}$. Then, the sample set corresponding with each j-th task arrives after obtaining mean and MSE vectors for the j-th task. Mean and MSE vectors $\boldsymbol{\tau}_j^{j-1}$ and $\boldsymbol{s}_j^{j-1}$ can be obtained from forward mean and MSE vectors $\boldsymbol{\tau}_{j-1}^{j-1}$ and $\boldsymbol{s}_{j-1}^{j-1}$ given by (8) and (9) as

$$\boldsymbol{\tau}_j^{j-1} = \boldsymbol{\tau}_{j-1}^{j-1}, \quad \boldsymbol{s}_j^{j-1} = \boldsymbol{s}_{j-1}^{j-1} + \boldsymbol{d}_j. \tag{23}$$

The proposed methodology obtains at each step the classifier parameters and minimax risks for each task using an accelerated subgradient method (ASM) that solves the convex optimization problem (5) using Nesterov extrapolation strategy (Nesterov and Shikhman, 2015; Tao et al., 2019). In addition, we propose efficient algorithms that use a warm-start for the ASM iterations.

The proposed techniques efficiently obtain classifier parameter $\boldsymbol{\mu}$ and minimax risk $R(\mathcal{U})$ for each task from updated mean vector estimate $\boldsymbol{\tau}$ and confidence vector $\boldsymbol{\lambda}$. The ASM algorithm applied to optimization (5) obtains classifier parameters using the iterations for $l = 1, 2, \ldots, K$

$$\begin{aligned} \bar{\boldsymbol{\mu}}^{l+1)} &= \boldsymbol{\mu}^{l)} + a_l \left(\boldsymbol{\tau} - \partial\varphi(\boldsymbol{\mu}^{l)}) - \boldsymbol{\lambda}\text{sign}(\boldsymbol{\mu}^{l)}) \right) \\ \boldsymbol{\mu}^{l+1)} &= \bar{\boldsymbol{\mu}}^{l+1)} + \theta_{l+1}(\theta_l^{-1} - 1) \left(\boldsymbol{\mu}^{l)} - \bar{\boldsymbol{\mu}}^{l)} \right) \end{aligned} \tag{24}$$

where $\boldsymbol{\mu}^{l)}$ is the l-th iterate for $\boldsymbol{\mu}$, $\theta_l = 2/(l+1)$ and $a_l = 1/(l+1)^{3/2}$ are step sizes and $\partial\varphi(\boldsymbol{\mu}^{l)})$ denotes a subgradient of $\varphi(\cdot)$ at $\boldsymbol{\mu}^{l)}$.

The proposed algorithm reduces the number of ASM iterations by using a warm-start. Specifically, for each task, the proposed algorithm initializes the parameters in (24) with the solution obtained for the closest task (see Alg. 5 in Appendix G).

6.1 Batch learning scenarios

In MDA information from preceding tasks can be used to improve the performance of the last task. Specifically, MDA methods use k sample sets corresponding with different source tasks to obtain a classification rule for the k-th target task with $|D_k| > 0$ (Reeve et al., 2021; Tripuraneni et al., 2020) or $|D_k| = 0$ (Stojanov et al., 2020; Li et al., 2022). Algorithm 1 details the implementation of the proposed methodology applied to MDA that first obtains forward mean and MSE vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ as in (8)-(9) for $j = 1, 2, \ldots, k-1$. Then, we obtain mean and MSE vectors $\boldsymbol{\tau}_k^k$ and $\boldsymbol{s}_k^k$ as in (8)-(9) if $|D_k| > 0$ and mean and MSE vectors $\boldsymbol{\tau}_k^{k-1}$ and $\boldsymbol{s}_k^{k-1}$ as in (23) if $|D_k| = 0$. We take the confidence vector as in (6)

Algorithm 1 MDA

Input: $D_1, D_2, \ldots, D_k$
Output: $\boldsymbol{\mu}_k^k, R(\mathcal{U}_k^k)$ if $|D_k| > 0$ and $\boldsymbol{\mu}_k^{k-1}, R(\mathcal{U}_k^{k-1})$ if $|D_k| = 0$.
for $j = 1, 2, ..., k-1$ **do**
  Obtain forward mean and MSE vectors $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j$ as in (8)-(9)
if $|D_k| > 0$ **then**
  Obtain mean and MSE vectors $\boldsymbol{\tau}_k^k, \boldsymbol{s}_k^k$ as in (8)-(9)
  Take confidence vector $\boldsymbol{\lambda}_k^k$ as $\lambda_0\sqrt{\boldsymbol{s}_k^k}$
  Obtain classifier parameter and minimax risk $\boldsymbol{\mu}_k^k$ and $R(\mathcal{U}_k^k)$ solving (5)
else
  Obtain mean and MSE vectors $\boldsymbol{\tau}_k^{k-1}, \boldsymbol{s}_k^{k-1}$ as in (23)
  Take confidence vector $\boldsymbol{\lambda}_k^{k-1}$ as $\lambda_0\sqrt{\boldsymbol{s}_k^{k-1}}$
  Obtain classifier parameter and minimax risk $\boldsymbol{\mu}_k^{k-1}$ and $R(\mathcal{U}_k^{k-1})$ solving (5)

and obtain the classifier parameter and the minimax risk for the k-th task solving (5) (see Alg. 5 in Appendix G). Algorithm 1 has computational complexity $O(km + n2^{|\mathcal{Y}|}Km)$ and memory complexity $O(m)$ where m is the length of the feature mapping, n is the sample size, and K is the number of iterations of the optimization step.

In MTL, at each time step, information from preceding tasks can be used to improve the performance of the last task and, reciprocally, the information from the last task can be used to improve the performance of the preceding tasks. Specifically, MTL methods use k sample sets corresponding with different tasks to obtain classification rules $\mathrm{h}_1^k, \mathrm{h}_2^k, \ldots, \mathrm{h}_k^k$ for the k tasks (Zhang and Yang, 2018; Lin et al., 2020). Algorithm 2 details the implementation of the proposed methodology applied to MTL that first obtains forward mean and MSE vectors $\boldsymbol{\tau}_1^1, \boldsymbol{\tau}_2^2, \ldots, \boldsymbol{\tau}_k^k$ and $\boldsymbol{s}_1^1, \boldsymbol{s}_2^2, \ldots, \boldsymbol{s}_k^k$ as in (8)-(9). Then, we obtain forward and backward mean and MSE vectors $\boldsymbol{\tau}_1^k, \boldsymbol{\tau}_2^k, \ldots, \boldsymbol{\tau}_k^k$ and $\boldsymbol{s}_1^k, \boldsymbol{s}_2^k, \ldots, \boldsymbol{s}_k^k$ as in (14)-(15), take the confidence vectors $\boldsymbol{\lambda}_1^k, \boldsymbol{\lambda}_2^k, \ldots, \boldsymbol{\lambda}_k^k$ as in (6), and obtain the classifier parameters $\boldsymbol{\mu}_1^k, \boldsymbol{\mu}_2^k, \ldots, \boldsymbol{\mu}_k^k$ and the minimax risks $R(\mathcal{U}_1^k), R(\mathcal{U}_2^k), \ldots, R(\mathcal{U}_k^k)$ for the k tasks in the sequence solving (5) (see Alg. 5 in Appendix G). Algorithm 2 has computational complexity $\mathcal{O}(mk + n2^{|\mathcal{Y}|}Kmk)$ and memory complexity $\mathcal{O}(mk + k)$.

Algorithm 2 MTL

Input: $D_1, D_2, \ldots, D_k$
Output: $\{\boldsymbol{\mu}_j^k\}_{1\leq j\leq k}$ and $\{R(\mathcal{U}_j^k)\}_{1\leq j\leq k}$
for $j = 1, 2, ..., k$ **do**
  Obtain forward mean and MSE vectors $\boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j$ as in (8)-(9)
for $j = k-1, k-2, ..., 1$ **do**
  Obtain forward and backward mean and MSE vectors $\boldsymbol{\tau}_j^k, \boldsymbol{s}_j^k$ as in (14)-(15)
  Take confidence vector $\boldsymbol{\lambda}_j^k$ as $\lambda_0\sqrt{\boldsymbol{s}_j^k}$
  Obtain classifier parameter and minimax risk $\boldsymbol{\mu}_j^k$ and $R(\mathcal{U}_j^k)$ solving (5)

6.2 Online learning scenarios

In SCD, sample sets corresponding with different tasks arrive over time and at each time step, information from preceding tasks can be used to improve the performance of the last task. SCD methods use for each k-th task the most recent sample set D_{k-1} and information retained from preceding tasks to obtain the classification rule h_k^{k-1} (Elwell and Polikar, 2011; Brzezinski and Stefanowski, 2013). Algorithm 3 details the implementation of the proposed methodology applied to SCD that first obtains forward mean and MSE vectors $\boldsymbol{\tau}_{k-1}^{k-1}$ and $\boldsymbol{s}_{k-1}^{k-1}$ as in (8)-(9) from those for the preceding task $\boldsymbol{\tau}_{k-2}^{k-2}$ and $\boldsymbol{s}_{k-2}^{k-2}$ together with sample set D_{k-1}. Then, we obtain mean and MSE vectors $\boldsymbol{\tau}_k^{k-1}$ and $\boldsymbol{s}_k^{k-1}$ as in (23). We take the confidence vector $\boldsymbol{\lambda}_k^{k-1}$ as in (6) and obtain the classifier parameter $\boldsymbol{\mu}_k^{k-1}$ and the minimax risk $R(\mathcal{U}_k^{k-1})$ for the k-th task solving (5) (see Alg. 5 in Appendix G). Algorithm 3 has at each step k computational complexity $O(m + n2^{|\mathcal{Y}|}Km)$ and memory complexity $O(m)$.

Algorithm 3 SCD at step k

Input: D_{k-1}, $\boldsymbol{\tau}_{k-2}^{k-2}, \boldsymbol{s}_{k-2}^{k-2}$, and $\boldsymbol{\mu}_{k-1}^{k-2}$
Output: $\boldsymbol{\tau}_{k-1}^{k-1}, \boldsymbol{s}_{k-1}^{k-1}, \boldsymbol{\mu}_k^{k-1}$, and $R(\mathcal{U}_k^{k-1})$
Obtain forward mean and MSE vectors $\boldsymbol{\tau}_{k-1}^{k-1}, \boldsymbol{s}_{k-1}^{k-1}$ as in (8)-(9)
Obtain mean and MSE vectors $\boldsymbol{\tau}_k^{k-1}, \boldsymbol{s}_k^{k-1}$ as in (23)
Take confidence vector $\boldsymbol{\lambda}_k^{k-1}$ as $\lambda_0\sqrt{\boldsymbol{s}_k^{k-1}}$
Obtain classifier parameter and minimax risk $\boldsymbol{\mu}_k^{k-1}$ and $R(\mathcal{U}_k^{k-1})$ solving (5)

In CL, sample sets corresponding with different tasks arrive over time and at each time step, information from preceding tasks can be used to improve the performance of the last task and, reciprocally, the information from the last task can be used to improve the performance of the preceding tasks. CL methods use at each step k the sample set D_k and information retained from preceding tasks to obtain classification rules $\mathrm{h}_1^k, \mathrm{h}_2^k, \ldots, \mathrm{h}_k^k$ for the sequence of tasks (Henning et al., 2021; Hurtado et al., 2021). Algorithm 4 details the implementation of the proposed methodology applied CL that first obtains at each step k forward mean and MSE vectors $\boldsymbol{\tau}_k^k$ and $\boldsymbol{s}_k^k$ as in (8)-(9). Then, we obtain forward and backward mean and MSE vectors $\boldsymbol{\tau}_{k-b}^k, \boldsymbol{\tau}_{k-b+1}^k, \ldots, \boldsymbol{\tau}_{k-1}^k$ and $\boldsymbol{s}_{k-b}^k, \boldsymbol{s}_{k-b+1}^k, \ldots, \boldsymbol{s}_{k-1}^k$ as in (14)-(15) for $b = k - j$ backward steps. In particular, if $b = 0$, the proposed methodology carries out only forward learning. Then, we take confidence vectors $\boldsymbol{\lambda}_{k-b}^k, \boldsymbol{\lambda}_{k-b+1}^k, \ldots, \boldsymbol{\lambda}_k^k$ as in (6) and obtain classifier parameters $\boldsymbol{\mu}_{k-b}^k, \boldsymbol{\mu}_{k-b+1}^k, \ldots, \boldsymbol{\mu}_k^k$ and minimax risks $R(\mathcal{U}_{k-b}^k), R(\mathcal{U}_{k-b+1}^k), \ldots, R(\mathcal{U}_k^k)$ solving (5) (see Alg. 5 in Appendix G). Algorithm 4 has at each step k computational complexity $\mathcal{O}((b+1)mk + bn2^{|\mathcal{Y}|}Km)$ and memory complexity $\mathcal{O}((b+k)m + n2^{|\mathcal{Y}|}mb)$. The number of backward steps $b = k - j$ can be taken to be rather small since the benefits of learning from succeeding tasks are achieved using only $b = 3$ backward steps in most of the situations (see Fig. 6 in Section 5). The proposed techniques can be extended to situations in which a new sample set can correspond with a precedingly learned task by updating the sample average and the MSE vector in (6) with the new sample set (see Appendix H).

Algorithm 4 CL at step k

Input: $D_k, \boldsymbol{\tau}_j^j, \boldsymbol{s}_j^j, \boldsymbol{\mu}_j^j$ for $k-b \leq j < k$
Output: $\boldsymbol{\mu}_j^k$ for $k-b+1 \leq j \leq k$, $\boldsymbol{\tau}_k^k, \boldsymbol{s}_k^k$
Obtain forward mean and MSE vectors $\boldsymbol{\tau}_k^k, \boldsymbol{s}_k^k$ using (8)-(9)
Take confidence vector $\boldsymbol{\lambda}_k^k = \lambda_0 \sqrt{\boldsymbol{s}_k^k}$
Obtain classifier parameter $\boldsymbol{\mu}_k^k$ and minimax risk $R(\mathcal{U}_k^k)$ solving (5)
for $j = k-1, k-2, \ldots, k-b$ **do**
  Obtain forward and backward mean and MSE vectors $\boldsymbol{\tau}_j^k, \boldsymbol{s}_j^k$ using (14)-(15)
  Take confidence vector $\boldsymbol{\lambda}_j^k$ as $\lambda_0 \sqrt{\boldsymbol{s}_j^k}$
  Obtain classifier parameter $\boldsymbol{\mu}_j^k$ and minimax risk $R(\mathcal{U}_j^k)$ solving (5)

7. Extension to higher-order dependences

This section describes algorithms that account for higher-order dependences among tasks. Such algorithms utilize methods that are commonly used for target tracking to describe target trajectories using kinematic models (see e.g., Bar-Shalom et al. (2004)).

For each j-th task, each component of the mean vector $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x,y)\} \in \mathbb{R}^m$ is estimated leveraging information from k tasks and accounting for high-order dependences between consecutive tasks. Let $\tau_j^{\infty(i)}$ denote the i-th component of the mean vector for $i = 1, 2, \ldots, m$. We assume that $\tau_j^{\infty(i)}$ is p times differentiable with respect to time and denote by $\boldsymbol{\gamma}_{j,i}^\infty \in \mathbb{R}^{p+1}$ the vector composed by the i-th component of the mean vector $\tau_j^{\infty(i)}$ and its successive derivatives up to order p. If $p = 0$, the state vector $\gamma_{j,i}^\infty$ is the mean vector component $\tau_j^{\infty(i)}$ and the recursions below coincide with the recursions in Section 4.

As is usually done for target tracking Bar-Shalom et al. (2004), we model the evolution of the state vector $\boldsymbol{\gamma}_{j,i}^\infty$ using the partially-observed linear dynamical system

$$\begin{aligned} \boldsymbol{\gamma}_{j,i}^\infty &= \mathbf{T}_j \boldsymbol{\gamma}_{j-1,i}^\infty + \overline{\boldsymbol{w}}_{j,i} \\ \tau_j^{(i)} &= \tau_j^{\infty(i)} + v_{j,i} \end{aligned} \tag{25}$$

with transition matrix $\mathbf{T}_j = \mathbf{I} + \sum_{s=1}^p \Delta_j^s \mathbf{U}_s / s!$ where $\mathbf{U}_s$ is the $(p+1)\times(p+1)$ matrix with ones on the s-th upper diagonal and zeros in the rest of components, Δ_j is the time increment between the j-th and the $(j-1)$-th tasks, $i = 1, 2, \ldots, m$, and $j = 1, 2, \ldots$. The variables $\overline{\boldsymbol{w}}_{j,i}$ and $v_{j,i}$ represent uncorrelated zero-mean noise processes with variance $\mathbf{D}_{j,i} = \boldsymbol{g}_j \boldsymbol{g}_j^\top \bar{d}_j^{(i)}$ and $s_j^{(i)} = \sigma_j^{2(i)}/n_j$, respectively, where $\boldsymbol{g}_j$ is given by $\boldsymbol{g}_j = [\Delta_j^{p+1}/(p+1)!, \Delta_j^p/p!, \ldots, \Delta_j]^\top$. Variances $\mathbf{D}_{j,i}$ and $s_j^{(i)}$ can be estimated online using methods such as those proposed in Odelson et al. (2006); Akhlaghi et al. (2017). Dynamical systems as that given by (25) are known in target tracking as kinematic state models and can be derived using the p-th order Taylor expansion of $\tau_j^{\infty(i)}$.

Analogously to Section 4, for a sequence of k tasks, the proposed techniques first obtain for each j-th task with $j = \{1, 2, \dots, k\}$ forward state vector and MSE matrix $\boldsymbol{\gamma}_{j,i}^{j}, \boldsymbol{\Sigma}_{j,i}^{j}$ as

$$\boldsymbol{\gamma}_{j,i}^{j} = \mathbf{T}_j \boldsymbol{\gamma}_{j-1,i}^{j-1} + \boldsymbol{\eta}_{j,i}^{j} \left(\tau_j^{(i)} - \mathbf{e}_1^\top \mathbf{T}_j \boldsymbol{\gamma}_{j-1,i}^{j-1} \right) \tag{26}$$

$$\boldsymbol{\Sigma}_{j,i}^{j} = \left(\mathbf{I} - \boldsymbol{\eta}_{j,i}^{j} \mathbf{e}_1^\top \right) \left(\mathbf{T}_j \boldsymbol{\Sigma}_{j-1,i}^{j-1} \mathbf{T}_j^\top + \mathbf{D}_{j,i} \right) \tag{27}$$

$$\boldsymbol{\eta}_{j,i}^{j} = \frac{\left(\mathbf{T}_j \boldsymbol{\Sigma}_{j-1,i}^{j-1} \mathbf{T}_j^\top + \mathbf{D}_{j,i} \right) \mathbf{e}_1}{\mathbf{e}_1^\top \left(\mathbf{T}_j \boldsymbol{\Sigma}_{j-1,i}^{j-1} \mathbf{T}_j^\top + \mathbf{D}_{j,i} \right) \mathbf{e}_1 + s_j^{(i)}} \tag{28}$$

for $j = 2, 3, \dots, k$ and $i = 1, 2, \dots, m$ with $\boldsymbol{\tau}_j, \boldsymbol{s}_j$ given by (6), $\boldsymbol{\gamma}_{1,i}^{1} = \tau_1^{(i)} \mathbf{e}_1$, and $\boldsymbol{\Sigma}_{1,i}^{1} = s_1^{(i)} \mathbf{e}_1 \mathbf{e}_1^\top$. State vector and MSE matrix $\boldsymbol{\gamma}_{j,i}^{j}, \boldsymbol{\Sigma}_{j,i}^{j}$ are obtained leveraging information up to the j-th task from sample sets $D_1, D_2, \dots, D_j$. Specifically, for each j-th task, those vectors and matrices are obtained by acquiring information from sample set D_j through $\tau_j^{(i)}, s_j^{(i)}$ and retaining information from sample sets $D_1, D_2, \dots, D_{j-1}$ through state vector and MSE matrix $\boldsymbol{\gamma}_{j-1,i}^{j-1}, \boldsymbol{\Sigma}_{j-1,i}^{j-1}$.

The proposed techniques obtain for each j-th task with $j = \{1, 2, \dots, k\}$ forward and backward state vector and MSE matrix leveraging information from the k tasks in the sequence. Let $\boldsymbol{\gamma}_{j,i}^{k}$ and $\boldsymbol{\Sigma}_{j,i}^{k}$ denote the state vector and MSE matrix for forward and backward learning corresponding to the j-th task for $j = \{1, 2, \dots, k\}$ and the i-th component. The following recursions allow us to obtain $\boldsymbol{\gamma}_{j,i}^{k}, \boldsymbol{\Sigma}_{j,i}^{k}$ using forward state vector and MSE matrix $\boldsymbol{\gamma}_{j,i}^{j}, \boldsymbol{\Sigma}_{j,i}^{j}$ as

$$\boldsymbol{\gamma}_{j,i}^{k} = \boldsymbol{\gamma}_{j,i}^{j} + \mathbf{H}_{j,i}^{k} \left(\boldsymbol{\gamma}_{j+1,i}^{k} - \boldsymbol{\gamma}_{j,i}^{j} \right) \tag{29}$$

$$\boldsymbol{\Sigma}_{j,i}^{k} = \boldsymbol{\Sigma}_{j,i}^{j} + \mathbf{H}_{j,i}^{k} (\boldsymbol{\Sigma}_{j+1,i}^{k} - \mathbf{T}_{j+1} \boldsymbol{\Sigma}_{j,i}^{j} \mathbf{T}_{j+1}^\top - \mathbf{D}_{j+1,i}) {\mathbf{H}_{j,i}^{k}}^\top \tag{30}$$

$$\mathbf{H}_{j,i}^{k} = \boldsymbol{\Sigma}_{j,i}^{j} \mathbf{T}_j^\top \left(\mathbf{T}_{j+1} \boldsymbol{\Sigma}_{j,i}^{j} \mathbf{T}_{j+1}^\top + \mathbf{D}_{j+1,i} \right)^{-1} \tag{31}$$

for $j = 1, 2, \dots k-1$ and $i = 1, 2, \dots, m$ with $\boldsymbol{\gamma}_{j,i}^{j}$, $\boldsymbol{\Sigma}_{j,i}^{j}$ given by (26)-(27). State vectors and MSE matrices $\boldsymbol{\gamma}_{j,i}^{k}, \boldsymbol{\Sigma}_{j,i}^{k}$ are obtained leveraging information from sample sets $D_1, D_2, \dots, D_k$. Specifically, for each j-th task and $i = 1, 2, \dots, m$, state vectors and MSE matrices are obtained acquiring information from sample sets $D_{j+1}, D_{j+2}, \dots, D_k$ through state vector and MSE matrix $\boldsymbol{\gamma}_{j+1,i}^{k}, \boldsymbol{\Sigma}_{j+1,i}^{k}$ and retaining information from sample sets $D_1, D_2, \dots, D_j$ through state vector and MSE matrix $\boldsymbol{\gamma}_{j,i}^{j}, \boldsymbol{\Sigma}_{j,i}^{j}$.

The above recursions allow us to obtain state vectors as well as the MSE matrices for each task accounting for high-order dependences among tasks, accounting for multidimensional tasks' changes, and leveraging information from all tasks in the sequence. Such recursions adapt to multidimensional tasks' changes by accounting for the change between consecutive mean vector components. Specifically, for each j-th task and each i-th component with $i = 1, 2, \dots, m$, the i-th component of the mean vector estimate is updated by using the corresponding component of the expected quadratic change, the estimate for the consecutive task, and the most recent sample average.

The result below shows that the state vectors obtained above are the estimates of the state vectors with minimum MSE.

Theorem 7 *Let* $\sigma_j^{2(i)}$ *and* $\bar{d_j}^{(i)}$ *in recursions* (26)-(31) *be* $\sigma_j^{2(i)} = \mathbb{V}ar_{\mathrm{p}_j}\{\Phi(x,y)^{(i)}\}$ *and* $\bar{d_j}^{(i)} = \tau_j^{\infty(p+1)}$. *If the tasks' distributions satisfy the dynamical system in* (25), *then we have that*

1. $\boldsymbol{\gamma}_{j,i}^{j}$ *given by* (26) *is the unbiased linear estimator of the state vector* $\boldsymbol{\gamma}_{j,i}^{\infty}$ *based on* $D_1, D_2, \ldots, D_j$ *that has the minimum MSE and* $\boldsymbol{\Sigma}_{j,i}^{j}$ *given by* (27) *is its MSE.*

2. $\boldsymbol{\gamma}_{j,i}^{k}$ *given by* (29) *is the unbiased linear estimator of the state vector* $\boldsymbol{\gamma}_{j,i}^{\infty}$ *based on* $D_1, D_2, \ldots, D_k$ *that has the minimum MSE and* $\boldsymbol{\Sigma}_{j,i}^{k}$ *given by* (30) *is its MSE.*

Proof *See Appendix A.* ■

The above theorem shows that equations in (26)-(31) enable to recursively obtain, for each j-th task, the state vector estimate as well as its MSE matrix leveraging all the information at each step.

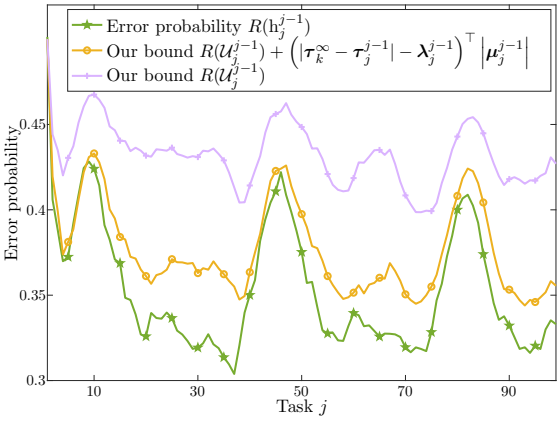
The proposed methodology extended to higher-order dependences can be applied to the supervised learning scenarios described in Sections 6.1 and 6.2 above. The application of the extension for high-order time dependences in these scenarios is similar to the implementation described in Algorithms 1-4 by changing the estimate of mean and MSE vectors to the estimate of state vector and MSE matrix.

8. Numerical results

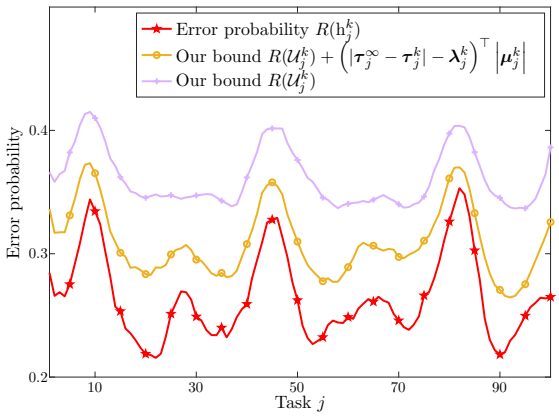
This section evaluates the performance of the proposed methods in comparison with the presented performance guarantees and the state-of-the-art. We show the reliability of the performance guarantees; show the improvement of the multidimensional adaptation; evaluate the performance in batch and in online learning scenarios; and quantify the performance of the extension presented. The methods presented can be implemented using MRCpy library (Bondugula et al., 2021) and the specific code used in the experimental results is provided on the web `https://github.com/MachineLearningBCAM/Supervised-learning-evolving-task-JMLR-2025`.

We utilize 13 public datasets that have been often used as benchmark for tasks that are in a sequence (see Table 4 in Appendix I). The tabular datasets (7) are divided in segments of 300 samples corresponding to consecutive times where each task corresponds to each of those segments; while the rest of the datasets (6) are composed by evolving tasks (images with characteristics/quality/realism that change over time). The samples in each task are randomly splitted in 100 samples for test and the rest of samples for training. In each repetition, the samples used for training are randomly sampled from the pool of training samples for each task.

The choice of the feature mapping has a significant impact on the final performance. The results for the proposed methods are obtained using a feature mapping defined by multiple features over instances together with one-hot encoding of labels as described in (2). The map Ψ in (2) that represents instances as real vectors is given by the pixel values for the "Rotated MNIST" dataset, by the last layer of the ResNet18 pre-trained network for the rest of image



(a) Bounds for error probabilities of each j-th task leveraging information from the $j-1$ preceding tasks.



(b) Bounds for error probabilities of each j-th task leveraging information from the k tasks.

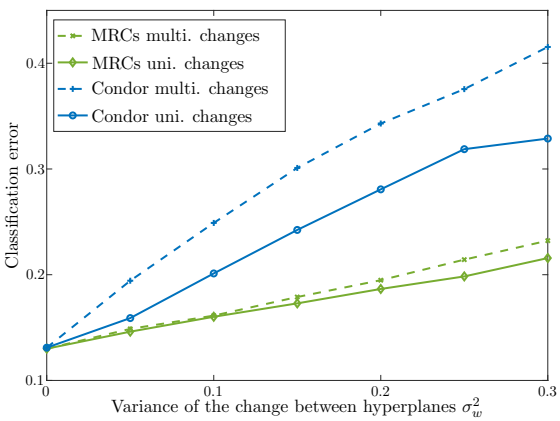
Figure 7: Results on synthetic data show the evolution over tasks of performance bounds and error probabilities.

datasets, and by random Fourier featuress (RFFs) (Shen et al., 2019) with 200 Gaussian weights and covariance matrix given by $\sigma^2 = 10$ for the tabular datasets. The confidence vector $\boldsymbol{\lambda}$ in equation (6) is obtained with $\lambda_0 = 0.7$, vector $\boldsymbol{\sigma}_j^2$ in equation (6) is given by the variance of n_j samples, vector $\boldsymbol{d}_j$ in equation (13) is estimated using $W = 2$, variance $\overline{\boldsymbol{d}}_j$ of the noise process $\overline{\boldsymbol{w}}_j$ in (25) is estimated using the recursive approach presented in Akhlaghi et al. (2017); and the proposed methodology applied to CL in Section 6 is implemented using $b = 3$ backward steps. Value for hyper-parameter λ_0 can be selected by methods such as cross-validation over a grid of possible values. For simplicity, we select the value by inspection over the synthetic data. We use the same value for all the numerical results for fair comparison with the state-of-the-art and to show that the methodology presented does not heavily rely on its value.

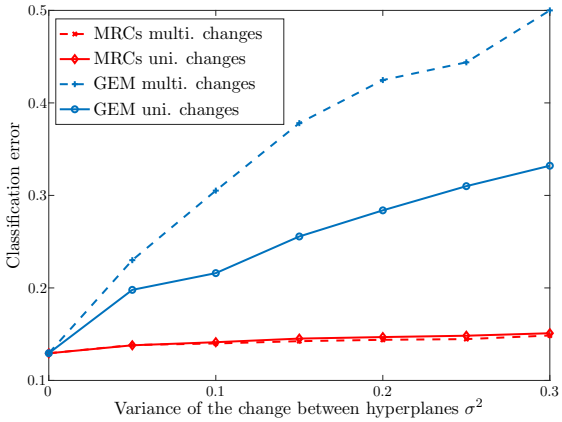
8.1 Tightness of the performance guarantees

In this section, we show the tightness of the presented bounds for error probabilities using the synthetic data since the error probability cannot be computed using real datasets. These numerical results are obtained averaging the classification errors and the bounds achieved with 10000 random instantiations of data samples in the synthetic data. Such data comprises a rotating hyperplane in 2 dimensional space where each coefficient of the hyperplane rotates 5 degrees between consecutive tasks.

Figures 7a and 7b show the averaged bounds for error probabilities corresponding to inequality (7) and the minimax risk in comparison with the true error probabilities using $n = 10$ samples per task. Figure 7a shows bounds for error probabilities of each j-th task $R(\mathrm{h}_j^{j-1})$ obtained leveraging information from sample sets $D_1, D_2, \ldots, D_{j-1}$ (forward learning) and Figure 7b shows bounds for error probabilities of each j-th task $R(\mathrm{h}_j^{k})$ obtained leveraging information from sample sets $D_1, D_2, \ldots, D_k$ (forward and backward learning).



(a) Classification error leveraging information from the $j-1$ preceding tasks.



(b) Classification error leveraging information from the k tasks.

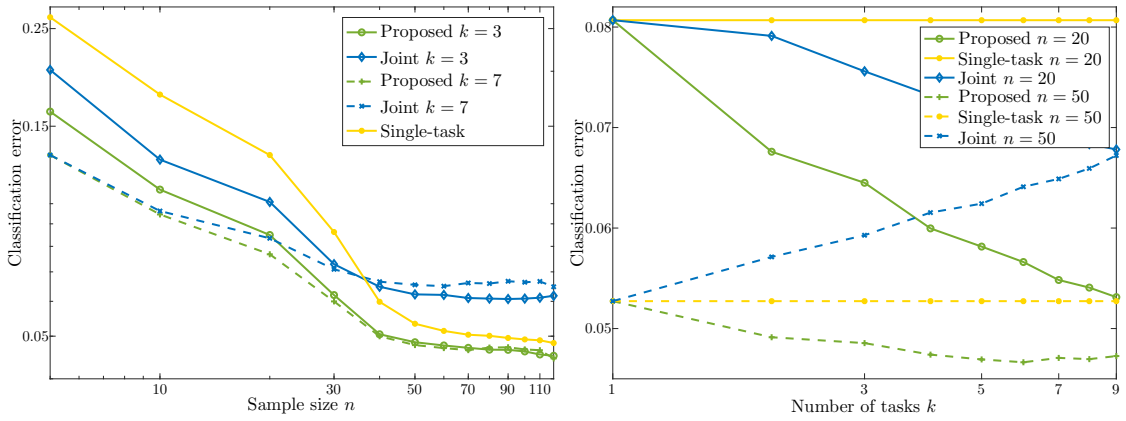
Figure 8: Results on synthetic data show the multidimensional adaptation to tasks' changes of the proposed methodology.

Such figures show that the bounds $R(\mathcal{U}_j^{j-1})$ and $R(\mathcal{U}_j^k)$ can offer, for each j-th task, tight upper bounds for error probabilities $R(\mathrm{h}_j^{j-1})$ and $R(\mathrm{h}_j^k)$, respectively.

8.2 Improvement of multidimensional adaptation

This section shows the improvement of multidimensional adaptation using the synthetic data since the tasks' changes cannot be modified in real datasets. These numerical results are obtained averaging the classification errors achieved with 100 random instantiations of data samples in the synthetic data with $n = 100$ samples per task. Such data comprises a rotating hyperplane in 5 dimensional space where each coefficient of the hyperplane changes between consecutive tasks. Specifically, for each j-th task, we obtain the coefficients of the hyperplane $\mathbf{w}_j$ by adding a Gaussian random variable to the coefficients of the preceding task as $\mathbf{w}_j = \mathbf{w}_{j-1} + N(\mathbf{0}, \sigma_w^2 \mathbf{I})$ for multidimensional (multi.) changes and as $\mathbf{w}_j = \mathbf{w}_{j-1} + \mathbf{1}N(0, \sigma_w^2)$ for unidimensional (uni.) changes where σ_w^2 denotes the variance of the change between consecutive hyperplanes. The results of the proposed methodology are compared with state-of-the-art techniques: Condor (Zhao et al., 2020) that leverages information from preceding tasks and gradient episodic memory (GEM) (Lopez-Paz and Ranzato, 2017) that leverages information from all the tasks in the sequence.

Figures 8a and 8b show the classification error of the proposed methodology and state-of-the-art techniques increasing the variance of the change between consecutive tasks. Such figures show the performance improvement due to the multidimensional adaptation in comparison with state-of-the-art techniques leveraging information from preceding tasks (forward learning) and from all the tasks in the sequence (forward and backward learning). Figure 8 shows that the proposed methodology better account for multidimensional tasks' changes than state-of-the-art techniques.



(a) Classification error for MDA per sample size.

(b) Classification error for MDA per number of tasks.

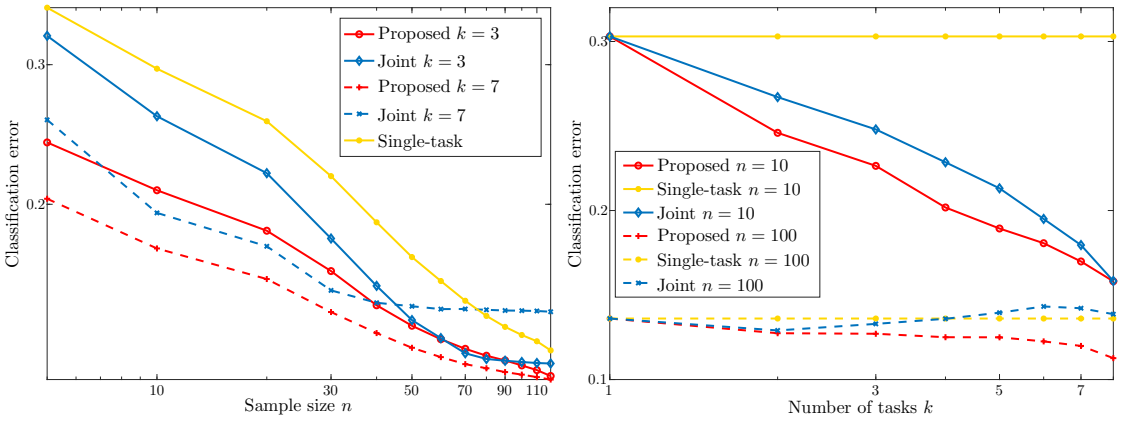
Figure 9: Results on "Yearbook" dataset show the relationship among classification error, number of tasks, and sample size for MDA.

8.3 Classification error in batch learning scenarios

In the following we show the relationship among classification error, number of tasks, and sample size in batch learning scenarios using real datasets. These numerical results are obtained averaging, for each number of tasks and sample size, the classification errors achieved with 10 random instantiations of data samples in "Yearbook" dataset. The performance improvement of the proposed methodology is compared with relevant baselines for learning from a sequence of tasks: joint learning (also known as offline learning) (Mai et al., 2022) and single-task learning (also known as independent learning) (Riemer et al., 2018). Joint learning and single-task learning obtain classification rules as in standard supervised classification using the samples from all the tasks and using samples only from the corresponding task, respectively.

Figures 9a and 9b show the classification error for MDA for different sample sizes and number of tasks. Such figures show that the proposed learning methodology achieves significantly better results than joint learning and single-task learning. In particular, Figure 9a shows that for $n = 20$ samples per tasks joint learning requieres $k = 7$ tasks to achieve similar results to the proposed methodology with $k = 3$; and shows that single-task learning requieres $n = 30$ samples to achieve similar results to the proposed methodology with $k = 3$ tasks and $n = 20$ samples per task. In addition, Figure 9b shows that the proposed methodology improves performance increasing the number of tasks, while the performance of single-task learning remains constant increasing the number of tasks and the performance of joint learning decreases with $n = 50$ samples per task.

Figures 10a and 10b show the classification error for MTL for different sample sizes and number of tasks. Such figures show that the proposed learning methodology achieves significantly better results than joint learning and single-task learning. In particular, Figure 10a shows that with $n = 40$ samples per task, joint learning requieres $k = 7$ tasks to achieve similar results to the proposed methodology with $k = 3$; and shows that single-task learning requieres $n = 70$ samples to achieve similar results to the proposed methodology



(a) Classification error for MTL per sample size. (b) Classification error for MTL per number of tasks.

Figure 10: Results on "Yearbook" dataset show the relationship among classification error, number of tasks, and sample size for MTL.

Table 2: Classification error and standard deviation of the proposed methodology in comparison with the state-of-the-art techniques SCD.

Dataset	Condor		Drift Surf		AUE		Proposed methods	
n	10	100	10	100	10	100	10	100
BAF	1.11 ± 0.00	1.12 ± 0.00	1.08 ± 0.00	0.96 ± 0.00	1.06 ± 0.00	1.05 ± 0.00	**0.59** ± 0.00	**0.59** ± 0.00
Elec2	38.90 ± 0.45	40.10 ± 0.30	42.90 ± 0.61	43.66 ± 0.43	42.38 ± 0.55	43.34 ± 0.37	**38.83** ± 0.66	**38.29** ± 0.21
Airlines	43.32 ± 0.20	43.42 ± 0.19	44.41 ± 0.00	45.66 ± 0.00	44.54 ± 0.00	45.74 ± 0.00	**39.07** ± 0.00	**38.74** ± 0.00
USPS	48.52 ± 5.58	48.01 ± 5.50	**38.00** ± 6.30	38.00 ± 5.25	43.60 ± 6.46	38.80 ± 5.74	40.61 ± 4.43	**34.66** ± 1.89
Spam	**24.72** ± 1.59	26.25 ± 1.93	33.68 ± 3.85	32.17 ± 2.17	27.38 ± 3.02	29.70 ± 1.59	26.23 ± 1.53	**20.80** ± 0.91
Power supply	**34.33** ± 0.45	33.10 ± 0.24	46.27 ± 0.95	43.32 ± 0.43	46.27 ± 0.95	43.32 ± 0.43	40.26 ± 0.80	**28.99** ± 0.29

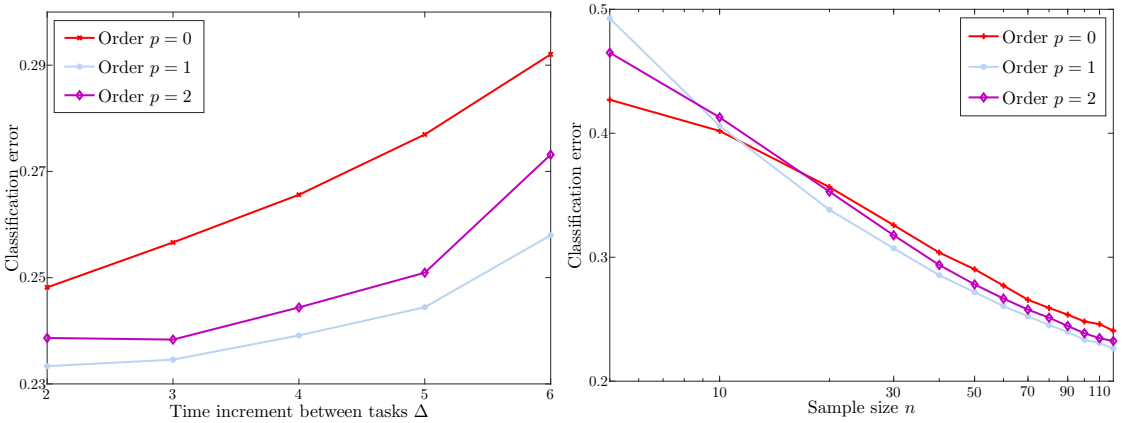
with $k = 3$ tasks and $n = 40$ samples per task. In addition, Figure 10b shows that the proposed methodology can improve performance as tasks arrive. The methods proposed can effectively adapt to tasks' changes that improves classification performance in all the experimental results.

8.4 Classification error in online learning scenarios

In this section, we compare the performance of the proposed methodology applied to online learning scenarios with the state-of-the-art techniques for $n = 10$ and $n = 100$ samples per task. These numerical results are obtained computing the average classification error over all the tasks in 100 random instantiations of data samples. The proposed methodology is compared with 3 state-of-the-art SCD techniques: Condor (Zhao et al., 2020), DriftSurf (Tahmasbi et al., 2021), and accuracy updated ensemble (AUE) (Brzezinski and Stefanowski, 2013); and 3 state-of-the-art CL techniques: GEM (Lopez-Paz and Ranzato, 2017), meta-experience replay (MER) (Riemer et al., 2018), and elastic weight consolidation (EWC) (Kirkpatrick et al., 2017). The hyper-parameters in these methods are set to the default values provided by the authors.

Table 3: Classification error and standard deviation of the proposed methodology in comparison with the state-of-the-art techniques for CL.

Dataset	GEM		MER		EWC		Proposed methods	
n	10	100	10	100	10	100	10	100
Yearbook	43.53 ± 10.70	23.45 ± 6.98	38.62 ±11.92	19.37 ± 7.05	48.94 ± 6.06	37.17 ± 8.27	**14.18** ± 1.51	**9.42** ± 0.57
ImageNet noise	39.09 ± 7.71	13.78 ± 7.31	27.25 ± 8.39	12.71 ± 5.15	45.75 ± 6.69	30.68 ± 8.12	**14.48** ± 1.55	**8.68** ± 0.54
DomainNet	69.78 ± 5.06	53.60 ± 10.79	47.58 ± 9.71	30.26 ± 11.01	50.03 ± 6.31	41.17 ± 2.92	**33.81** ± 2.31	**24.53** ± 1.21
UTKFaces	12.20 ± 0.00	12.10 ± 0.00	12.13 ± 0.00	12.13 ± 0.00	12.13 ± 0.00	12.13 ± 0.00	**10.32** ± 0.18	**10.32** ± 0.00
CLEAR	56.60 ± 10.35	8.60 ± 2.02	20.53 ± 9.75	7.40 ± 3.27	62.73 ± 4.28	38.53 ± 7.11	**8.22** ± 1.74	**4.06** ± 0.73
Rotated MNIST	45.28 ± 4.48	32.02 ± 2.64	**36.34** ± 3.79	34.54 ± 4.47	47.05 ± 1.69	39.93 ± 1.53	37.02 ± 0.87	**21.04** ± 0.39
Rotated MNIST iid	37.43 ± 3.84	25.97 ± 1.13	**19.34** ± 0.93	**18.30** ± 0.15	43.41 ± 0.02	33.13 ± 1.08	44.05 ± 0.01	24.65 ± 0.01



(a) Classification error per time increment.

(b) Classification error per sample size.

Figure 11: Results on "Rotated MNIST" dataset show the improvement of accounting for high-order time dependences.

Tables 2 and 3 show the classification error and the standard deviation of the state-of-the-art techniques using 12 real datasets. [1] Such tables show that the proposed methodology applied to SCD and CL offers an overall improved performance compared to existing techniques. The proposed methodology can significantly improve performance using evolving tasks with respect to the state-of-the-art techniques.

In order to more clearly assess the performance improvement of the proposed method using datasets with evolving tasks, we also use "Rotated MNIST i.i.d." dataset that is composed by non-evolving tasks. This dataset satisfies the (i.i.d.-A) assumption since it contains rotated images with angles uniformly sampled from $[0, 360)$ over tasks. Table 3 shows that the proposed methodology outperforms existing techniques in cases with evolving tasks; while existing techniques obtain improved performance in cases with non-evolving tasks. In particular, the performance of existing methods is significantly better in "Rotated MNIST i.i.d." than in "Rotated MINST" while the performance of the proposed method is significantly better in "Rotated MNIST" than in "Rotated MINST i.i.d." As shown in Table 3, the presented methodology can enable significant performance improvements using evolving tasks but not in datasets with non-evolving tasks.

1. Bold numbers indicate the top result for $n = 10$ and $n = 100$ samples per task.

8.5 High-order dependences

This section shows the classification performance of the proposed methodology accounting for high-order dependences among tasks. These numerical results are obtained averaging the classification errors for CL scenarios achieved with 100 random instantiations of data samples in "Yearbook" dataset using $n = 10$ samples per task. Figure 11 shows the relationship between classification error, order, and time increment. Such figure shows that order $p = 1$ can result in an improved overall performance at the expenses of worst initial performance. In addition, Figure 11 shows that order $p = 2$ may perform worse than order $p = 1$ because the number of model parameters increases with higher order.

The experimental results show that the proposed methodology achieves better performance than existing techniques. In addition, the numerical results show that the proposed performance guarantees can offer tight upper and lower bounds for error probabilities of each task.

9. Conclusion

The paper proposes a learning methodology for evolving tasks that is applicable to multiple supervised learning scenarios and provides computable performance guarantees. The proposed methodology accounts for multidimensional adaptation to changes by estimating multiple statistical characteristics of the underlying distribution. In addition, the paper analytically characterizes the increase in ESS achieved by the proposed methodology in terms of the expected quadratic change and the number of tasks. The numerical results assess the reliability of the performance guarantees presented and show the performance improvement in multiple supervised learning scenarios using multiple datasets, sample sizes, and number of tasks. The methodology proposed can lead to efficient methods for multiple learning scenarios and provide performance guarantees.

Acknowledgments

Funding in direct support of this work has been provided by projects PID2022-137063NBI00, PID2022-137442NB-I00, CNS2022-135203, and CEX2021-001142-S funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR, BCAM Severo Ochoa accreditation CEX2021-001142-S/MICIN/AEI/10.13039/501100011033 funded by the Ministry of Science and Innovation, and programes ELKARTEK, IT1504-22, and BERC-2022-2025 funded by the Basque Government.

Appendix A. Proof of Theorems 1, 2, and 7

Proof To prove Theorems 1 and 2, we use that the recursions in (8), (9), and recursions (14), (15) correspond to those for filtering in linear dynamical systems that obtain the unbiased linear estimator with minimum MSE.

The mean vectors $\boldsymbol{\tau}_j^\infty = \mathbb{E}_{\mathrm{p}_j}\{\Phi(x,y)\}$ evolve over tasks through the linear dynamical system

$$\boldsymbol{\tau}_{j+1}^\infty = \boldsymbol{\tau}_j^\infty + \boldsymbol{w}_{j+1} \tag{32}$$

because

$$\begin{aligned}\boldsymbol{\tau}_{j+1}^\infty = \mathbb{E}_{\mathrm{p}_{j+1}}\{\Phi(x,y)\} &= \int \Phi(x,y) d\mathrm{p}_{j+1}(x,y) \\ &= \int \Phi(x,y) d(\mathrm{p}_j + \varepsilon_{j+1})(x,y) = \boldsymbol{\tau}_j^\infty + \boldsymbol{w}_{j+1}\end{aligned} \tag{33}$$

with $\boldsymbol{w}_j = \int \Phi(x,y) d\varepsilon_j(x,y)$. Equality (33) follows since the sequence of probability distributions satisfies the (Evo-A) assumption, that is

$$\mathrm{p}_{j+1} = \mathrm{p}_j + \varepsilon_{j+1}.$$

So that the random vectors $\boldsymbol{w}_j$ are independent and zero-mean because they are a linear function of the independent zero-mean random measures ε_{j+1} (see e.g., Kallenberg (2017)).

Each state variable $\boldsymbol{\tau}_j^\infty$ is observed at each step j through $\boldsymbol{\tau}_j$ that is the sample average of i.i.d. samples from p_j, so that we have

$$\boldsymbol{\tau}_j = \boldsymbol{\tau}_j^\infty + \boldsymbol{v}_j \tag{34}$$

where $\boldsymbol{v}_j$ for $j \in \{1, 2, \ldots, k\}$ are independent and zero-mean, and independent of $\boldsymbol{w}_j$ for $j \in \{1, 2, \ldots, k\}$. Therefore, equations (32) and (34) above describe a linear dynamical system (state-space model with white noise processes) (Bishop, 2006; Anderson and Moore, 2012). For such systems, the Kalman filter recursions provide the unbiased linear estimator with minimum MSE based on samples corresponding to preceding steps $D_1, D_2, \ldots, D_j$ (Bishop, 2006; Anderson and Moore, 2012). The Kalman filter recursions are given by

$$\boldsymbol{\tau}_j^{j-1} = \boldsymbol{\tau}_{j-1}^{j-1} \tag{35}$$

$$\boldsymbol{s}_j^{j-1} = \boldsymbol{s}_{j-1}^{j-1} + \boldsymbol{d}_j \tag{36}$$

$$\boldsymbol{\tau}_j^{j} = \boldsymbol{\tau}_j^{j-1} + \boldsymbol{\eta}_j^j (\boldsymbol{\tau}_j - \boldsymbol{\tau}_j^{j-1}) \tag{37}$$

$$\boldsymbol{s}_j^{j} = \boldsymbol{s}_j^{j-1} + \boldsymbol{\eta}_j^j \boldsymbol{s}_j^{j-1} \tag{38}$$

$$\boldsymbol{\eta}_j^j = \frac{\boldsymbol{s}_j^{j-1}}{\boldsymbol{s}_j^{j-1} + \boldsymbol{s}_j}. \tag{39}$$

that lead to (8)-(10) substituting $\boldsymbol{\tau}_j^{j-1}$ and $\boldsymbol{s}_j^{j-1}$ in equations (37)-(39).

The Rauch-Tung-Striebel smoother recursions provide the unbiased linear estimator with minimum MSE based on samples corresponding to preceding and succeeding steps

$D_1, D_2, \ldots, D_k$ (Bishop, 2006; Anderson and Moore, 2012). The Rauch-Tung-Striebel smoother recursions are given by

$$\boldsymbol{\tau}_j^k = \boldsymbol{\tau}_j^j + \boldsymbol{\eta}_j^j(\boldsymbol{\tau}_{j+1}^k - \boldsymbol{\tau}_{j+1}^j) \tag{40}$$
$$\boldsymbol{s}_j^k = \underline{\mathrm{s}}_j^j + \boldsymbol{\eta}_j^j(\boldsymbol{s}_{j+1}^k - \boldsymbol{s}_{j+1}^j)\boldsymbol{\eta}_j^j \tag{41}$$
$$\boldsymbol{\eta}_j^j = \frac{\boldsymbol{s}_j^j}{\boldsymbol{s}_{j+1}^j} \tag{42}$$

Substituting $\boldsymbol{\tau}_{j+1}^j$ and $\boldsymbol{s}_{j+1}^j$ in equations (40)-(42), we have that

$$\boldsymbol{\tau}_j^k = \boldsymbol{\tau}_j^j + \boldsymbol{\eta}_j^j(\boldsymbol{\tau}_{j+1}^k - \boldsymbol{\tau}_j^j) \tag{43}$$
$$\boldsymbol{s}_j^k = \boldsymbol{s}_j^j + \boldsymbol{\eta}_j^j(\boldsymbol{s}_{j+1}^k - \boldsymbol{s}_j^j + \boldsymbol{d}_{j+1})\boldsymbol{\eta}_j^j \tag{44}$$
$$\boldsymbol{\eta}_j^j = \frac{\boldsymbol{s}_j^j}{\boldsymbol{s}_j^j + \boldsymbol{d}_{j+1}} \tag{45}$$

that lead to (14)-(16) by defining $\boldsymbol{\eta}_j^k = 1 - \boldsymbol{\eta}_j^j = \frac{\boldsymbol{d}_{j+1}}{\boldsymbol{s}_j^j + \boldsymbol{d}_{j+1}}$. Then, equations (8) and (9), and equations (14) and (15) are obtained after some algebra from the Kalman filter recursions and Rauch-Tung-Striebel smoother recursions.

Similarly as proof of Theorems 1 and 2, we prove Theorem 7 using that recursions (26), (27), and recursions (29), (30) are obtained after some algebra from the Kalman recursions for updated state vector and updated MSE matrix and from the Rauch-Tung-Striebel smoother recursions. The unbiased linear estimator with minimum MSE for a dynamical system such as (25) is given by the Kalman filter recursions (see e.g., Bishop (2006); Anderson and Moore (2012)).

Let $\tau_j^{\infty(i)}$ denote the i-th component of the mean vector τ_j^{∞}. Dynamical systems as that given by (25) can be derived using the p-th order Taylor expansion of the expectations $\tau_j^{\infty(i)}$ with $i = 1, 2, \ldots, m$. The Taylor expansions of $\tau_j^{\infty(i)}$ and its successive derivatives up to order p are given by

$$\begin{aligned}
\tau_j^{\infty(i)} &\approx {\tau_{j-1}^{\infty}}^{(i)} + \Delta_j \tau_j^{\infty(i)\prime} + \frac{\Delta_j^2}{2}\tau_j^{\infty(i)\prime\prime} + \ldots + \frac{\Delta_j^p}{p!}{\tau_{j-1}^{\infty}}^{(i)p)} \\
\tau_j^{\infty(i)\prime} &\approx {\tau_{j-1}^{\infty}}^{(i)\prime} + \Delta_j {\tau_{j-1}^{\infty}}^{(i)\prime\prime} + \ldots + \frac{\Delta_j^{p-1}}{(p)!}{\tau_{j-1}^{\infty}}^{(i)p-1)} \\
&\vdots \\
\tau_j^{\infty(i)p)} &\approx {\tau_{j-1}^{\infty}}^{(i)p)}
\end{aligned}$$

where Δ_j is the time increment between the j-th and the $j-1$-th tasks and $\tau_j^{\infty(i)^{p)}}$ denotes the p derivative of $\tau_j^{\infty(i)}$. The above equations lead to

$$\boldsymbol{\gamma}_{j,i}^{\infty} \approx \begin{pmatrix} 1 & \Delta_j & \frac{\Delta_j^2}{2} & ... & \frac{\Delta_j^p}{p!} \\ 0 & 1 & \Delta_j & ... & \frac{\Delta_j^{p-1}}{(p-1)!} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & ... & 1 \end{pmatrix} \boldsymbol{\gamma}_{j-1,i}^{\infty} \tag{46}$$

since $\boldsymbol{\gamma}_{j,i}^{\infty}$ is composed by $\tau_j^{\infty(i)}$ and its derivatives up to order p. In addition, $\tau_j^{\infty(i)}$ is observed at each time t through the sample set D_j, so that we have

$$\frac{1}{n_j} \sum_{l=1}^{n_j} \Phi(x_{j,l}, y_l)^{(i)} \approx \tau_j^{\infty(i)} \tag{47}$$

with $i = 1, 2, \ldots, m$. Equations (46) and (47) above lead to the dynamical system in (25). ■

Appendix B. Proof of Theorem 3

Proof To obtain bound in (18) we first prove that the mean vector estimate and the MSE vector given by (8) and (9), respectively, satisfy

$$\mathbb{P}\left\{ |\tau_j^{\infty(i)} - \tau_j^{j(i)}| \leq \kappa \sqrt{2 s_j^{j(i)} \log\left(\frac{2m}{\delta}\right)} \right\} \geq (1-\delta) \tag{48}$$

for any component $i = 1, 2, \ldots, m$. Then, we prove that $\| \sqrt{\boldsymbol{s}_j^j} \|_\infty \leq M/\sqrt{n_j^j}$ for $j \in \{1, 2, \ldots, k\}$, where the ESSs satisfy $n_1^1 = n_1$ and $n_j^j \geq n_j + n_{j-1}^{j-1} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + \|\boldsymbol{d}_j\|_\infty n_{j-1}^{j-1}}$ for $j \geq 2$.

To obtain inequality (48), we prove by induction that each component $i = 1, 2, \ldots, m$ of the error in the mean vector estimate $z_j^{j(i)} = \tau_j^{\infty(i)} - \tau_j^{j(i)}$ is sub-Gaussian with parameter $\rho_j^{j(i)} \leq \kappa \sqrt{s_j^{j(i)}}$. Firstly, for $j = 1$, we have that

$$z_1^{1(i)} = \tau_1^{\infty(i)} - \tau_1^{1(i)} = \tau_1^{\infty(i)} - \tau_1^{(i)}.$$

Since the bounded random variable $\Phi_1^{(i)}$ is sub-Gaussian with parameter $\sigma(\Phi_1^{(i)})$, then the error in the mean vector estimate $z_1^{1(i)}$ is sub-Gaussian with parameter that satisfies

$$\left(\rho_1^{1(i)}\right)^2 = \frac{\sigma\left(\Phi_1^{(i)}\right)^2}{n_1} \leq \frac{\kappa^2 \sigma_1^{2(i)}}{n_1} = \kappa^2 s_1^{(i)}.$$

If $z_{j-1}^{j-1(i)} = \tau_{j-1}^{\infty}{}^{(i)} - \tau_{j-1}^{j-1(i)}$ is sub-Gaussian with parameter $\rho_{j-1}^{j-1(i)} \leq \kappa\sqrt{s_{j-1}^{j-1(i)}}$ for any $i = 1, 2, \ldots, m$, then using the recursions (8) and (9) we have that

$$\begin{aligned} z_j^{j(i)} &= \tau_j^{\infty(i)} - \tau_j^{j(i)} = \tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} - \tau_j^{(i)} - \frac{s_j^{(i)}}{s_{j-1}^{j-1(i)} + s_j^{(i)} + d_j^{2(i)}} \left(\tau_{j-1}^{j-1(i)} - \tau_j^{(i)}\right) \\ &= \tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} - \tau_{j-1}^{j-1(i)} + \left(1 - \frac{s_j^{(i)}}{s_{j-1}^{j-1(i)} + s_j^{(i)} + d_j^{2(i)}}\right) \left(\tau_{j-1}^{j-1(i)} - \tau_j^{(i)}\right) \\ &= \tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} - \tau_{j-1}^{j-1(i)} - \frac{s_j^{j(i)}}{s_j^{(i)}} \left(\tau_j^{(i)} - \tau_{j-1}^{j-1(i)}\right) \end{aligned}$$

since $\boldsymbol{w}_j = \boldsymbol{\tau}_j^{\infty} - \boldsymbol{\tau}_{j-1}^{\infty}$. If $\boldsymbol{v}_j = \boldsymbol{\tau}_j - \boldsymbol{\tau}_j^{\infty}$, the error in the mean vector estimate is given by

$$\begin{aligned} z_j^{j(i)} &= \tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} - \tau_{j-1}^{j-1(i)} - \frac{s_j^{j(i)}}{s_j^{(i)}} \left(\tau_j^{\infty(i)} + v_j^{(i)} - \tau_{j-1}^{j-1(i)}\right) \\ &= \tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} - \tau_{j-1}^{j-1(i)} - \frac{s_j^{j(i)}}{s_j^{(i)}} \left(\tau_{j-1}^{\infty}{}^{(i)} + w_j^{(i)} + v_j^{(i)} - \tau_{j-1}^{j-1(i)}\right) \\ &= \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right) z_{j-1}^{j-1(i)} + \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right) \left(w_j^{(i)}\right) - \frac{s_j^{j(i)}}{s_j^{(i)}} v_j^{(i)} \end{aligned}$$

where $w_j^{(i)}$ and $v_j^{(i)}$ are sub-Gaussian with parameter $\sigma(w_j^{(i)})$ and $\sigma\left(\Phi_j^{(i)}\right)/\sqrt{n_j}$, respectively. Therefore, we have that $z_j^{j(i)}$ is sub-Gaussian with parameter $\rho_j^{j(i)}$ that satisfies

$$\left(\rho_j^{j(i)}\right)^2 = \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \left(\rho_{j-1}^{j-1(i)}\right)^2 + \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \sigma\left(w_j^{(i)}\right)^2 + \left(\frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \frac{\sigma\left(\Phi_j^{(i)}\right)^2}{n_j}$$

since $\boldsymbol{z}_{j-1}^{j-1}$, $\boldsymbol{w}_j$, and $\boldsymbol{v}_j$ are independent. Using that $\rho_{j-1}^{j-1(i)} \leq \kappa\sqrt{s_{j-1}^{j-1(i)}}$ and the definition of κ, we have that

$$\begin{aligned} \left(\rho_j^{j(i)}\right)^2 &\leq \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \kappa^2 s_{j-1}^{j-1(i)} + \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \kappa^2 d_j^{2(i)} + \left(\frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \kappa^2 \frac{\sigma_j^{2(i)}}{n_j} \\ &\leq \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right)^2 \kappa^2 \left(\left(\frac{1}{s_j^{j(i)}} - \frac{1}{s_{j^{(i)}}}\right)^{-1} + d_j^{2(i)}\right) + \frac{\left(s_j^{j(i)}\right)^2}{s_j^{(i)}} \kappa^2 \qquad (49) \\ &= \left(1 - \frac{s_j^{j(i)}}{s_j^{(i)}}\right) \kappa^2 s_j^{j(i)} + \kappa^2 \frac{\left(s_j^{j(i)}\right)^2}{s_j^{(i)}} \end{aligned}$$

where (49) is obtained using (9).

The inequality in (48) is obtained using the union bound together with the Chernoff bound (concentration inequality) (see e.g., Wainwright (2019)) for the random variables $z_j^{j(i)}$ that are sub-Gaussian with parameter $\rho_j^{j(i)}$.

Now, we prove by induction that, for any j, $\|\sqrt{\boldsymbol{s}_j^j}\|_\infty \leq M/\sqrt{n_j^j}$ where the ESSs satisfy $n_1^1 = n_1$ and

$$n_j^j \geq n_j + n_{j-1}^{j-1} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + \|\boldsymbol{d}_j\|_\infty n_{j-1}^{j-1}}$$

for $j \geq 2$. For $j = 1$, using the definition of $\boldsymbol{s}_j^j$ in equation (9), we have that for any component i

$$\left(s_1^{1(i)}\right)^{-1} = \left(s_1^{(i)}\right)^{-1} = \frac{n_1}{\sigma_1^{2(i)}} \geq \frac{n_1}{M^2}.$$

Then, vector $\boldsymbol{s}_1^1$ satisfies

$$\|\sqrt{\boldsymbol{s}_1^1}\|_\infty \leq \frac{M}{\sqrt{n_1}} = \frac{M}{\sqrt{n_1^1}}.$$

If $\|\sqrt{\boldsymbol{s}_{j-1}^{j-1}}\|_\infty \leq M/\sqrt{n_{j-1}^{j-1}}$, then we have that for any component i

$$\begin{aligned}\left(s_j^{j(i)}\right)^{-1} &= \frac{1}{s_j^{(i)}} + \frac{1}{s_{j-1}^{j-1(i)} + d_j{}^{(i)}} \geq \frac{1}{s_j^{(i)}} + \frac{1}{\frac{M^2}{n_{j-1}^{j-1}} + d_j{}^{(i)}} \geq \frac{1}{M^2}\left(n_j + \frac{1}{\frac{1}{n_{j-1}^{j-1}} + \frac{d_j{}^{(i)}}{M^2}}\right) \\ &\geq \frac{1}{M^2}\left(n_j + \frac{1}{\frac{1}{n_{j-1}^{j-1}} + \frac{\|\boldsymbol{d}_j\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty}}\right)\end{aligned}$$

by using the recursion (9) and the induction hypothesis. Then, vector $\boldsymbol{s}_j^j$ satisfies

$$\left\|\sqrt{\boldsymbol{s}_j^j}\right\|_\infty \leq \frac{M}{\sqrt{n_j + n_{j-1}^{j-1} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + \|\boldsymbol{d}_j\|_\infty n_{j-1}^{j-1}}}}.$$

The inequality in (18) is obtained because the minimax risk is bounded by the smallest minimax risk as shown in (Mazuelas et al., 2020, 2022, 2023) so that

$$R(\mathcal{U}_j^j) \leq R_k^\infty + \left(\|\boldsymbol{\tau}_k^\infty - \boldsymbol{\tau}_j^j\|_\infty + \|\boldsymbol{\lambda}_j^j\|_\infty\right) \|\boldsymbol{\mu}_k^\infty\|_1$$

that leads to (18) using (48) and $\|\sqrt{\boldsymbol{s}_j^j}\|_\infty \leq M/\sqrt{n_j^j}$. ■

Appendix C. Proof of Theorem 4

Proof To obtain bound in (20), we proceed by induction. For $j = 1$, using the expression for the ESS in (18), we have that

$$n_1^1 = n_1 \geq n.$$

If (20) holds for the $(j-1)$-task, then for the j-th task, we have that

$$n_j^j \geq n_j + n_{j-1}^{j-1} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + n_{j-1}^{j-1}\|\boldsymbol{d}_j\|_\infty} \geq n + n_{j-1}^{j-1} \frac{1}{1 + n_{j-1}^{j-1} d} = n\left(1 + \frac{1}{\frac{n}{n_{j-1}^{j-1}} + nd}\right)$$

where the second inequality is obtained because $n_j \geq n$, $\|\boldsymbol{\sigma}_j^2\|_\infty \leq 1$, and $\|\boldsymbol{d}_j\|_\infty \leq d$. Using that $n_{j-1}^{j-1} \geq n\left(1 + \frac{(1+\alpha)^{2j-3}-1-\alpha}{\alpha(1+\alpha)^{2j-3}+\alpha}\right)$, the ESS of the j-th task satisfies

$$\begin{aligned} n_j^j &\geq n\left(1 + \frac{1}{\frac{n}{n\left(1+\frac{(1+\alpha)^{2j-3}-1-\alpha}{\alpha(1+\alpha)^{2j-3}+\alpha}\right)} + nd}\right) = n\left(1 + \frac{1}{\frac{\alpha(1+\alpha)^{2j-3}+\alpha}{(1+\alpha)^{2j-2}-1} + nd}\right) \\ &= n\left(1 + \frac{1}{\frac{\alpha(1+\alpha)^{2j-3}+\alpha}{(1+\alpha)^{2j-2}-1} + \frac{\alpha^2}{\alpha+1}}\right) \qquad (50) \\ &= n\left(1 + \frac{(1+\alpha)^{2j-1} - 1 - \alpha}{\alpha(1+\alpha)^{2j-2} + \alpha(\alpha+1) + \alpha^2(1+\alpha)^{2j-2} - \alpha^2}\right) \end{aligned}$$

where (50) is obtained because $nd = \frac{\alpha^2}{\alpha+1}$ since $\alpha = \frac{nd}{2}\left(\sqrt{1+\frac{4}{nd}}+1\right)$.

Now, we obtain bounds for the ESS if $nd < \frac{1}{j^2}$. In the following, the constant ϕ represents the golden ratio $\phi = 1.618\ldots$.

1. If $nd < \frac{1}{j^2} \Rightarrow \sqrt{nd} \leq \alpha \leq \sqrt{nd}\phi \leq \frac{\phi}{j} \leq 1$ similarly as in the preceding case, then we have that n_j^j satisfies

   $$n_j^j \geq n\frac{1}{\alpha}\frac{\alpha(2j-2)}{2+\alpha(2j-1)} = n\frac{2j-2}{2+\alpha(2j-1)}$$

   where the first inequality follows because $(1+\alpha)^{2j-2} \geq 1 + \alpha(2j-2)$. Using $\alpha \leq \frac{\phi}{j}$, we have that

   $$n_j^j \geq n\frac{2j-2}{2+\frac{\phi}{j}(2j-1)} \geq n\frac{2j-2}{2+2\phi-\frac{\phi}{j}} \geq n\frac{j-1}{1+\phi}.$$

2. If $\frac{1}{j^2} \leq nd < 1 \Rightarrow \frac{1}{j} < \sqrt{nd} < \alpha < \sqrt{nd}\phi$ because $\alpha = \frac{nd}{2}\left(\sqrt{1+\frac{4}{nd}}+1\right) = \sqrt{nd}\frac{\sqrt{nd+4}+\sqrt{nd}}{2}$, then we have that n_j^j satisfies

   $$n_j^j \geq n\frac{1}{\sqrt{nd}}\frac{1}{\phi}\frac{(1+\alpha)^{2j-2}-1}{(1+\alpha)^{2j-2}+1} \geq n\frac{1}{\sqrt{nd}}\frac{1}{\phi}\frac{(1+\frac{1}{j})^{2j-2}-1}{(1+\frac{1}{j})^{2j-2}+1}$$

where the first inequality follows because $\alpha < \sqrt{nd}\phi$ and the second inequality follows because the expression is monotonically increasing for α and $\frac{1}{j} < \alpha$. Since $(1 + \frac{1}{j})^{2j-2} \geq 1 + \frac{2j-2}{j}$, we have that

$$n_j^j \geq n \frac{1}{\sqrt{nd}} \frac{1}{\phi} \frac{\frac{2j-2}{j}}{2 + \frac{2j-2}{j}} \geq n \frac{1}{\sqrt{nd}} \frac{1}{\phi} \frac{1}{3} \quad \text{because } j \geq 2.$$

3. If $nd \geq 1 \;\Rightarrow 1 \leq nd \leq \alpha \leq nd\phi$ because $\alpha = \frac{nd}{2}\left(\sqrt{1 + \frac{4}{nd}} + 1\right)$, then we have that n_j^j satisfies

$$n_j^j \geq n \frac{1}{nd\phi} \frac{(1+\alpha)^{2j-1} - 1 - \alpha}{(1+\alpha)^{2j-1} + 1} \geq n \frac{1}{nd\phi} \frac{(1+\alpha)^{2j-2} - 1}{(1+\alpha)^{2j-2} + 1}$$

where the first inequality follows because $\alpha \leq nd\phi$ and the second inequality follows multiplying and dividing by $1 + \alpha$ and because $1/(1 + \alpha) < 1$. Since the above expression is monotonically increasing for α and $\alpha \geq 1$, we have that

$$n_j^j \geq n \frac{1}{nd\phi} \frac{2^{2j-2} - 1}{2^{2j-2} + 1} \geq n \frac{1}{nd\phi} \frac{3}{5} \quad \text{because } j \geq 2.$$

■

Appendix D. Proof of Theorem 5

Proof To obtain bound in (21) we first prove that the mean vector estimate and the MSE vector given by (14), (15) satisfy

$$\mathbb{P}\left\{ |\tau_k^{\infty(i)} - \tau_j^{k(i)}| \leq \kappa \sqrt{2 s_j^{k(i)} \log\left(\frac{2m}{\delta}\right)} \right\} \geq (1 - \delta) \tag{51}$$

for any component $i = 1, 2, \ldots, m$. Then, we prove that $\|\boldsymbol{s}_j^k\|_\infty \leq M/\sqrt{n_j^k}$ for $j \in \{1, 2, \ldots, k\}$, where the ESSs satisfy $n_k^k = n_k^k$ and $n_j^k \geq n_j^j + n_{j+1}^{-k} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + n_{j+1}^{-k} \|\boldsymbol{d}_{j+1}^2\|_\infty}$ for $j \geq 2$.

As described in Appendix A, recursions in (14) and (15) for $\boldsymbol{\tau}_j^k$ and $\boldsymbol{s}_j^k$ correspond to the Rauch-Tung-Striebel smoother recursions. An alternative manner to obtain such mean and MSE vectors is using the fixed-lag smoother recursions that become

$$\boldsymbol{\tau}_j^k = \boldsymbol{\tau}_j^j + \frac{\boldsymbol{s}_j^j}{\boldsymbol{s}_j^j + \boldsymbol{s}_{j+1}^{-k} + \boldsymbol{d}_{j+1}} \left(\boldsymbol{\tau}_{j+1}^{-k} - \boldsymbol{\tau}_j^j \right), \quad \boldsymbol{s}_j^k = \left(\frac{1}{\boldsymbol{s}_j^j} + \frac{1}{\boldsymbol{s}_{j+1}^{-k} + \boldsymbol{d}_{j+1}} \right)^{-1} \tag{52}$$

where backward mean and MSE vectors $\boldsymbol{\tau}_j^{-k}, \boldsymbol{s}_j^{-k}$ are obtained using recursions in (8) and (9) in retrodiction. Specifically, vectors $\boldsymbol{\tau}_j^{-k}$ and $\boldsymbol{s}_j^{-k}$ are obtained using the same recursion as for $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ in (8) and (9) with $\boldsymbol{s}_{j+1}^{-k}, \boldsymbol{d}_{j+1}$, and $\boldsymbol{\tau}_{j+1}^{-k}$ instead of $\boldsymbol{s}_{j-1}^j, \boldsymbol{d}_j$, and $\boldsymbol{\tau}_{j-1}^{j-1}$.

To obtain inequality (51), we prove that each component $i = 1, 2, \ldots, m$ of the error in the mean vector estimate $z_j^{k(i)} = \tau_j^{\infty(i)} - \tau_j^{k(i)}$ is sub-Gaussian with parameter $\eta_j^{k(i)} \leq \kappa\sqrt{s_j^{k(i)}}$. Analogously to the proof of Theorem 3, it is proven that each component in the error of the backward mean vector $\boldsymbol{\tau}_{j+1}^{-k}$ is sub-Gaussian with parameters satisfying $\boldsymbol{\eta}_{j+1}^{-k} \preceq \kappa\sqrt{\boldsymbol{s}_{j+1}^{-k}}$. The error in the forward and backward mean vector estimate is given by

$$z_j^{k(i)} = \tau_j^{\infty(i)} - \tau_j^{k(i)} = \tau_j^{\infty(i)} - \tau_j^{j(i)} - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \left(\tau_{j+1}^{-k\,(i)} - \tau_j^{j(i)}\right)$$

where the second equality is obtained using the recursion for $\tau_j^{k(i)}$ in (52). Adding and subtracting $\frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \tau_{j+1}^{\infty}{}^{(i)}$, we have that

$$\begin{aligned} z_j^{k(i)} &= z_j^{j(i)} - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \left(\tau_{j+1}^{\infty}{}^{(i)} - \tau_{j+1}^{\infty}{}^{(i)} + \tau_{j+1}^{-k\,(i)} - \tau_j^{j(i)}\right) \\ &= z_j^{j(i)} - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \left(\tau_j^{\infty(i)} + w_{j+1}^{(i)} - z_{j+1}^{-k\,(i)} - \tau_j^{j(i)}\right) \end{aligned}$$

since $\boldsymbol{w}_j = \boldsymbol{\tau}_j^{\infty} - \boldsymbol{\tau}_{j-1}^{\infty}$ and $z_j^{j(i)} = \tau_j^{\infty(i)} - \tau_j^{j(i)}$. Then, we have that

$$\begin{aligned} z_j^{k(i)} &= z_j^{j(i)} - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \left(z_j^{j(i)} + w_{j+1}^{(i)} - z_{j+1}^{-k\,(i)}\right) \qquad (53) \\ &= \left(1 - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}^2{}^{(i)}}\right) z_j^{j(i)} - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}} \left(w_{j+1}^{(i)} - z_{j+1}^{-k\,(i)}\right) \end{aligned}$$

where $z_j^{j(i)}$, $z_{j+1}^{-k\,(i)}$, and $w_{j+1}^{(i)}$ are sub-Gaussian with parameters $\eta_j^{j(i)} \leq \kappa\sqrt{s_j^{j(i)}}$, $\eta_{j+1}^{-k\,(i)} \leq \kappa\sqrt{s_{j+1}^{-k\,(i)}}$, and $\sigma(w_j^{(i)})$, respectively. Since $\boldsymbol{z}_j^j$, $\boldsymbol{z}_{j+1}^{-k}$, and $\boldsymbol{w}_{j+1}$ are independent, we have that $z_j^{k(i)}$ given by (53) is sub-Gaussian with parameter that satisfies

$$\begin{aligned} \left(\eta_j^{k(i)}\right)^2 = &\left(1 - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}}\right)^2 \left(\eta_j^{j(i)}\right)^2 \\ &+ \left(\frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\,(i)} + d_{j+1}{}^{(i)}}\right)^2 \left(\sigma\left(w_j^{(i)}\right)^2 + \left(\eta_{j+1}^{-k\,(i)}\right)^2\right) \end{aligned}$$

$$\left(\eta_j^{k(i)}\right)^2 \leq \left(1 - \frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\right)^2 \kappa^2 s_j^{j(i)} + \left(\frac{s_j^{j(i)}}{s_j^{j(i)} + s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\right)^2 \kappa^2 \left({d_{j+1}}^{(i)} + s_{j+1}^{-k\ (i)}\right).$$

The sub-Gaussian parameter satisfies

$$\begin{aligned}\left(\eta_j^{k(i)}\right)^2 &\leq \left(1 - \frac{s_j^{k(i)}}{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\right)^2 \kappa^2 \left(\frac{1}{s_j^{k(i)}} - \frac{1}{s_{j+1}^{-k2(i)} + {d_{j+1}}^{(i)}}\right)^{-1} + \frac{\left(s_j^{k(i)}\right)^2}{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\kappa^2 \\ &= \left(\frac{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)} - s_j^{k(i)}}{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\right)\kappa^2 s_j^{k(i)} + \frac{\left(s_j^{k(i)}\right)^2}{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}}\kappa^2 = \kappa^2 s_j^{k(i)}.\end{aligned}$$

The inequality in (51) is obtained using the union bound together with the Chernoff bound (concentration inequality) (see e.g., Wainwright (2019)) for the random variables $z_j^{k(i)}$ that are sub-Gaussian with parameter $\eta_j^{k(i)}$.

Now, we prove that, for any j, $\|\sqrt{\boldsymbol{s}_j^k}\| \leq M/\sqrt{n_j^k}$ where the ESSs satisfy $n_j^k \geq n_j^j + n_{j+1}^{-k} \frac{\|\boldsymbol{\sigma}_j^2\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty + n_{j+1}^{-k}\|\boldsymbol{d}_{j+1}\|_\infty}$ for $j \geq 2$. Analogously to the proof of Theorem 3, we prove that the backward MSE vector $\boldsymbol{s}_{j+1}^{-k}$ satisfies $\|\sqrt{\boldsymbol{s}_{j+1}^{-k}}\|_\infty \leq M/\sqrt{n_{j+1}^{-k}}$. Then, using that $\|\sqrt{\boldsymbol{s}_{j+1}^{-k}}\|_\infty \leq M/\sqrt{n_{j+1}^{-k}}$, we have that for every component i

$$\begin{aligned}\left(s_j^{k(i)}\right)^{-1} &= \frac{1}{s_j^{j(i)}} + \frac{1}{s_{j+1}^{-k\ (i)} + {d_{j+1}}^{(i)}} \geq \frac{n_j^j}{\sigma_j^{2(i)}} + \frac{1}{\frac{M^2}{n_{j+1}^{-k}} + {d_{j+1}}^{(i)}} \\ &\geq \frac{1}{M^2}\left(n_j^j + \frac{1}{\frac{1}{n_{j+1}^{-k}} + \frac{d_{j+1}}{M^2}}\right) \geq \frac{1}{M^2}\left(n_j^j + \frac{1}{\frac{1}{n_{j+1}^{-k}} + \frac{\|d_{j+1}\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty}}\right).\end{aligned}$$

Then, we obtain

$$\|\sqrt{\boldsymbol{s}_j^k}\|_\infty \leq \frac{M}{\sqrt{n_j^j + \frac{1}{\frac{1}{n_{j+1}^{-k}} + \frac{\|\boldsymbol{d}_{j+1}\|_\infty}{\|\boldsymbol{\sigma}_j^2\|_\infty}}}}. \tag{54}$$

The inequality in (21) is obtained because the minimax risk is bounded by the smallest minimax risk as shown in Mazuelas et al. (2020, 2022) so that

$$R(\mathcal{U}_j^k) \leq R_j^\infty + \left(\|\boldsymbol{\tau}_j^\infty - \boldsymbol{\tau}_j^k\|_\infty + \|\boldsymbol{\lambda}_j^k\|_\infty\right)\left\|\boldsymbol{\mu}_j^\infty\right\|_1$$

that leads to (21) using (51) and (54). ■

Appendix E. Proof of Theorem 6

Proof To obtain bound in (22), we proceed by induction. For $j = k$, we have that

$$n_j^j = n_k^k.$$

If (22) holds for the $(j+1)$-th task, the for the j-th task, we have that

$$n_j^k \geq n_j^j + \frac{n_{j+1}^k(1 + n_j^j\|\boldsymbol{d}_{j+1}\|_\infty) - n_j^j}{n_{j+1}^k\|\boldsymbol{d}_{j+1}\|_\infty(1 + n_j^j\|\boldsymbol{d}_{j+1}\|_\infty) + 1} \geq n_j^j + \frac{n_{j+1}^k(1 + n_j^j d) - n_j^j}{n_{j+1}^k d(1 + n_j^j d) + 1}$$

where the second inequality is obtained because $\|\boldsymbol{d}_j\|_\infty \leq d$. Using that

$$n_{j+1}^k \geq n\left(1 + \frac{(1+\alpha)^{2j+1} - 1 - \alpha}{\alpha(1+\alpha)^{2j+1} + \alpha} + \frac{(1+\alpha)^{2(k-j-1)+1} - 1 - \alpha}{\alpha(1+\alpha)^{2(k-j-1)+1} + \alpha}\right)$$

the ESS of the j-th task satisfies

$$\begin{aligned} n_j^k &\geq n_j^j + n\left(1 + \frac{(1+\alpha)^{2(k-j)-1} - 1 - \alpha}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha}\right)\left(1 + \frac{n\left(1 + \frac{(1+\alpha)^{2(k-j)-1} - 1 - \alpha}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha}\right)}{nd}\right)^{-1} \\ &= n_j^j + n\frac{(1+\alpha)^{2(k-j)} - 1}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha}\left(1 + \frac{\alpha^2}{\alpha+1}\left(1 + \frac{(1+\alpha)^{2(k-j)-1} - 1 - \alpha}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha}\right)\right)^{-1} \end{aligned}$$

where the second equality follows because $nd = \frac{\alpha^2}{\alpha+1}$ since $\alpha = \frac{nd}{2}\left(\sqrt{1 + \frac{4}{nd}} + 1\right)$. Then, we have that

$$\begin{aligned} n_j^k \geq& n_j^j + n\frac{(1+\alpha)^{2(k-j)} - 1}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha} \\ &\cdot \left(\frac{((1+\alpha)^{2(k-j)-1} + 1)(\alpha + 1 + \alpha^2) + \alpha((1+\alpha)^{2(k-j)-1} - 1 - \alpha)}{(\alpha+1)((1+\alpha)^{2(k-j)-1} + 1)}\right)^{-1} \\ \geq& n_j^j + n\frac{(1+\alpha)^{2(k-j)} - 1}{\alpha(1+\alpha)^{2(k-j)-1} + \alpha}\frac{(\alpha+1)((1+\alpha)^{2(k-j)-1} + 1)}{(1+\alpha)^{2(k-j)+1} + 1}. \end{aligned}$$

Now, we obtain bounds for the ESS depending on the value value of nd. Such bounds are obtained similarly as in Theorem 4 and we also denote by ϕ the golden ratio $\phi = 1.618\ldots$.

1. If $nd < \frac{1}{j^2} \Rightarrow \sqrt{nd} \leq \alpha \leq \sqrt{nd}\phi \leq \frac{\phi}{j} \leq 1$ similarly as in the preceding case, then we have that n_j^k satisfies

$$n_j^k \geq n_j^j + n\frac{1}{\alpha}\frac{\alpha(2(k-j))}{2 + \alpha 2(k-j)} = n_j^j + n\frac{k-j}{1 + \alpha(k-j)} \geq n_j^j + n\frac{k-j}{1 + \frac{\phi}{j}(k-j)}$$

where the first inequality follows because $(1+\alpha)^{2(k-j)-1} \geq 1 + \alpha(2(k-j)-1)$ and the second inequality is obtained using $\alpha \leq \frac{\phi}{j}$.

2. If $\frac{1}{j^2} \leq nd < 1 \Rightarrow \frac{1}{j} \leq \sqrt{nd} \leq \alpha \leq \sqrt{nd}\phi$ because $\alpha = nd\frac{\sqrt{1+\frac{4}{nd}}+1}{2} = \sqrt{nd}\frac{\sqrt{nd+4}+\sqrt{nd}}{2}$, then we have that n_j^k satisfies

$$n_j^k \geq n_j^j \frac{n}{\alpha}\frac{(1+\alpha)^{2(k-j)}-1}{(1+\alpha)^{2(k-j)}+1} \geq n_j^j \frac{n}{\alpha}\frac{(1+\sqrt{nd})^{2(k-j)}-1}{(1+\sqrt{nd})^{2(k-j)}+1}$$

where the second inequality follows because the ESS is monotonically increasing for α and $\alpha \geq nd$. Since $(1+\sqrt{nd})^{2(k-j)} \geq 1 + 2\sqrt{nd}(k-j)$ and $k - j \geq 1$, we have that

$$n_j^k \geq n_j^j + \frac{n}{\alpha}\frac{\sqrt{nd}}{1+\sqrt{nd}} \geq n_j^j + n\frac{1}{\phi}\frac{1}{1+\sqrt{nd}}$$

because $\alpha \leq \sqrt{nd}\phi$.

3. If $nd \geq 1 \;\; \Rightarrow 1 \leq nd \leq \alpha \leq nd\phi$ because $\alpha = nd\frac{\sqrt{1+\frac{4}{nd}}+1}{2}$, then we have that n_j^k satisfies

$$n_j^k \geq n_j^j + n\frac{1}{\alpha}\frac{2^{2(k-j)}-1}{2^{2(k-j)}+1} \geq n_j^j + n\frac{1}{nd}\frac{1}{\phi}\frac{3}{5}$$

where the first inequality follows because the ESS is monotonically increasing for α and $\alpha \geq 1$ and the second inequality is obtained using $k - j \geq 1$ and $\alpha \leq nd\phi$.

■

Appendix F. ESSs of the baseline approach that utilizes sliding windows

The ESS of the proposed forward learning techniques can be compared with the ESS of techniques based on sliding windows that are commonly used to adapt to evolving tasks (Bifet and Gavalda, 2007; Zhang et al., 2016; Tahmasbi et al., 2021). Such techniques obtain classification rules for each task using the sample sets corresponding with the $\overline{W}$ closest tasks with $\overline{W}$ the window size value. In the following, we first show the ESS using sliding windows with preceding tasks and then using sliding windows with preceding and succeeding tasks.

If the mean vector is obtained for each j-th task as the sample average of the $\overline{W}$ sample sets $D_{j-\overline{W}}, D_{j-\overline{W}+2}, \ldots, D_{j-1}$, then the ESS satisfies

$$n_{j,\overline{W}} \geq n\frac{6\overline{W}}{(\overline{W}+1)(2\overline{W}+1)nd+6} \tag{55}$$

and if the mean and confidence vectors are obtained for each j-th task with $j \in \{1, 2, \ldots, k\}$ as the sample average of the $\overline{W}$ closest sample sets $D_{j-\overline{W}+\widehat{W}+1}, D_{j-\overline{W}+\widehat{W}+2}, \ldots, D_j, \ldots, D_{j+\overline{W}-\widehat{W}}$ then, the ESS satisfies

$$n_{j,\overline{W},\widehat{W}} \geq n\frac{6\overline{W}}{(6\widehat{W}^2 + 2\overline{W}^2 + 3\overline{W} + 1 - 6\widehat{W}\overline{W} - 6\widehat{W})nd+6}$$

with $1 \leq \overline{W} \leq j-1$ and $\boldsymbol{\sigma}_j, d$, and n as in Theorem 4.

Proof Let $\tau^{(i)}_{j,\overline{W}}$ be the i-th component of the sample average of the $\overline{W}$ closest sample sets to the j-th task, $\tau^{(i)}_{j,\overline{W},\widehat{W}}$ be the sample average of the $\overline{W}$ sample sets closest the j-th task, and $\tau^{(i)}_j$ be the i-th component of the sample average corresponding to the j-th task. We prove that

$$\mathbb{P}\left\{|\tau_j^{\infty(i)} - \bar{\tau}_j^{(i)}| \leq \kappa\sqrt{2\frac{1}{\bar{n}_j}\log\left(\frac{2m}{\delta}\right)}\right\} \geq (1-\delta) \tag{56}$$

for any component $i = 1, 2, \ldots, m$ with $\bar{\tau}^{(i)}_j = \tau^{(i)}_{j,\overline{W}}$ and $\bar{n}_j = n_{j,\overline{W}}$ and with $\bar{\tau}^{(i)}_j = \tau^{(i)}_{j,\overline{W},\widehat{W}}$ and $\bar{n}_j = n_{j,\overline{W},\widehat{W}}$. For any $j = 1, 2, \ldots, k$, we have that

$$\begin{aligned}
&\tau_j^{\infty(i)} - \tau^{(i)}_{j,\overline{W}} = \tau_j^{\infty(i)} - \frac{1}{W}\left(\tau^{(i)}_{j-\overline{W}} + \tau^{(i)}_{j-\overline{W}+1} + \ldots + \tau^{(i)}_{j-1}\right)\\
=&\frac{1}{\overline{W}}\left((\tau_j^{\infty(i)} - \tau^{(i)}_{j-\overline{W}}) + (\tau_j^{\infty(i)} - \tau^{(i)}_{j-\overline{W}+1}) + \ldots + (\tau_j^{\infty(i)} - \tau^{(i)}_{j-1})\right)\\
=&\frac{1}{\overline{W}}((\tau_j^{\infty(i)} - {\tau^{\infty}_{j-\overline{W}}}^{(i)} - v^{(i)}_{j-\overline{W}}) + (\tau_j^{\infty(i)} - {\tau^{\infty}_{j-\overline{W}+1}}^{(i)} - v^{(i)}_{j-\overline{W}+1}) + \ldots \qquad (57)\\
&+ (\tau_j^{\infty(i)} - {\tau^{\infty}_{j-1}}^{(i)} - v^{(i)}_{j-1}))\\
=&\frac{1}{\overline{W}}((\tau_j^{\infty(i)} - {\tau^{\infty}_{j-\overline{W}+1}}^{(i)} + w^{(i)}_{j-\overline{W}+1}) + (\tau_j^{\infty(i)} - {\tau^{\infty}_{j-\overline{W}+2}}^{(i)} + w^{(i)}_{j-\overline{W}+2}) + \ldots - \sum_{l=j-\overline{W}}^{k-1} v_l^{(i)})
\end{aligned}$$

where the first equality is by the definition of $\tau^{(i)}_{j,\overline{W}}$, (57) is due to $\boldsymbol{v}_j = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_{j,i}, y_{j,i})$, and (58) is due to $\boldsymbol{w}_j = \boldsymbol{\tau}_j^{\infty} - \boldsymbol{\tau}_{j-1}^{\infty}$. Using the definition of $\boldsymbol{w}_j$, we have that

$$\begin{aligned}
\tau_j^{\infty(i)} - \tau^{(i)}_{j,\overline{W}} &= \frac{1}{\overline{W}}\left(\sum_{l=j-W+1}^{j} w_l^{(i)} + \sum_{l=j-W+2}^{j} w_l^{(i)} + \ldots + w_j^{(i)} - \sum_{l=j-\overline{W}}^{j-1} v_l^{(i)}\right)\\
&= \frac{1}{\overline{W}}\sum_{l=j-\overline{W}+1}^{j}(l - j + \overline{W})w_l^{(i)} - \frac{1}{\overline{W}}\sum_{l=j-\overline{W}}^{j-1} v_l^{(i)}.
\end{aligned}$$

The sub-Gaussian parameter of the error in the mean vector estimate is given by

$$\left(\rho^{(i)}_{j,\overline{W}}\right)^2 \leq \frac{1}{\overline{W}^2}\sum_{l=j-W+1}^{j}(l-j+\overline{W})^2\sigma(w_l^{(i)})^2 + \frac{1}{\overline{W}^2}\sum_{l=j-\overline{W}}^{j-1}\frac{\sigma(\Phi_l^{(i)})^2}{n_l}.$$

Let κ and n_l be such that $\sigma(w_l^{(i)}) \preceq \kappa\sqrt{d_l}, \sigma(\Phi_l^{(i)}) \preceq \kappa\sigma_l$, and $n_l \geq n$ for any l, then we obtain

$$\begin{aligned}
\left(\rho^{(i)}_{j,\overline{W}}\right)^2 &\leq \frac{1}{\overline{W}^2}\sum_{l=j-\overline{W}+1}^{j}(l-j+\overline{W})^2\kappa^2 d_l + \frac{1}{\overline{W}^2}\sum_{i=j-\overline{W}}^{j-1}\kappa^2\frac{\sigma_l^2}{n}\\
&\leq \frac{1}{6\overline{W}}(\overline{W}+1)(2\overline{W}+1)\kappa^2 d + \kappa^2\frac{1}{\overline{W}n}. \qquad (58)
\end{aligned}$$

The inequality in (56) is obtained using the union bound together with the Chernoff bound (concentration inequality) (see e.g., Wainwright (2019)) for the random variables $\tau_j^{\infty(i)} - \tau_{j,\overline{W}}^{(i)}$ that are sub-Gaussian with parameter $\rho_{j,\overline{W}}^{(i)}$ and because $n_j \geq n$, $\|\boldsymbol{\sigma}_j^2\|_\infty \leq 1$, and $\|\boldsymbol{d}_j\|_\infty \leq d$.

Since $\tau_{j,\overline{W},\widehat{W}}^{(i)}$ is given by the sample average of $\overline{W}$ sample sets, then we have that

$$\begin{aligned}
\tau_j^{\infty(i)} - \tau_{j,\overline{W},\widehat{W}}^{(i)} =& \tau_j^{\infty(i)} - \frac{1}{\overline{W}} \left(\tau_{j-\widehat{W}+1}^{(i)} + \tau_{j-\widehat{W}+2}^{(i)} + \ldots + \tau_j^{(i)} + \tau_{j+1}^{(i)} + \ldots + \tau_{j+\overline{W}-\widehat{W}}^{(i)} \right) \\
=& \frac{1}{\overline{W}} \Big((\tau_j^{\infty(i)} - \tau_{j-\widehat{W}+1}^{(i)}) + (\tau_j^{\infty(i)} - \tau_{j-\widehat{W}+2}^{(i)}) + \ldots + (\tau_j^{\infty(i)} - \tau_j^{(i)}) \\
& + \ldots + (\tau_j^{\infty(i)} - \tau_{j+W-\widehat{W}}^{(i)}) \Big) \\
=& \frac{1}{W} \Big((\tau_j^{\infty(i)} - {\tau_{j-\widehat{W}+1}^{\infty}}^{(i)} - {v_{j-\widehat{W}+1}}^{(i)}) + \ldots + (\tau_j^{\infty(i)} - \tau_j^{\infty(i)} - v_j^{(i)}) \\
& + \ldots + (\tau_j^{\infty(i)} - {\tau_{j+\overline{W}-\widehat{W}}^{\infty}}^{(i)} - v_{j+\overline{W}-\widehat{W}}^{(i)}) \Big) \qquad (59) \\
=& \frac{1}{W} \Big((\tau_j^{\infty(i)} - {\tau_{j-\widehat{W}+2}^{\infty}}^{(i)} + w_{j-\widehat{W}+2}^{(i)}) + \ldots \\
& + (\tau_j^{\infty(i)} - {\tau_{j+W-\widehat{W}-1}^{\infty}}^{(i)} - w_{j+\overline{W}-\widehat{W}}^{(i)}) - \sum_{l=j-\widehat{W}+1}^{j+\overline{W}-\widehat{W}} v_l^{(i)} \Big) \qquad (60)
\end{aligned}$$

where the first equality is by the definition of $\tau_{j,\overline{W},\widehat{W}}^{(i)}$, (59) is due to $\boldsymbol{v}_k = \frac{1}{n}\sum_{i=1}^n \Phi(x_{k,i}, y_{k,i})$, and (60) is due to $\boldsymbol{w}_k = \boldsymbol{\tau}_k^\infty - \boldsymbol{\tau}_{k-1}^\infty$. Using the definition of $\boldsymbol{w}_k$, we have that

$$\begin{aligned}
\tau_j^{\infty(i)} - \tau_{j,\overline{W},\widehat{W}}^{(i)} &= \frac{1}{W} \left(\sum_{l=j-\widehat{W}+2}^{j} w_l^{(i)} + \ldots - \sum_{l=j+1}^{j+\overline{W}-\widehat{W}} w_l^{(i)} - \sum_{l=j-\widehat{W}+1}^{j+\overline{W}-\widehat{W}} v_l^{(i)} \right) \\
&= \frac{1}{\overline{W}} \left(\sum_{l=j-\widehat{W}+2}^{j} (i - j + \widehat{W} - 1) w_l^{(i)} - \sum_{l=j+1}^{j+\overline{W}-\widehat{W}} (j + \overline{W} - \widehat{W} + 1 - i) w_l^{(i)} - \sum_{l=j-\widehat{W}+1}^{j+\overline{W}-\widehat{W}} v_l^{(i)} \right).
\end{aligned}$$

The sub-Gaussian parameter of the error in the mean vector estimate is given by

$$\begin{aligned}
(\rho_{j,\overline{W},\widehat{W}}^{(i)})^2 \leq& \frac{1}{\overline{W}^2} \sum_{l=j-\widehat{W}+2}^{j} (i - j + \widehat{W} - 1)^2 \sigma(w_l^{(i)})^2 \\
& + \frac{1}{\overline{W}^2} \sum_{l=j+1}^{j+\overline{W}-\widehat{W}} (j + \overline{W} - \widehat{W} + 1 - i)^2 \sigma(w_l^{(i)})^2 + \frac{1}{\overline{W}^2} \sum_{l=j-\widehat{W}+1}^{j+\overline{W}-\widehat{W}} \frac{\sigma(\Phi_l^{(i)})^2}{n_l}.
\end{aligned}$$

Let κ and n_l be such that $\sigma(w_l^{(i)}) \preceq \kappa\sqrt{d_l}, \sigma(\Phi_l^{(i)}) \preceq \kappa\sigma_l$, and $n_l \geq n$ for any l. Then, we have that

$$\begin{aligned}
(\rho^{(i)}_{j,\overline{W},\widehat{W}})^2 \leq & \frac{1}{\overline{W}^2} \sum_{l=j-\widehat{W}+2}^{j} (i-j+\widehat{W}-1)^2\kappa^2 d_l^{(i)} \\
& + \frac{1}{\overline{W}^2} \sum_{l=j+1}^{j+\overline{W}-\widehat{W}} (j+\overline{W}-\widehat{W}+1-i)^2\kappa^2 d_l^{(i)} + \frac{1}{\overline{W}^2} \sum_{l=j-\widehat{W}+1}^{j+\overline{W}-\widehat{W}} \frac{\kappa^2\sigma_l^{2(i)}}{n} \\
\leq & \frac{\widehat{W}(\widehat{W}-1)(2\widehat{W}-1)}{6\overline{W}^2} d \\
& + \frac{(2\widehat{W}^2+2\overline{W}^2+3\overline{W}+1-\widehat{W}(4\overline{W}+3))(\overline{W}-\widehat{W})}{6\overline{W}^2} d + \frac{1}{\overline{W}n} \\
= & \frac{6\widehat{W}^2+2\overline{W}^2+3\overline{W}+1-6\widehat{W}\overline{W}-6\widehat{W}}{6\overline{W}} d + \frac{1}{\overline{W}n}
\end{aligned}$$

The inequality in (56) is obtained using the union bound together with the Chernoff bound (concentration inequality) for the random variables $\tau_j^{\infty(i)} - \tau^{(i)}_{j,\overline{W},\widehat{W}}$ that are sub-Gaussian with parameter $\rho^{(i)}_{j,\overline{W},\widehat{W}}$ and because $n_j \geq n$, $\|\boldsymbol{\sigma}_j^2\|_\infty \leq 1$, and $\|\boldsymbol{d}_j\|_\infty \leq d$. ■

Appendix G. Optimization

Algorithm 5 details the implementation of the optimization step of the proposed methodology.

Algorithm 5 Optimization

Input: $\boldsymbol{\tau}$, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and $\mathcal{X}$
Output: $\boldsymbol{\mu}, R(\mathcal{U})$
for $x \in \mathcal{X}, \mathcal{C} \subseteq \mathcal{Y}, \mathcal{C} \neq \emptyset$ **do**
  $\mathbf{F}, \mathbf{h} \leftarrow$ append rows $\sum_{y\in\mathcal{C}} \Phi(x,y)^\top/|\mathcal{C}|, 1/|\mathcal{C}|$ to $\mathbf{F}, \mathbf{h}$
$\boldsymbol{\mu}(1) \leftarrow \boldsymbol{\mu}$, $\bar{\boldsymbol{\mu}}(1) \leftarrow \boldsymbol{\mu}$
for $l = 1, 2, \ldots, K$ **do**
  $a_l \leftarrow 1/(l+1)^{3/2}$, $\theta_l \leftarrow 2/(l+1)$, $\theta_{l+1} \leftarrow 2/(l+2)$
  $\mathbf{f}_i^\top \leftarrow$ row of $\mathbf{F}$ such that $\mathbf{f}_i^\top \boldsymbol{\mu}(l) - h^{(i)} = \max\{\mathbf{F}\boldsymbol{\mu}(l) - \mathbf{h}\}$
  $\bar{\boldsymbol{\mu}}(l+1) \leftarrow \boldsymbol{\mu}(l) + a_l\,(\boldsymbol{\tau} - \mathbf{f}_i - \boldsymbol{\lambda}\text{sign}(\boldsymbol{\mu}(l)))$
  $\boldsymbol{\mu}(l+1) \leftarrow \bar{\boldsymbol{\mu}}(l+1) + \theta_{l+1}(\theta_l^{-1} - 1)\,(\boldsymbol{\mu}(l) - \bar{\boldsymbol{\mu}}(l))$
$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}(K+1)$
$R(\mathcal{U}) \leftarrow 1 - \boldsymbol{\tau}^\top\boldsymbol{\mu} + \max\{\mathbf{F}\boldsymbol{\mu} - \mathbf{h}\} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}|$

Table 4: Dataset characteristics.

Dataset	Type	Samples	$\lvert\mathcal{Y}\rvert$	Tasks	Reference
Rotating hyperplane	Synthetic	30,000	2	100	(Pavlidis et al., 2011)
BAF	Tabular	1,000,000	2	3,333	(Jesus et al., 2022)
Elec2	Tabular	45,312	2	151	(Bifet and Gavalda, 2007)
Airlines	Tabular	539,383	2	1,797	(Bifet et al., 2010)
USPS	Tabular	2,930	2	9	(Dekel et al., 2008)
Spam	Tabular	6,213	2	20	(Sethi and Kantardzic, 2017)
Power supply	Tabular	29,928	2	99	(Tahmasbi et al., 2021)
Yearbook	Images	37,921	2	10	(Ginosar et al., 2015)
ImageNet Noise	Images	12,000	2	10	(Mai et al., 2022)
DomainNet	Images	6,256	4	6	(Peng et al., 2019)
UTKFace	Images	23,500	2	94	(Zhang et al., 2017)
Rotated MNIST	Images	70,000	2	60	`http://yann.lecun.com/exdb/mnist/`
Rotated MNIST i.i.d.	Images	70,000	2	60	`http://yann.lecun.com/exdb/mnist/`
CLEAR	Images	10,490	3	10	(Lin et al., 2021)

Appendix H. Extensions of the proposed methodology applied to CL

This section shows how the proposed methodology applied to CL can be extended to situations in which a new sample set can correspond with a precedingly learned task.

A sample set D corresponding with a t-th task for $t \in \{1, 2, \ldots, k\}$ can arrive at any time step. Since the t-th task is a precedingly learned task, we first update the sample average and MSE vector $\boldsymbol{\tau}_t$ and $\boldsymbol{s}_t$ given by (6) of the t-th task using the new sample set; secondly, we update mean and MSE vectors $\boldsymbol{\tau}_j^j$ and $\boldsymbol{s}_j^j$ as in (8)-(9) for each j-th task with $j \in \{t, t+1, \ldots, k\}$; and then, we update mean and MSE vectors $\boldsymbol{\tau}_j^k$ and $\boldsymbol{s}_j^k$ as in (14)-(15) for each j-th task with $j \in \{1, 2, \ldots, k\}$. Such mean and MSE vectors coincide with the mean and MSE vectors obtained in Section 4 using all samples corresponding with the t-th task at time step t. Then, taking $\boldsymbol{\sigma}_j^2 = \mathbb{V}\mathrm{ar}_{\mathrm{p}_j}\{\Phi(x, y)\}$ and $\boldsymbol{d}_j = \mathbb{E}\{\boldsymbol{w}_j^2\}$, the mean estimate $\boldsymbol{\tau}_j^k$ is the unbiased linear estimator that has the minimum MSE, and the updated $\boldsymbol{s}_j^k$ is its MSE (see Theorem 2).

Appendix I. Additional numerical results and implementation details

In this section we describe the datasets used for the numerical results in Section 8, we provide further details for the numerical experimentations carried out, and include several additional results. Specifically, in the first set of additional results, we show the reliability of the presented performance guarantees; in the second set of additional results, we show the performance improvement leveraging information from preceding and succeeding tasks; and in the third set of additional results, we show the classification error and the running time for different hyper-parameter values. In addition, the code of the proposed methods is available on the web `https://github.com/MachineLearningBCAM/Supervised-learning-evolving-task-JMLR-2024`.

The datasets used in Section 8 are publicly available Ginosar et al. (2015); Zhang et al. (2017); Peng et al. (2019); Lin et al. (2021), and `http://yann.lecun.com/exdb/mnist/`. The summary of these datasets is provided in Table 4 that shows the number of classes,

the number of samples, and the number of tasks. In the following, we further describe the tasks and the time-dependency of each dataset used.

- The "rotating hyperplane" dataset that has been often used as benchmark for evolving environments (Pavlidis et al., 2011). In such dataset, a hyperplane in D-dimensional space $\sum_{i=1}^{D} w_i x_i = 0$ rotates between consecutive tasks. Instances for $\sum_{i=1}^{D} w_i x_i \geq 0$ correspond with class 1 and instances for $\sum_{i=1}^{D} w_i x_i < 0$ correspond with class 2. In the following, we use multiple rotation angles of the hyperplane to evaluate the error in terms of the change in the distribution.
- The "BAF" dataset contains real-world bank account fraud detection information and the goal is to predict fraud or not fraud.
- The "Elec2" dataset contains half-hourly energy-related information including the day of week, the time stamp, the New South Wales electricity demand, the Victoria electricity demand, and the scheduled electricity transfer between states; and the goal is to predict energy price increases or decreases in New South Wales relative to a moving average of the last 24 hours.
- The "Airlines" dataset contains records of flight schedules and the goal is to predict if a flight is delayed or not.
- The "USPS" dataset contains numeric data obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service.
- The "Spam" dataset contains emails and the task is to predict if a email is malicious spam email or legitimate email.
- The "Yearbook" dataset contains portraits' photographs over time and the goal is to predict males and females. Each task corresponds to portraits from one decade from 1905 to 2013.
- The "ImageNet noise" dataset contains images with increasing noise over tasks and the goal is to predict if an image is a bird or a snake. The sequence of tasks corresponds to the noise factors $[0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6]$ (Mai et al., 2022).
- The "DomainNet" dataset contains six different domains with decreasing realism and the goal is to predict if an image is an airplane, bus, ambulance, or police car. The sequence of tasks corresponds to the six domains: real, painting, infograph, clipart, sketch, and quickdraw.
- The "UTKFaces" dataset contains face images in the wild with increasing age and the goal is to predict males and females. The sequence of tasks corresponds to face images with different ages from 0 to 116 years.
- The "Rotated MNIST" dataset contains rotated images with increasing angles over tasks and the goal is to predict if the number in an image is greater than 5 or not. Each j-th task corresponds to a rotation angle randomly selected from $\left[\frac{180(j-1)}{k}, \frac{180j}{k}\right]$ degrees where $j \in \{1, 2, ..., k\}$ and k is the number of tasks.
- The "Rotated MNIST" dataset contains rotated images with increasing angles over tasks and the goal is to predict if the number in an image is greater than 5 or not. Each j-th task corresponds to a rotation angle randomly selected from $[0, 360)$ degrees.

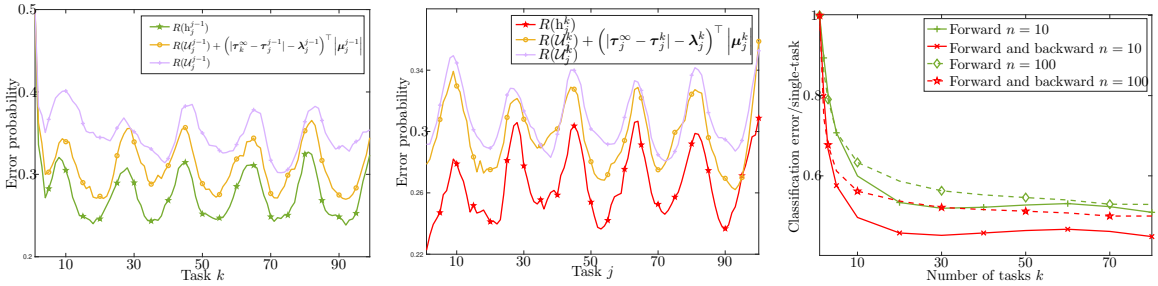
- The "CLEAR" dataset contains images with a natural temporal evolution of visual concepts in the real world and the goal is to predict if an image is soccer, hockey, or racing. Each task corresponds to one year from 2004 to 2014.

In Section 8, we compare the results of IMRC methods with 6 state-of-the-art-techniques Lopez-Paz and Ranzato (2017); Riemer et al. (2018); Kirkpatrick et al. (2017); Zhao et al. (2020); Tahmasbi et al. (2021); Brzezinski and Stefanowski (2013). In the following, we briefly describe each method used.

- Condor method Zhao et al. (2020) is a technique developed for concept drift adaptation. The method provided by Zhao et al. is an ensemble method that adapts to evolving tasks by learning weighting the models in the ensemble at each time step.
- DriftSurf method Tahmasbi et al. (2021) is a technique developed for concept drift adaptation. The method provided by Tahmasbi et al. adapts to evolving tasks by using a drift detection method. Such method allows to restart a new model when a change in the distribution is detected.
- AUE method Brzezinski and Stefanowski (2013) is a technique developed for concept drift adaptation. The method provided by Brzezinski & Stefanowski is an ensemble method that adapts to evolving tasks by incrementally updating all classifiers in the ensemble and weighting them with non-linear error functions.
- GEM method Lopez-Paz and Ranzato (2017) is a technique developed for continual learning. The method provided by Lopez-Paz & Ranzato learns each new task using a stochastic gradient descent with inequality constraints given by the losses of preceding tasks. Such constraints avoid the increase of the loss of each preceding tasks.
- MER method Riemer et al. (2018) is a technique developed for continual learning. The method provided by Riemer et al. learns each new task using sample sample sets that include random samples of preceding tasks. Such samples of preceding tasks are stored in a memory buffer.
- EWC method Kirkpatrick et al. (2017) is a technique developed for continual learning. The method provided by Kirkpatrick et al. learns each new task regularizing the loss with regularization parameters given by the Fisher information.

In the first set of additional results we further illustrate the reliability of the bounds using the synthetic dataset described in Section 8. Specifically, we extend such results for the proposed methodology with $n = 100$ samples per task completing those in the main paper with $n = 10$ samples per task. Figures 12a and 12b show, for each task, the averaged instantaneous bounds of probabilities of error in comparison with the true probabilities of error $R(\mathrm{h}_j^{j-1})$ and $R(\mathrm{h}_j^k)$, respectively. Such figures show that the bounds $R(\mathcal{U}_j^{j-1})$ and $R(\mathcal{U}_j^k)$ obtained at learning can offer accurate estimates for the probability of error of each task.

In the second set of additional results, we analyze the contribution of forward and backward learning to the final performance of the proposed methodology applied to MTL and CL using "Yearbook" dataset. In particular, we show the relationship among classification error, number of tasks, and sample size for single-task, forward, and forward and backward learning. These numerical results are obtained computing the classification error over all the



(a) Bounds for error probabilities using information from preceding tasks (SCD and MDA).

(b) Bounds for error probabilities using information from preceding, the target, and succeeding tasks (CL and MTL).

(c) Classification error per number of tasks in "Yearbook" dataset.

Figure 12: Bounds for error probabilities with $n = 100$ samples per task and forward and backward learning can sharply boost performance and ESS as tasks arrive.

Table 5: Classification error of the proposed method using forward and backward learning varying W and b.

Hyper-parameter	$W=2$		$W=4$		$W=6$		$b=1$		$b=2$		$b=3$		$b=4$		$b=5$	
Sample size n	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Yearbook	.13	.08	.13	.09	.13	.09	.14	.10	.09	.14	.13	.08	.13	.08	.13	.08
ImageNet noise	.15	.09	.15	.09	.15	.09	.15	.09	.15	.09	.15	.09	.15	.08	.15	.08
DomainNet	.34	.28	.32	.27	.33	.28	.36	.29	.35	.28	.34	.28	.34	.28	.34	.28
UTKFaces	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10	.10
Rotated MNIST	.36	.21	.35	.21	.36	.21	.36	.22	.36	.22	.36	.21	.36	.21	.36	.21
CLEAR	.09	.05	.09	.05	.09	.06	.10	.05	.09	.05	.09	.05	.09	.05	.08	.05

sequences of consecutive tasks of length k in the dataset. Then, we repeat such experiment 10 times with randomly chosen training sets of size n. Figures 12c shows the classification error of the proposed method divided by the classification error of single-task learning for different number of tasks with $n = 10$ and $n = 100$ sample sizes using "Yearbook" dataset. Such figure shows that forward and backward learning can significantly improve the performance increasing the number of tasks.

In the third set of additional results, we assess the change in classification error and the running time varying the hyper-parameters. Table 5 shows the classification error of the proposed methodology using forward and backward learning varying the values of hyper-parameter for the window size W and the number of backward steps b, completing those in the paper that show the results for $W = 2$ and $b = 3$. As shown in the table, the proposed methodology do not require a careful fine-tuning of hyper-parameters and similar performances are obtained by using different values. In addition, Table 6 shows the mean running time per task in seconds for $b = 1, 2, \ldots, 5$ backward steps in comparison with the state-of-the-art techniques that learn a sequence of tasks. Such table shows that the methods proposed for backward learning do not require a significant increase in complexity,

Table 6: Running time of the proposed method using forward and backward learning in comparison with the state-of-the-art-techniques.

Dataset	GEM		MER		EWC		$b = 1$		$b = 2$		$b = 3$		$b = 4$		$b = 5$	
Sample size n	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Yearbook	0.10	0.48	0.17	3.73	0.36	3.03	0.10	0.32	0.11	0.40	0.13	0.49	0.17	0.58	0.18	0.66
ImageNet noise	0.01	0.04	0.08	1.05	0.03	0.25	0.26	0.49	0.26	0.53	0.28	0.56	0.30	0.59	0.44	0.60
DomainNet	0.01	0.02	0.07	0.90	0.02	0.16	0.52	8.46	0.54	8.98	0.54	9.51	0.56	9.70	0.57	9.92
UTKFaces	0.31	0.18	0.13	3.46	0.25	2.25	0.11	0.35	0.12	0.40	0.13	0.49	0.16	0.57	0.19	0.66
Rotated MNIST	0.18	1.09	0.21	4.09	0.59	5.30	0.14	0.47	0.17	0.60	0.21	0.74	0.25	0.88	0.29	1.01
CLEAR	0.01	0.03	0.07	1.06	0.03	0.25	0.24	1.31	0.25	1.41	0.26	1.47	0.27	1.60	0.36	1.69

and the running time of the proposed method is similar to that of other state-of-the-art methods.

References

Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In *IEEE Power & Energy Society general meeting*, pages 1–5, 2017.

Verónica Álvarez, Santiago Mazuelas, and Jose A Lozano. Minimax classification under concept drift with multidimensional adaptation and performance guarantees. In *International Conference on Machine Learning*, volume 162, pages 486–499, 2022.

Verónica Álvarez, Santiago Mazuelas, and Jose A Lozano. Minimax forward and backward learning of evolving tasks with performance guarantees. In *Advances in Neural Information Processing Systems*, volume 36, pages 65678–65702, 2023.

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *International Conference on Machine Learning*, volume 80, pages 205–214, 2018.

Brian DO Anderson and John B Moore. *Optimal Filtering*. Courier Corporation, 2012.

Guangji Bai, Chen Ling, and Liang Zhao. Temporal domain generalization with drift-aware dynamic neural networks. In *International Conference on Learning Representations*, 2022.

Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, volume 97, pages 424–433, 2019.

Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: Theory algorithms and software*. John Wiley & Sons, 2004.

Peter L Bartlett. Learning with a slowly changing distribution. In *Workshop on Computational Learning Theory*, pages 243–252, 1992.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

Samuel J Bell and Neil D Lawrence. The effect of task ordering in continual learning. *arXiv preprint arXiv:2205.13323*, 2022.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, pages 41–48, 2009.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *SIAM International Conference on Data Mining*, pages 443–448, 2007.

Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. MOA: Massive online analysis, a framework for stream classification and clustering. In *Workshop on Applications of Pattern Analysis*, pages 44–50, 2010.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Kartheek Bondugula, Verónica Álvarez, Jose I Segovia-Martín, Aritz Pérez, and Santiago Mazuelas. MRCpy: A library for minimax risk classifiers. *arXiv preprint arXiv:2108.01952*, 2021.

Dariusz Brzezinski and Jerzy Stefanowski. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):81–94, 2013.

Qi Chen, Changjian Shui, Ligong Han, and Mario Marchand. On the stability-plasticity dilemma in continual meta-learning: Theory and algorithm. In *Advances in Neural Information Processing Systems*, volume 36, pages 27414–27468, 2023.

Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

Paul SP Cowpertwait and Andrew V Metcalfe. *Introductory time series with R*. Springer Science & Business Media, 2009.

Ofer Dekel, Shai Shalev-Shwartz, and Yoram Singer. The forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing*, 37(5):1342–1372, 2008.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, volume 97, pages 1566–1575, 2019.

Giulia Denevi, Dimitris Stamos, Carlo Ciliberto, and Massimiliano Pontil. Online-within-online meta-learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 1–11, 2019.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019.

Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011.

Theodoras Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, 2004.

Itay Evron, Edward Moroshko, Rachel Ward, Nathan Srebro, and Daniel Soudry. How catastrophic can catastrophic forgetting be in linear regression? In *Conference on Learning Theory*, volume 178, pages 4028–4079, 2022.

Alec Farid and Anirudha Majumdar. Generalization bounds for meta-learning via PAC-Bayes and uniform stability. In *Advances in Neural Information Processing Systems*, volume 34, pages 2173–2186, 2021.

Farzan Farnia and David Tse. A minimax approach to supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 4240–4248.

Rizal Fathony, Anqi Liu, Kaiser Asif, and Brian D. Ziebart. Adversarial multiclass classification: A risk minimization perspective. In *Advances in Neural Information Processing Systems*, volume 29, pages 559–567, 2016.

Filippo Fedeli, Alberto Maria Metelli, Francesco Trovò, and Marcello Restelli. IWDA: Importance weighting for drift adaptation in streaming supervised learning problems. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):6813–6823, 2023.

Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *IEEE International Conference on Computer Vision Workshops*, pages 1–7, 2015.

Husheng Guo, Yang Zhang, and Wenjian Wang. Dynamical targeted ensemble learning for streaming data with concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Steve Hanneke and Liu Yang. Statistical learning under nonstationary mixing processes. In *International Conference on Artificial Intelligence and Statistics*, volume 89, pages 1678–1686, 2019.

Christian Henning, Maria Cervera, Francesco D'Angelo, Johannes Von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F Grewe, and João Sacramento. Posterior meta-replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 14135–14149, 2021.

Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. In *Advances in Neural Information Processing Systems*, volume 36, pages 48510–48533, 2024.

Julio Hurtado, Alain Raymond, and Alvaro Soto. Optimizing reusable knowledge for continual learning via metalearning. In *Advances in Neural Information Processing Systems*, volume 34, pages 14150–14162, 2021.

Sérgio Jesus, José Pombal, Duarte Alves, André Cruz, Pedro Saleiro, Rita P. Ribeiro, João Gama, and Pedro Bizarro. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ML evaluation. In *Advances in Neural Information Processing Systems*, volume 35, pages 33563–33575, 2022.

Olav Kallenberg. *Random measures, theory and applications*. Springer, 2017.

Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, volume 32, pages 5917–5928, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *National Academy of Sciences*, volume 114, pages 3521–3526, 2017.

Parker Knight and Rui Duan. Multi-task learning with summary statistics. In *Advances in Neural Information Processing Systems*, volume 36, pages 54020–54031, 2024.

Guy Lebanon and John Lafferty. Boosting and maximum likelihood for exponential models. In *Advances in neural information processing systems*, volume 14, 2001.

Keqiuyin Li, Jie Lu, Hua Zuo, and Guangquan Zhang. Multi-Source contribution learning for domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5293–5307, 2022.

Zijian Li, Ruichu Cai, Guangyi Chen, Boyang Sun, Zhifeng Hao, and Kun Zhang. Subspace identification for multi-source domain adaptation. In *Advances in Neural Information Processing Systems*, volume 36, pages 34504–34518, 2023.

Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 12037–12047, 2020.

Zhiqiu Lin, Jia Shi, Deepak Pathak, and Deva Ramanan. The CLEAR benchmark: Continual learning on real-world imagery. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Philip M Long. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354, 1999.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 6470–6479, 2017.

Yang Lu, Yiu-Ming Cheung, and Yuan Yan Tang. Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8):2764–2778, 2019.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, volume 22, pages 1041–1048, 2009.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Santiago Mazuelas, Andrea Zanoni, and Aritz Perez. Minimax classification with 0-1 loss and performance guarantees. In *Advances in Neural Information Processing Systems*, volume 33, pages 302–312, 2020.

Santiago Mazuelas, Yuan Shen, and Aritz Pérez. Generalized maximum entropy for supervised classification. *IEEE Transactions on Information Theory*, 68(4):2530–2550, 2022.

Santiago Mazuelas, Mauricio Romero, and Peter Grünwald. Minimax risk classifiers with 0-1 loss. *Journal of Machine Learning Research*, 24(208):1–48, 2023.

Alessio Mazzetto and Eli Upfal. An adaptive algorithm for learning with unknown distribution drift. In *Advances in Neural Information Processing Systems*, volume 36, pages 10068–10087, 2023.

Mehryar Mohri and Andres Munoz Medina. New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pages 124–138, 2012.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

John F Muth. Optimal properties of exponentially weighted forecasts. *Journal of the American Statistical Association*, 55(290):299–306, 1960.

Yu Nesterov and Vladimir Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, 2015.

Brian J Odelson, Murali R Rajamani, and James B Rawlings. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006.

Francesco Orabona, Joseph Keshet, and Barbara Caputo. The projectron: A bounded kernel-based perceptron. In *International Conference on Machine Learning*, pages 720–727, 2008.

NG Pavlidis, DK Tasoulis, NM Adams, and DJ Hand. λ-perceptron: An adaptive classifier for data streams. *Pattern Recognition*, 1(44):78–96, 2011.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

Anastasia Pentina and Christoph H Lampert. Lifelong learning with non-iid tasks. In *Advances in Neural Information Processing Systems*, volume 28, pages 1540–1548, 2015.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. Curriculum learning of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5492–5500, 2015.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, 2007.

Henry W.J. Reeve, Timothy I. Cannings, and Richard J. Samworth. Adaptive transfer learning. *Annals of Statistics*, 49(6):3618–3649, 2021.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018.

Paul Ruvolo and Eric Eaton. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, volume 28, pages 507–515, 2013.

Tegjyot Singh Sethi and Mehmed Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77–99, 2017.

Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

Yanning Shen, Tianyi Chen, and Georgios B Giannakis. Random feature-based online multi-kernel learning in environments with unknown dynamics. *The Journal of Machine Learning Research*, 20(1):773–808, 2019.

Petar Stojanov, Mingming Gong, Jaime G. Carbonell, and Kun Zhang. Data-driven approach to multiple-source domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, volume 89, pages 3487–3496, 2020.

Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, and Phillip B Gibbons. DriftSurf: Stable-state/reactive-state learning under concept drift. In *International Conference on Machine Learning*, volume 139, pages 10054–10064, 2021.

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. The strength of Nesterov's extrapolation in the individual convergence of nonsmooth optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2557–2568, 2019.

Jeevan Thapa and Rui Li. Bayesian adaptation of network depth and width for continual learning. In *International Conference on Machine Learning*, volume 235, pages 48038–48061, 2024.

Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. In *Advances in Neural Information Processing Systems*, volume 33, pages 7852–7862, 2020.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, volume 80, pages 5238–5246, 2018.

Mixue Xie, Shuang Li, Longhui Yuan, Chi Liu, and Zehui Dai. Evolving standardization for continual domain generalization over temporal drift. In *Advances in Neural Information Processing Systems*, volume 36, pages 21983–22002, 2023.

Yuxiang Yang, Lu Wen, Pinxian Zeng, Binyu Yan, and Yan Wang. Dane: A dual-level alignment network with ensemble learning for multi-source domain adaptation. *IEEE Transactions on Instrumentation and Measurement*, 2024.

Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: A causal view. In *National Conference on Artificial Intelligence*, volume 4, pages 3150–3157, 2015.

Liangwei Zhang, Jing Lin, and Ramin Karim. Sliding window-based fault detection from high-dimensional data streams. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(2):289–303, 2016.

Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5810–5818, 2017.

Peng Zhao, Le-Wen Cai, and Zhi-Hua Zhou. Handling concept drift via model reuse. *Machine Learning*, 109(3):533–568, 2020.