Unsupervised Transfer Aided Lifelong Regression For Learning New Tasks without Target Output

Tong Liu, *Member, IEEE*, Xulong Wang, Po Yang, *Senior Member, IEEE*, Sheng Chen, *Life Fellow, IEEE*, and Chris J. Harris

Abstract—As an emerging learning paradigm, lifelong learning solves multiple consecutive tasks based upon previously accumulated knowledge. When facing with a new task, existing lifelong learning approaches need both input and desired output data to construct task models before knowledge transfer can succeed. However, labeling each task requires extensive labors and time, which can be prohibitive for real-world lifelong regression problems. To reduce this burden, we propose to incorporate unsupervised feature into lifelong regression via coupled dictionary learning, enabling to learn new tasks without target output data. Specifically, the input data for each task is encoded as unsupervised feature while both input and output data are used to construct task predictor. The unsupervised feature is linked with task predictor through two dictionaries that are coupled by a joint sparse representation. Because of the learned coupling between the two spaces, the task predictor for the new coming task can be recovered given only the input data. We further incorporate active task selection into this framework, enabling actively choosing tasks to learn in a task-efficient manner. Three case studies are used to evaluate the effectiveness of our method, in comparison with existing lifelong learning approaches. Results show that our method is able to accurately predict new tasks through unsupervised transfer, eliminating the need to label tasks before constructing the predictor.

Index Terms—Lifelong regression, unsupervised feature, coupled dictionary learning, knowledge transfer, active task selection

1 Introduction

Transfer learning and multi-task learning methods reduce the amount of experience needed to learn individual tasks by reusing knowledge from other related tasks [1], [2]. This knowledge transfer significantly improves learning efficiency and modeling performance, as compared to learning tasks in isolation following the traditional machine learning paradigm. Transfer learning methods transfer knowledge from source tasks to help learning new task [3], [4], [5], [6], which however fail to optimize the performance over all the tasks, while multi-task learning methods jointly learn all the observed tasks by sharing knowledge [7], [8], [9], [10], [11], but they cannot learn new unseen task.

T. Liu, X. Wang and P. Yang are with Department of Computer Science, University of Sheffield, Sheffield S1 4DP, UK (E-mail: t.liu@sheffield.ac.uk, xl.wang@sheffield.ac.uk, po.yang@sheffield.ac.uk).
S. Chen and C.J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK (E-mails: sqc@ecs.soton.ac.uk, chrisharris57@msn.com). S. Chen is also with Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266100 China.

To combat both limitations of transfer learning and multi-task learning, lifelong learning as a new research paradigm, was proposed to learn consecutive new task based upon previously built knowledge as well as to automatically update the past knowledge accumulated from the past encountered tasks upon the learning of the new task [12]. This technique is particularly suitable to solve some scenarios with multiple consecutive tasks over long-time scales [13], [14], [15], [16], [17], such as applications of sentiment classification, robotic control, natural language processing and diseases modeling [12]. It is widely understood that a fundamental principle for better learning is to incorporate available prior knowledge in the learning process [18], [19], [20]. Anything learned from a previous learning task can be regarded as a piece of knowledge, and this knowledge can be reserved to help future learning. This is the core idea of lifelong learning. More specifically, lifelong learning maintains a knowledge base which stores the knowledge learned in the previous learning tasks. When learning a new task, the knowledge accumulated provides available prior knowledge for the current learning task. New knowledge acquired in the current learning process is then used to update the knowledge base. For example, a student who has never studied psychology before wants to study it. This can be regarded as a new task. The student has the past education of learning philosophy, literature, and other subjects. Knowledge the student gained in these past learning tasks are stored in the student's knowledge base, i.e., the student's brain, and these 'prior' knowledge can help the student in learning the new subject psychology. New knowledge that the student will gain in studying psychology in turn will enhance the student's knowledge base. It can be seen that lifelong learning imitates *human learning*.

Among lifelong learning community, the efficient lifelong learning algorithm (ELLA) framework is one of the most popular approaches [21], [22]. The ELLA factorizes learned task models into a shared latent dictionary as the knowledge base to facilitate knowledge transfer as tasks arrive consecutively. When new task arrives, the ELLA transfers knowledge through the shared dictionary to learn new model, and refines the dictionary with the knowledge learned from current task. By updating the dictionary over time, newly acquired knowledge is incorporated into the knowledge base, thereby improving previously learned

models' performance. The ELLA framework was first created for regression and classification, and it was later developed for policy gradient reinforcement learning (PG-ELLA) [23], [24], [25], [26], [27]. By replacing the task model with policy, the PG-ELLA enables to learn decision making tasks consecutively, transferring knowledge to accelerate learning new policy. The work of [28] further extended ELLA from a single agent to a network of agents, and proposed the collective lifelong learning algorithm to enable sharing knowledge in a distributed manner for multiple agents. Different from ELLA, another typical lifelong learning model is deep neural network, where *catastrophic forgetting* is the key issue in its continuous learning process. Inspired by synaptic consolidation in human brains, elastic weight consolidation (EWC) was proposed to combat the catastrophic forgetting problem in deep networks by restricting the change of important neural network weights of previous tasks when learning new task [29]. The EWC has been successfully applied to object detection [30], neural machine translation [31], image generation [32], and so on.

One notable issue is that lifelong learning in the above problem setting is a passive process, in which the leaner must learn every encountering task and it also has no control over the learning order for tasks. In some situations, the agent may have a pool of candidate tasks to learn, and it can intelligently choose the next task to learn in order to maximize the overall performance. With this goal, the work [33] incorporates active curriculum selection strategy into ELLA, enabling the learner to choose tasks in certain order so as to maximize future learning performance using as few tasks as possible. The authors of [33] proposed several active task selection mechanisms for selecting the next best task, and demonstrated that the diversity heuristic method (ELLA-diver) has superior efficiency to build knowledge library over other methods. Considering a different active task selection, the work [34] integrates outlier detection into lifelong learning so as to selectively learn the next task based on the tasks' importance. By either choosing tasks in certain order or selectively choosing important tasks to learn, both these methods learn in a task-efficient manner, which is particularly important when dealing with massive tasks.

While above lifelong learning methods demonstrate outstanding performance in many applications, one important preliminary need is to gather sufficient both input and desired output data for the new coming task and characterize task relationships. For lifelong regression problems, desired output is also referred to as target output. When new task arrives, the learner requires sufficient training data of both input and target output to identify task relationships before bootstrapping a model via transfer. This need for desired output data imposes a serious challenge for practical lifelong regression problems, as persistent manual data annotation for every new coming task is time-consuming and economically costly, and often the leaner is expected to learn new task rapidly without the delay to wait for labeling task. To overcome this restriction, one famous early work of [35] incorporates high-level task descriptors into lifelong reinforcement learning (TaDeLL), and use both task descriptors and training data to model inter-task relationships. The results of [35] show that using task descriptors improves the performance of learned policies, and moreover, it enables predicting policy for new task without training data via zero-shot transfer given only task descriptors. TaDeLL was further extended for regression problem in [36], where task model can be predicted given only descriptors for new task. This 'learning without training data' seems very appealing. But the fact is that TaDeLL requires domain-specific task descriptors that must characterize the underlying dynamics of data in individual tasks well. For instance, the work [36] used the engineering system's basic parameters, such as length, mass, damping constant, etc., as task descriptors for the engineering system considered, because these parameters define the system's underlying dynamics and have a close relation to the data characteristics. However, for most real-world tasks, seeking such appropriate and unified descriptors to identify different tasks requires in-depth cross-domain knowledge, which is generally impossible to achieve. Moreover, inaccurate task descriptors will lead to wrong task model and degrade the achievable learning performance considerably. Hence, TaDell is not generally applicable to many applications.

Consequently, to our best knowledge, how to efficiently utilize large amount of unlabeled data in characterizing and learning each consecutive task with improved performance is an important challenge for the lifelong learning community. This motivates our current work to develop an effective lifelong regression model that enables to learn new task without target output data, thus reducing the burden for labeling every coming task. We explore the use of input data to achieve unsupervised transfer for learning new task without desired output. Our approach to incorporate input information into lifelong regression is general, as it does not need domain-specific task descriptors that require human expert. Instead, we encode input data as feature vectors that identify each task and treat these unsupervised features as side information to augment task predictor on the individual tasks. Similar to [35], [36], [37], we use coupled dictionary learning to link the unsupervised feature space with the task predictor's parameter space, where the two spaces are linked through the two dictionaries that are coupled by the same sparse coding. Because of the learned coupling between the two spaces, the unsupervised features act as backup to the task predictor, enabling the learner to accurately construct predictors for the unseen tasks given only their unsupervised features. This capacity is very important in the online setting of lifelong regression process. It enables the agent to rapidly learn new tasks through unsupervised transfer from the previously learned tasks, without the need to first label the future tasks for collecting the target output data. To make our lifelong model learns in a task-efficient manner, we further incorporate active task selection into this framework. Three case studies, 1) school examination score prediction, 2) Parkinson disease symptom score prediction, and 3) Alzheimer disease progression modeling, are used to demonstrate the effectiveness of the proposed scheme, in comparison with existing lifelong learning approaches. Extensive experiments demonstrate that our method can accurately predict the new task using only input data via unsupervised transfer.

Notably, it should be emphasize that our proposed unsupervised transfer aided lifelong learning differs from the unsupervised transfer learning or domain adaptation. The

goal of unsupervised domain adaptation is to train a single model for a target domain or task with unlabeled data by transferring knowledge from a source task in which desired output data is accessible [38], [39], [40]. These methods usually consider only a single target task, and they fail to learn in a lifelong setting where multiple tasks are acquired sequentially over long-time scales. Unlike the traditional unsupervised domain adaptation methods that are only restricted to single-source single-target, the recently emerged multi-target domain adaptation are able to deal with multiple domains [41], [42], [43], [44]. But they still fail to learn in a continual manner. Another related learning paradigm is continual learning [45]. The continual learning aims to address the catastrophic forgetting problem in which the model is likely to forget the past learned tasks when encountering new tasks. Existing continual learning methods use either model regularization or experience replay to tackle catastrophic forgetting [29], [46]. By incorporating continual learning mechanism into unsupervised domain adaptation, the recent continual domain adaptation is most similar to our problem setting. In continual domain adaptation, the unlabeled or labeled target task data are received in streaming batches, and the model is continuously adapted with each batch of target data [47], [48], [49], [50], [51]. Note that the continual domain adaptation aims to learn adaptively to deal with domain shift when encountering new unseen tasks. By contrast, our method enables to learn new task adaptively while in the meantime optimize the performance of overall encountered tasks by updating the accumulated knowledge. A comparison of various learning paradigms is tabulated in Table. 1. Moreover, the continual domain adaptation methods only focus on object recognition or classification, and they are not applicable for regression learning [47], [48]. Although the existing lifelong learning approaches, such as ELLA [22] and TaDeLL [36] can be used for regression problem, they fail to learn and predict new consecutive tasks using solely unannotated data.

It is worth recapping that although TaDell [36] is the most similar to our method, its capacity of learning without data heavily depends on finding an appropriate task descriptor. As aforementioned, seeking such appropriate task descriptors typically requires in-depth expert knowledge, which are generally unavailable for most real-world applications. By contrast, our method is immune to this restriction, and it only requires unlabeled data that is easily to obtain for new tasks. Our method can be considered as an improvement over existing lifelong learning methods with the key idea of unsupervised transfer. Specifically, this paper provides the following contributions:

1) Based on coupled dictionary learning, we incorporate unsupervised features into lifelong learning that use a factorized representation of the learned knowledge to facilitate transfer and improve predictive performance.
2) Most importantly, we show that our proposed method is able to accurately modeling and predict new consecutive tasks using solely unannotated data through unsupervised transfer.
3) The proposed scheme is integrated with active task selection mechanism, which enables further improving learning efficiency when encountering massive tasks.
4) We analysis the method theoretically, and use three real-world datasets to validate its effectiveness.

The rest of this paper is organized as follows. Section 2 reviews the background on lifelong learning. Section 3 presents the proposed unsupervised transfer aided lifelong regression framework in detail. Section 4 summarizes our proposed algorithm with theoretical analysis. Section 5 evaluates the proposed method with three case studies. Section 5 concludes the paper with remarks about future works.

2 Lifelong Machine Learning

2.1 Problem Definition

In the lifelong regression setting, the learner faces multiple consecutive regression tasks $\left\{\mathbb{Z}^{(1)}, \mathbb{Z}^{(2)}, \cdots, \mathbb{Z}^{(T_{\max})}\right\}$, and must rapidly learn each new task by building upon its previous knowledge. Each regression task $\mathbb{Z}^{(t)} = \left\{f^{(t)}, \boldsymbol{x}^{(t)}, y^{(t)}\right\}$ is specified by a function mapping $f^{(t)} : \boldsymbol{x}^{(t)} \mapsto y^{(t)}$ from the input space $\boldsymbol{x}^{(t)} \in \mathbb{R}^d$ onto the output space $y^{(t)} \in \mathbb{R}$. At each time step t, the agent receives a batch of n_t labeled training data $\left(\boldsymbol{x}_i^{(t)}, y_i^{(t)}\right)_{i=1}^{n_t}$ for learning task t, where $\boldsymbol{x}_i^{(t)}$ and $y_i^{(t)}$ denote the ith training input sample and the associated desired output sample, respectively, for task t.

Let T denote the number of tasks that the agent has encountered so far. Its goal is to consecutively construct a set of task models or predictors $\left\{\widehat{f}^{(1)}, \cdots, \widehat{f}^{(T)}\right\}$ such that each $\widehat{f}^{(t)}$ approximates $f^{(t)}$ to make accurate prediction on new data, and new model $\widehat{f}^{(t)}$ can be acquired efficiently when the agent encountering new task t. Ideally, knowledge learned from previous tasks $\left\{\mathbb{Z}^{(1)}, \ldots, \mathbb{Z}^{(T-1)}\right\}$ should accelerate training and improve performance on new task $\mathbb{Z}^{(T)}$.

It can be seen that the lifelong learning is very different from existing learning frameworks. In the traditional learning framework, the learner has multiple batches of data generated by the same underlying process, and therefore the learner may use the multiple models identified from the previously encountered multiple data batches to predict the new batch of data, using for example, selective ensemble regression. Lifelong learning represents a much more general learning setting. Every task can represent a different batch of data, characterized by its own task definition and associated underlying data generation mechanism. Hence, it is necessary to construct a new task model for each

TABLE 1
A comparison of various learning paradigms.

	Multi-task learning	**Continual domain adaptation**	**Unsupervised domain adaptation**	**Our method**
Optimizing performance over	All tasks	Target tasks	Target tasks	All tasks
Learning tasks consecutively	No	Yes	No	Yes
Computational cost	High	Low	Low	Low
Labeled data for new tasks	Yes	Yes	No	No

coming new task, and the task data can be discarded after learning of the new model. On the other hand, all the tasks share some common characteristics or have the shared knowledge, which can be exploited to facilitate faster and better learning of the new task. This is the essence of the lifelong learning.

2.2 Efficient Lifelong Learning Algorithm

The ELLA [22] was developed to operate in this lifelong learning setting. To be specific, the ELLA learns and maintains a shared knowledge library $\boldsymbol{L} \in \mathbb{R}^{d\times k}$, which forms a basis for all task models and facilitates knowledge transfer between tasks. For each task t, the ELLA learns a model $\widehat{f}^{(t)}(\boldsymbol{x}) = \widehat{f}(\boldsymbol{x};\boldsymbol{\theta}^{(t)})$ that is parametrized by a d-dimensional task-specific parameter vector $\boldsymbol{\theta}^{(t)}$. This model parameter is a linear combination of the columns of $\boldsymbol{L}$ using the sparse coefficients $\boldsymbol{s}^{(t)} \in \mathbb{R}^k$ as $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$. The dictionary $\boldsymbol{L}$ stores chunks of knowledge that are shared for all the tasks, and the sparse code $\boldsymbol{s}^{(t)}$ extracts the relevant pieces of knowledge for a particular task t. Hence, this model parameter factorization enables effective knowledge transfer among tasks.

Given the training data $\left(\boldsymbol{x}_i^{(t)}, y_i^{(t)}\right)_{i=1}^{n_t}$ for each task t, the ELLA minimizes the following objective function:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right) + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right) + \lambda\left\|\boldsymbol{L}\right\|_{\mathrm{F}}^2, \tag{1}$$

where $\mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right) = \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{J}\left(y_i^{(t)} - \widehat{f}\left(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}\right)\right)$ with $\mathcal{J}(\bullet)$ being a squared-loss function for regression problem $\widehat{y}_i^{(t)} = \widehat{f}\left(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}\right)$, $\boldsymbol{S} = \left[\boldsymbol{s}^{(1)}\ \boldsymbol{s}^{(2)}\cdots\boldsymbol{s}^{(T)}\right]$ is the matrix consisting of all the sparse coefficient vectors, and the L_1 norm is used to control the sparsity of $\boldsymbol{s}^{(t)}$ with the regularization parameter μ, while $\|\bullet\|_{\mathrm{F}}$ is the Frobenius norm, which regularizes the complexity of dictionary $\boldsymbol{L}$ with the regularization parameter λ. This problem can be solved in a batch learning setting for off-line multi-task learning framework [52].

To solve it in a lifelong learning setting, the ELLA tasks a second-order Taylor expansion to approximate the objective around an estimate $\widehat{\boldsymbol{\theta}}^{(t)} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right) = \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{J}\left(y_i^{(t)} - \widehat{f}\left(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}\right)\right)$ of the single-task model parameters for each task, and updates only the coefficients $\boldsymbol{s}^{(t)}$ for the current task at each time step. This process reduces the optimization (1) to the problem of sparse coding the single-task modeling in the shared dictionary $\boldsymbol{L}$, and enables solving $\boldsymbol{L}$ and $\boldsymbol{S}$ efficiently by the following recursive updating rules that constitute the ELLA:

$$\boldsymbol{s}^{(t)} = \arg\min_{\boldsymbol{s}}\left\|\widehat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s}\right\|_{\boldsymbol{\Upsilon}^{(t)}}^2 + \mu\left\|\boldsymbol{s}\right\|_1, \tag{2}$$

$$\boldsymbol{A} = \boldsymbol{A} + \left(\boldsymbol{s}^{(t)}\left(\boldsymbol{s}^{(t)}\right)^{\mathrm{T}}\right)\otimes\boldsymbol{\Upsilon}^{(t)}, \tag{3}$$

$$\boldsymbol{b} = \boldsymbol{b} + \mathrm{vec}\left[\boldsymbol{s}^{(t)}\otimes\left(\left(\widehat{\boldsymbol{\theta}}^{(t)}\right)^{\mathrm{T}}\boldsymbol{\Upsilon}^{(t)}\right)\right], \tag{4}$$

$$\boldsymbol{L} = \boldsymbol{L} + \mathrm{mat}\left[\left(\frac{1}{T}\boldsymbol{A} + \lambda\boldsymbol{I}_{(kd)}\right)^{-1}\frac{1}{T}\boldsymbol{b}\right]_{d\times k}, \tag{5}$$

where $\|\boldsymbol{v}\|_{\boldsymbol{A}}^2 = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}$, the elements of $\boldsymbol{L}$ are initialized by randomly taking values from $(0,\ 1)$, $\boldsymbol{\Upsilon}^{(t)} = \boldsymbol{\Upsilon}\left(\widehat{\boldsymbol{\theta}}^{(t)}\right)$ is the Hessian matrix of the loss $\mathcal{J}\left(\widehat{\boldsymbol{\theta}}^{(t)}\right)$, $\otimes$ denotes the Kronecker product, and $\boldsymbol{A} \in \mathbb{R}^{(kd)\times(kd)}$ is initialized to the all zero-elements matrix, while $\boldsymbol{b} \in \mathbb{R}^{kd}$ is initialized to the all zero-elements vector, the vector stacking operator $\mathrm{vec}[\bullet]$ stacks the columns of matrix one by one to form a vector, $\boldsymbol{I}_{(kd)}$ is the $(kd)\times(kd)$ identity matrix, and the matrix forming operator $\mathrm{mat}[\bullet]_{d\times k}$ converts a (dk)-dimensional vector into a $(d\times k)$-dimensional matrix.

Each time when new task t arrives, this method requires the input-output data $\left(\boldsymbol{x}_i^{(t)}, y_i^{(t)}\right)_{i=1}^{n_t}$ to first estimate the model parameters $\widehat{\boldsymbol{\theta}}^{(t)}$ before updating $\boldsymbol{s}^{(t)}$ and $\boldsymbol{L}$. However, labeling data for every upcoming task is time-consuming, and most of the time we only have unlabeled or input data at the first glance of a new task. In order to eliminate this need for desired output data, in this paper, we propose to incorporate unsupervised feature into the learning process, and hence to enable unsupervised transfer on new tasks. Specifically, upon learning a few tasks with complete input-output data, future task models can be constructed given only input information.

3 Unsupervised Transfer Aided Lifelong Regression

3.1 Overview of Proposed Framework

For task t, define its training input data matrix $\boldsymbol{X}^{(t)} \in \mathbb{R}^{d\times n_t}$ by $\boldsymbol{X}^{(t)} = \left[\boldsymbol{x}_1^{(t)}\ \boldsymbol{x}_2^{(t)}\cdots\boldsymbol{x}_{n_t}^{(t)}\right]$ and the corresponding desired output vector $\boldsymbol{y}^{(t)} \in \mathbb{R}^{n_t}$ as $\boldsymbol{y}^{(t)} = \left[y_1^{(t)}\ y_2^{(t)}\cdots y_{n_t}^{(t)}\right]^{\mathrm{T}}$. As depicted in Fig. 1, our proposed framework follows the lifelong learning setting. During the agent's lifetime, massive tasks are received consecutively. As a new task arrives, knowledge accumulated from the previous tasks is selectively transferred to learn the new task, and newly acquired knowledge from the current task is stored in the knowledge base for future use. In order to achieve unsupervised knowledge transfer on new task, we incorporate unsupervised feature into lifelong learning via sparse coding with a coupled dictionary, enabling the unsupervised feature and task predictor to augment each other. For each task with complete training data, the task predictor is constructed by input and output data, while the unsupervised feature is encoded only by input data. In order to link two feature spaces, we employ two dictionaries that act as knowledge repositories for the two spaces, and they are coupled by a joint sparse representation. Because of the learned coupling, the predictor for a new task can be reconstructed given only the unsupervised feature. This capacity of learning new task predictors without desired output eliminates the need to labeling new tasks in lifelong regression process.

The above lifelong learning framework is a passive process, in which the learner has no control over the order of the tasks to learn. In some situations, the learner has knowledge of the next several tasks that it needs to learn. Motivated by [33], we further extend this framework to active lifelong regression by incorporating a similar active task selection mechanism. Hence, our model can choose the next task to learn from a pool of candidate tasks in order to

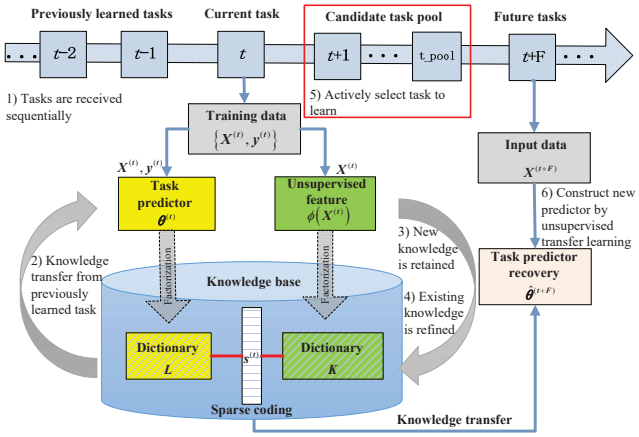

Fig. 1. Illustration of unsupervised transfer aided lifelong regression process.

maximize future learning performance. In the following, we will present each component of our algorithm in details.

3.2 Task Predictor

Ideally, each task has complete training input and output data $\left(\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}\right)$ that enables the construction of the task predictor $\widehat{f}(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\theta}$. We construct the task predictor by the regularized least square (LS) estimator and the model parameter parameter is thus computed as

$$\widehat{\boldsymbol{\theta}}^{(t)} = \left(\boldsymbol{X}^{(t)}\left(\boldsymbol{X}^{(t)}\right)^{\mathrm{T}} + \beta \boldsymbol{I}_d\right)^{-1}\boldsymbol{X}^{(t)}\boldsymbol{y}^{(t)}, \tag{6}$$

where β is a small positive regularization parameters, e.g., $\beta = 10^{-6}$. The Hessian $\boldsymbol{\Upsilon}^{(t)}$ of the squared-loss function $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ around the single task solution $\widehat{\boldsymbol{\theta}}^{(t)}$ is given by

$$\boldsymbol{\Upsilon}^{(t)} = \frac{1}{2n_t}\left(\boldsymbol{X}^{(t)}\left(\boldsymbol{X}^{(t)}\right)^{\mathrm{T}} + \beta \boldsymbol{I}_d\right), \tag{7}$$

For each task with complete training input and output data, we first compute the predictor's parameters $\widehat{\boldsymbol{\theta}}^{(t)}$ and Hessian $\boldsymbol{\Upsilon}^{(t)}$ before performing knowledge transfer in learning process.

3.3 Unsupervised Feature

When new task t arrives, it is often easy to obtain unlabeled or input data $\boldsymbol{X}^{(t)}$ while target output data $\boldsymbol{y}^{(t)}$ are difficult to acquire quickly. Although input data itself cannot be used to construct task predictor, it also contains vital information that identifies each task. Our goal is to use the input data to supplement the task predictor, treating it as a backup to learn new task when output data is unavailable.

To incorporate input information into the learning procedure, the input data matrix $\boldsymbol{X}^{(t)} \in \mathbb{R}^{d\times n_t}$ needs to be transformed into a d-dimensional feature vector that can link with the predictor's parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$. In order to link these two spaces, therefore, we transform the original input data matrix $\boldsymbol{X}^{(t)}$ into the d-dimensional feature vector $\phi(\boldsymbol{X}^{(t)})$, where $\phi(\bullet)$ is an operator that encodes a matrix as a vector. Express the i-th column of $\boldsymbol{X}^{(t)}$ as $\boldsymbol{x}_i^{(t)} = \left[x_{1,i}^{(t)}\ x_{2,i}^{(t)} \cdots x_{d,i}^{(t)}\right]^{\mathrm{T}}$. The simplest way to achieve this encoding is to compute the mean value of each row of $\boldsymbol{X}^{(t)}$, yielding

$$\phi(\boldsymbol{X}^{(t)}) = \left[\bar{x}_1^{(t)}\ \bar{x}_2^{(t)} \cdots \bar{x}_d^{(t)}\right]^{\mathrm{T}} = \bar{\boldsymbol{x}}^{(t)} \in \mathbb{R}^d, \tag{8}$$

where $\bar{x}_j^{(t)} = \frac{1}{n_t}\sum_{i=1}^{n_t} x_{j,i}$, $1 \le j \le d$. Hence, $\phi(\boldsymbol{X}^{(t)})$ is the unsupervised feature for task t. Our lifelong learner uses this feature vector to represent each task, treating it as side information to augment task predictor for individual tasks.

3.4 Coupled Dictionary Optimization

After obtaining the predictor parameter vector $\widehat{\boldsymbol{\theta}}^{(t)}$ and the unsupervised feature $\bar{\boldsymbol{x}}^{(t)} = \phi(\boldsymbol{X}^{(t)})$ for each task, the next step is to link the two feature spaces, so that each can augment the learning of the other. Motivated by [35], [36], we link the two feature spaces through the dual dictionaries that are coupled by a joint sparse representation. The original idea of using coupled dictionary learning is to link the high-level task descriptions with the learned model to achieve zero-shot transfer for new tasks. We use the coupled dictionaries to link the task predictor's space with the unsupervised feature' space, so as to make full use of input information and achieve learning new task without output data.

Recall that the lifelong learning approach factorizes the predictor parameters $\boldsymbol{\theta}^{(t)}$ for each task as a sparse linear combination of a shared dictionary by $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$, where each column of the dictionary $\boldsymbol{L}$ represents a cohesive chunk of knowledge. In lifelong learning, the dictionary $\boldsymbol{L}$ is refined overtime as the model learns more tasks. The sparse coefficient vectors $\boldsymbol{S}$ encode the task predictors in the shared dictionary, providing an embedding of the tasks based on how their predictors share knowledge. Similar to this, the unsupervised feature vector $\bar{\boldsymbol{x}}^{(t)}$ can also be linearly factorized using a shared dictionary $\boldsymbol{K} \in \mathbb{R}^{d\times k}$ over the unsupervised feature's space. Like $\boldsymbol{L}$, this dictionary $\boldsymbol{K}$ captures the relationships among the unsupervised features for multiple tasks, with the coefficients that similarly embed tasks based on the commonalities in their unsupervised features. In order to link the two spaces, we enforce the two dictionaries, $\boldsymbol{L}$ and $\boldsymbol{K}$, to share the same sparse coefficient vectors $\boldsymbol{S}$ so as to reconstruct both the predictors and the unsupervised features. Hence, for task t,

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)},\ \bar{\boldsymbol{x}}^{(t)} = \boldsymbol{K}\boldsymbol{s}^{(t)}. \tag{9}$$

Because we enforce the two dictionaries with the same sparse code $\boldsymbol{s}^{(t)}$, the relevant pieces of information for a task predictor are coupled with its associated unsupervised feature. To optimize $\boldsymbol{L}$ and $\boldsymbol{K}$, we first reformulate the objective (1) for the coupled dictionaries as

$$\min_{\boldsymbol{L},\boldsymbol{K},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right) + \rho\left\|\bar{\boldsymbol{x}}^{(t)} - \boldsymbol{K}\boldsymbol{s}^{(t)}\right\|_2^2 + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right) + \lambda\left(\|\boldsymbol{L}\|_{\mathrm{F}}^2 + \|\boldsymbol{K}\|_{\mathrm{F}}^2\right), \tag{10}$$

where the parameter ρ balances the task predictor's fit to the unsupervised feature's fit.

To solve the optimization (10) in a lifelong setting, we approximate $\mathcal{J}\left(\boldsymbol{\theta}^{(t)}\right)$ by a second-order Taylor expansion

around the regularized LS parameter estimate $\widehat{\boldsymbol{\theta}}^{(t)}$ given in (6). That is, we expand $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ around $\widehat{\boldsymbol{\theta}}^{(t)}$ for each task as:

$$\mathcal{J}(\boldsymbol{\theta}^{(t)}) \approx \mathcal{J}(\widehat{\boldsymbol{\theta}}^{(t)}) + \triangledown \mathcal{J}(\widehat{\boldsymbol{\theta}}^{(t)})(\boldsymbol{\theta}^{(t)} - \widehat{\boldsymbol{\theta}}^{(t)}) + \frac{1}{2}\left\|\boldsymbol{\theta}^{(t)} - \widehat{\boldsymbol{\theta}}^{(t)}\right\|^2_{\boldsymbol{\Upsilon}^{(t)}}, \tag{11}$$

where $\triangledown$ denotes the gradient operator. The first term $\mathcal{J}(\widehat{\boldsymbol{\theta}}^{(t)})$ is a constant and can be omitted. Since $\boldsymbol{\theta}^{(t)}$ is the minimizer of the objective $\mathcal{J}(\boldsymbol{\theta}^{(t)})$, $\triangledown \mathcal{J}(\widehat{\boldsymbol{\theta}}^{(t)})$ is zero, and the second term can also be removed. Thus, the loss function $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ is approximated by the last term of (11), which can be rewritten as $\left\|\widehat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Upsilon}^{(t)}}$, given $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$. With this approximated $\mathcal{J}(\boldsymbol{\theta}^{(t)})$, the optimization (10) is simplified as

$$\min_{\boldsymbol{L},\boldsymbol{K},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\left\|\widehat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Upsilon}^{(t)}} + \rho\left\|\bar{\boldsymbol{x}}^{(t)} - \boldsymbol{K}\boldsymbol{s}^{(t)}\right\|_2^2 + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right) + \lambda(\|\boldsymbol{L}\|_{\mathrm{F}}^2 + \|\boldsymbol{K}\|_{\mathrm{F}}^2). \tag{12}$$

Further defining:

$$\boldsymbol{\Theta}^{(t)} = \begin{bmatrix}\widehat{\boldsymbol{\theta}}^{(t)} \\ \bar{\boldsymbol{x}}^{(t)}\end{bmatrix},\ \boldsymbol{H} = \begin{bmatrix}\boldsymbol{L} \\ \boldsymbol{K}\end{bmatrix},\ \boldsymbol{\Psi}^{(t)} = \begin{bmatrix}\boldsymbol{\Upsilon}^{(t)} & \boldsymbol{0}_{d\times d} \\ \boldsymbol{0}_{d\times d} & \rho\boldsymbol{I}_d\end{bmatrix}, \tag{13}$$

where $\boldsymbol{0}_{d\times d}$ is the $d \times d$ zero matrix, the optimization (12) can be rewritten in a concise form as

$$\min_{\boldsymbol{H},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\left\|\boldsymbol{\Theta}^{(t)} - \boldsymbol{H}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Psi}^{(t)}} + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right) + \lambda\|\boldsymbol{H}\|_{\mathrm{F}}^2. \tag{14}$$

This optimization has the identical form to (1), and it can be solved efficiently in a lifelong setting. Specifically, similar to the classic ELLA, we can solve the sparse vector $\boldsymbol{s}^{(t)}$ given $\boldsymbol{H}$ acquired at task $(t-1)$, and then update $\boldsymbol{H}$, i.e., $\boldsymbol{L}$ and $\boldsymbol{K}$, given $\boldsymbol{s}^{(t)}$. Obviously, given $\boldsymbol{s}^{(t)}$, the two dictionaries $\boldsymbol{L}$ and $\boldsymbol{K}$ can be updated independently.

When a task arrives, we perform three operations to update our model, namely, compute $\boldsymbol{s}^{(t)}$, update $\boldsymbol{L}$ and update $\boldsymbol{K}$. Specifically, the sparse vector $\boldsymbol{s}^{(t)}$ is first computed using the current basis $\boldsymbol{H}$ by solving the following L_1-regularized regression problem, which is an example of the Lasso:

$$\boldsymbol{s}^{(t)} = \arg\min_{\boldsymbol{s}} \left\|\boldsymbol{\Theta}^{(t)} - \boldsymbol{H}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Psi}^{(t)}} + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1. \tag{15}$$

After $\boldsymbol{s}^{(t)}$ is obtained, the two dictionaries, $\boldsymbol{L}$ and $\boldsymbol{K}$, can be calculated independently by the recursive updating equations (3) to (5). In particular, to update the dictionary $\boldsymbol{K}$, we simply replace $\boldsymbol{\Upsilon}^{(t)}$ by $\rho\boldsymbol{I}_d$, $\widehat{\boldsymbol{\theta}}^{(t)}$ by $\bar{\boldsymbol{x}}^{(t)}$ and $\boldsymbol{L}$ by $\boldsymbol{K}$ in (3) to (5). The per-task updating rules are given in Algorithm 1.

Remark 1. Solving the sparse coding by (15) is basically learning new task with the previously built knowledge repository $\boldsymbol{H}$, that is, knowledge transfer from past learned tasks, while the adaptation of $\boldsymbol{L}$ and $\boldsymbol{K}$ is to retain knowledge from the current task and refine the existing knowledge base. These two operations form the core idea of lifelong learning.

Algorithm 1 Unsupervised transfer aided lifelong regression

Parameters: Size of dictionaries k, regularization parameters μ and λ, balance coefficient ρ.
Initialize: Randomly initialize $\boldsymbol{L}$ and $\boldsymbol{K}$, $T = 0$.
While some task is available **do**
Collect training input-output data $\{\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}\}$ from task $\mathbb{Z}^{(t)}$, set $T = T + 1$.
Construct task predictor, and compute model parameter $\widehat{\boldsymbol{\theta}}^{(t)}$ and Hessian $\boldsymbol{\Upsilon}^{(t)}$ using (6) and (7), respectively.
Encode $\boldsymbol{X}^{(t)}$ into feature vector $\bar{\boldsymbol{x}}^{(t)}$ using (8).
Construct matrices $\boldsymbol{\Theta}^{(t)}$, $\boldsymbol{H}$, and $\boldsymbol{\Psi}^{(t)}$ of (13).
Solve sparse coding $\boldsymbol{s}^{(t)}$ by Lasso of (15).
$\boldsymbol{L} \leftarrow \mathrm{update}L(\boldsymbol{L}, \boldsymbol{s}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}, \boldsymbol{\Upsilon}^{(t)}, \lambda)$ by (3)-(5).
$\boldsymbol{K} \leftarrow \mathrm{update}K(\boldsymbol{K}, \boldsymbol{s}^{(t)}, \phi(\boldsymbol{X}^{(t)}), \rho\boldsymbol{I}_d, \lambda)$ by (3)-(5).
For: $t \in \{1, \ldots, T\}$ **do**: $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$
End while

3.5 Unsupervised Transfer Learning

In a lifelong setting, multiple consecutive tasks arrive rapidly and it may have insufficient time to labeling every coming task, and for tasks with only input data, it is unable to construct predictor. Incorporating unsupervised feature however enables our approach to construct a predictor for the new task with only input data. This ability to perform unsupervised transfer is enabled by the coupled dictionary learning, which allows us to use unsupervised feature to recover task predictor through coupled dictionaries and sparse coding. The unsupervised transfer process for learning a new task using solely unlabeled data as well as the previously learned libraries $\boldsymbol{L}$ and $\boldsymbol{K}$ is shown in Fig. 2.

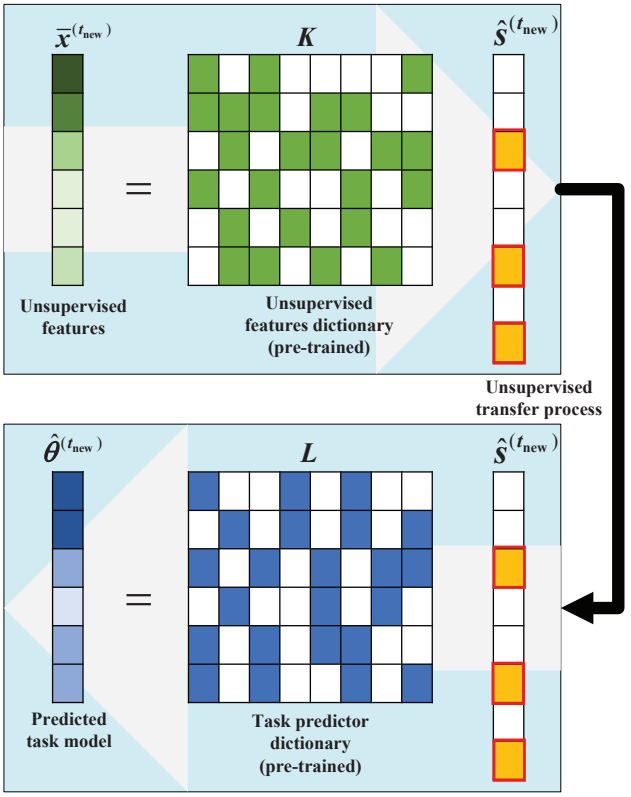

Fig. 2. Illustration of task predictor recovery using solely unlabeled data by unsupervised transfer.

Given the input data $\boldsymbol{X}^{(t_{new})}$ for a new task, we first encode $\boldsymbol{X}^{(t_{new})}$ as the feature vector $\bar{\boldsymbol{x}}^{(t_{new})} = \phi(\boldsymbol{X}^{(t_{new})})$, and then estimate the sparse coding in the latent unsupervised feature space via Lasso on the learned dictionary $\boldsymbol{K}$

$$\widehat{\boldsymbol{s}}^{(t_{new})} = \arg\min_{\boldsymbol{s}} \left\| \bar{\boldsymbol{x}}^{(t_{new})} - \boldsymbol{K}\boldsymbol{s} \right\|_2^2 + \mu \left\| \boldsymbol{s} \right\|_1. \tag{16}$$

Since this estimated $\widehat{\boldsymbol{s}}^{(t_{new})}$ also serves as the sparse coding for the latent dictionary $\boldsymbol{L}$, it can be used to recover the task predictor for the new task t_{new} as

$$\widehat{\boldsymbol{\theta}}^{(t_{new})} = \boldsymbol{L}\widehat{\boldsymbol{s}}^{(t_{new})}. \tag{17}$$

Hence, this new task predictor's parameter $\widehat{\boldsymbol{\theta}}^{(t_{new})}$ is obtained only through the task's input data $\boldsymbol{X}^{(t_{new})}$. This eliminates the need to collect output data for model construction. This unsupervised transfer learning procedure is given in Algorithm 2.

Algorithm 2 Unsupervised knowledge transfer to a new task

1: **Inputs**: input data for new task $\boldsymbol{X}^{(t_{new})}$, learned libraries $\boldsymbol{L}$ and $\boldsymbol{K}$.
2: Encode $\boldsymbol{X}^{(t_{new})}$ into feature vector $\bar{\boldsymbol{x}}^{(t_{new})}$ using (8).
3: Solve sparse coding $\boldsymbol{s}^{(t_{new})}$ by Lasso of (16).
4: Recover task predictor by computing its parameter vector $\widehat{\boldsymbol{\theta}}^{(t_{new})}$ using (17).

3.6 Active Task Selection

To make our method capable of learning in a task-efficient manner, we further incorporate an active task selection mechanism into our approach. The problem is formulated as follows. The agent has access to training data from a pool of candidate unlearned tasks $\left\{\mathbb{Z}^{(T+1)}, \ldots, \mathbb{Z}^{(T_{\text{pool}})}\right\}$, where $T+1 < T_{\text{pool}} < T_{\max}$. Based on training data for these candidate tasks, the learner selects the index of the next task to learn $t_{\text{next}} \in \{T+1, \ldots, T_{\text{pool}}\}$, which will maximize the learning performance. Without loss of generality, the value of T_{pool} is fixed and set as $T_{\text{pool}} = \frac{1}{2}T_{\max}$ in our study.

We employ the diversity heuristic proposed in [33] for selecting the next best task. The basic idea is to encourage the current learned model or library to capture information of the widest range of tasks. If the current library does not fit well for a new task t, it means that the information on task t has not been captured in the current library. Thus, in order to acquire information from the widest range of tasks, the next task should be the one that the current library is doing the worst, that is, the loss on the training data of this task is maximum. Although we have the dual dictionaries $\boldsymbol{L}$ and $\boldsymbol{K}$, we can simply use the main dictionary $\boldsymbol{L}$ that contains both input and output information, to calculate the heuristic as

$$t_{\text{next}} = \arg\max_{t \in \{T+1,\ldots,T_{\text{pool}}\}} \min_{\boldsymbol{s}} \left\| \widehat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s} \right\|_{\boldsymbol{\Upsilon}^{(t)}}^2 + \mu \left\| \boldsymbol{s} \right\|_1, \tag{18}$$

where $\widehat{\boldsymbol{\theta}}^{(t)}$ and $\boldsymbol{\Upsilon}^{(t)}$ are calculated by (6) and (7), respectively.

Remark 2. The active task selection mechanism (18) tends to select tasks that are encoded poorly with the current dictionary $\boldsymbol{L}$, and the selected tasks are likely to be significantly different from the previously learned tasks, thus encouraging the agent to learn diverse tasks. However, this does not means that after the active task selection based training of t_{next} tasks, the model generalization or test performance is necessarily better than the model with the non-active task selection based training of t_{next} tasks. Whether this is the case depends on the underlying data generating process. Furthermore, when all the training tasks are used, the models obtained with and without active task selection should have the same or similar test performance, because the both models have seen all the training tasks and the order of all the training tasks learned should have litter effect on the overall generalization performance.

4 Algorithm Summary and Analysis

Our proposed approach has two versions, namely, unsupervised transfer aided lifelong regression (UTLR), in which the agent has no control over learning order of tasks, and UTLR-Ac, which is equipped with active task selection mechanism presented in Subsection 3.5. The proposed framework has two phases: training phase and evaluation phase. During training phase, some training tasks serve as the candidate task pool. Each time the agent actively chooses (UTLR-Ac) or passively accepts (UTLR) one task from the task pool to learn so as to incrementally build its libraries $\boldsymbol{L}$ and $\boldsymbol{K}$. After the agent has encountered all the training tasks, the two libraries are fixed and they act as the knowledge base to help learning future unseen tasks. During evaluation phase, new task arrives sequentially. With the aid of $\boldsymbol{L}$ and $\boldsymbol{K}$, the agent performs either model prediction or unsupervised transfer depending on whether the new task is labeled or not. For the unlabeled new task, the agent only uses the input data to recover the task predictor so as to predict the new data of this task[1].

Convergence analysis: In order to prove the convergence of the proposed framework, we use the theoretical results of [22], since these results can directly apply to our framework.

The work [22] has proved that the learned dictionary becomes increasingly stable, i.e., converged, as more tasks are learned. This convergence result requires two conditions:

1) The tuples $\left(\boldsymbol{\Upsilon}^{(t)}, \widehat{\boldsymbol{\theta}}^{(t)}\right)$ are drawn from an independent identical distribution (i.i.d.) with compact support to bound the norms of $\boldsymbol{L}$ and $\boldsymbol{s}^{(t)}$.
2) For all the tasks up to task t, let $\boldsymbol{L}_k$ be the subset of the current dictionary $\boldsymbol{L}_t$, where only the columns corresponding to the non-zero elements of $\boldsymbol{s}^{(t)}$ are included. Then, all the eigenvalues of the matrix $\boldsymbol{L}_k^{\mathrm{T}}\boldsymbol{\Upsilon}^{(t)}\boldsymbol{L}_k$ need to be strictly positive.

The work [22] demonstrates that both these conditions are met for the lifelong learning framework given in (2) to (5).

We incorporate unsupervised feature into this framework by augmenting $\widehat{\boldsymbol{\theta}}^{(t)}$ into $\boldsymbol{\Theta}^{(t)}$, $\boldsymbol{L}$ into $\boldsymbol{H}$, and $\boldsymbol{\Upsilon}^{(t)}$ into $\boldsymbol{\Psi}^{(t)}$. Since $\widehat{\boldsymbol{\theta}}^{(t)}$ and $\boldsymbol{\Upsilon}^{(t)}$ are drawn from an i.i.d., clearly $\boldsymbol{\Theta}^{(t)}$ and $\boldsymbol{\Psi}^{(t)}$ are also drawn from an i.i.d., according to the definition of (13). Hence condition 1) holds for our

1. Code is available at: https://github.com/neuroton42/Unsupervised-Transfer-Aided-Lifelong-Regression-.git

method. To verify condition 2), we note that the eigenvalue of $\boldsymbol{H}_k^{\mathrm{T}}\boldsymbol{\Psi}^{(t)}\boldsymbol{H}_k$ are the eigenvalues of $\boldsymbol{L}_k^{\mathrm{T}}\boldsymbol{\Upsilon}^{(t)}\boldsymbol{L}_k$ and the positive ρ, and hence they are strictly positive. Therefore, both the two conditions are met for our proposed method, and the convergence result of [22] can be applied to our proposed approach.

Computational complexity: We now analyze the online computational complexity of learning new task by our method. The construction of task predictor by the regularized LS estimator (6) has a complexity on the order of $\mathsf{O}(d^3)$. The adaptation of single dictionary $\boldsymbol{L} \in \mathbb{R}^{d\times k}$ and sparse coding $\boldsymbol{s}^{(t)} \in \mathbb{R}^k$ costs $\mathsf{O}(k^2d^3)$. Since we incorporate unsupervised feature into lifelong learning by augmenting $\boldsymbol{L} \in \mathbb{R}^{d\times k}$ into $\boldsymbol{H} \in \mathbb{R}^{(2d)\times k}$, the coupled dictionary adaptation costs $\mathsf{O}\big(k^2(2d)^3\big)$. Thus, the overall complexity of per-task adaptation is $\mathsf{O}\big(d^3 + k^2(2d)^3\big)$, which is independent of task number.

5 Experiments

Three real-world applications, examination score prediction, Parkinson disease symptom score prediction and Alzheimer disease progression modeling, are included to demonstrate the effectiveness of our proposed approach.

5.1 Experimental Setup

Our two proposed methods, UTLR and UTLR-Ac, are compared with three existing lifelong learning approaches, the ELLA [22], the ELLA-diver [33], which actively chooses tasks to learn with diversity heuristic method, and the ELLA-diver++ [33], which is a stochastic version of ELLA-diver. The alternative lifelong learning approach EWC is also chosen as a benchmark for comparison. In the original work [29], the cross-entropy loss is used for classification problem, and we modify the loss of EWC to the mean square error in order to apply EWC to solve regression problems. Additionally, the single-task learning (STL) that learns multiple tasks independently, is used as the baseline method. The LS regression is used to construction task predictor for all the methods. It should be noted that either the unsupervised domain adaptation or continual domain adaptation methods are unsuitable to be compared with the proposed lifelong regression method, as they address very different problems. Also TaDell [36] cannot be used for comparison, because it needs domain-specific task descriptor, which is not available for most real-world datasets. Basically, our method can be regarded as a generalized version of TaDell using task input data rather than domain-specific task descriptor.

For all the lifelong models, the dictionary size k and the regularization parameters are independently chosen for each dataset using grid search over the ranges of $\{1, 2, \cdots, 5\}$ for k and $\{10^{-n}, n = 0, \cdots, 6\}$ for the regularization parameters, respectively, to achieve their best performance. The previous works [22], [33], [36] suggested to select the dictionary size from $k \in \{1, 2, \cdots, 10\}$ but the datasets used in these previous works were most classification problems. We have experimented with $k \in \{1, 2, \cdots, 10\}$ for our three case studies but the results were not better than with $k \in \{1, 2, \cdots, 5\}$. The analysis on the sensitivity of the algorithmic parameters can be found in [22], [52]. For EWC, a two-layer MLP with ReLU non-linearities in each layer is utilized as the training model. The network model is trained using stochastic gradient descent with learning rate 0.0001, and 10 independent experiments with different random seeds are conducted for model training. For our proposed method, ρ is a key parameter that balances the predictor's fit to the unsupervised feature's fit, and we empirically investigate its impact on the model prediction and unsupervised transfer performance.

In the experiments, we split the data 50%-50% as the training and testing datasets for each task. The training set is used to construct task predictor, while the testing set is for performance evaluation. Additionally, we divide the set of tasks into two subsets: one set of *training tasks* that serve as the pool for active task selection (for UTLR-Ac, ELLA-diver and ELLA-diver++) and are used to learn the knowledge library, and one set of *evaluation tasks* on which we measure the performance of the learned library. We set $T_{pool} = \frac{1}{2}T_{max}$ for all the experiments. After a model has learned all the training tasks, its knowledge library is fixed, and we use the learned library to measure the prediction performance on the testing datasets of evaluation tasks. For the evaluation of unsupervised transfer, the model has no access to the training set's output data for each evaluation task, and it can only use the input data to recover the task predictor for performance evaluation on the testing set.

The root mean squared error (RMSE) and the mean absolute error (MAE) are used to evaluate the testing prediction performance. In the lifelong learning setting, we are also interested in the online computational complexity for learning each task. Hence, the averaged computation time per task (ACTpT) is utilized to quantify the online computational complexity of a lifelong model. For all the lifelong models, training tasks are presented sequentially to the learner, following the corresponding online learning setting. To mitigate the impact of task order on the algorithms, the training and evaluation task orders are randomly generated over 10 independent experiments, and we report the mean and standard deviation (STD) of the RMSE, MAE and ACTpT over 10 realizations.

5.2 School Examination Score Prediction

We first evaluate the algorithms on school examination score dataset which has been widely used in multi-task and lifelong regression investigation [7], [9], [11], [13], [21], [22]. The dataset contains examination scores of 15362 students from 139 secondary schools, and each school is considered as a regression task. For each task, the goal is to predict scores for all the students in the school according to their input features. Each student has 28 features (the task dimension $d = 28$), including student-specific features and school-specific features, and the corresponding output is the student's examination score. The numbers of students for these 139 schools vary from 25 to 251 (the number of samples for each task $n_t \sim 25$ to 251). From the total of 139 tasks, we use 69 as the training tasks and the other 70 as the evaluation tasks.

The impact of ρ on the prediction and unsupervised transfer performance of our two models is investigated

TABLE 2
Performance comparison of STL, ELLA, ELLA-diver, ELLA-diver++, EWC as well as proposed UFLR and UFLR-Ac for school examination score dataset.

Method	Model prediction		Unsupervised transfer		ACTpT (ms)
	RMSE	MAE	RMSE	MAE	
STL	0.1697±0.0029	0.1321±0.0021	-	-	0.1575±0.0360
ELLA	0.1596±0.0060	0.1245±0.0042	-	-	**0.9044±0.3035**
ELLA-diver	0.1594±0.0066	0.1250±0.0052	-	-	18.6944±0.7049
ELLA-diver++	**0.1559±0.0039**	**0.1219±0.0029**	-	-	19.0246±1.2701
EWC	0.1613±0.0046	0.1304±0.0045	-	-	-
UFLR	**0.1517±0.0025**	**0.1203±0.0021**	0.1561±0.0029	0.1241±0.0025	**0.7987±0.2295**
UFLR-Ac	**0.1517±0.0026**	**0.1203±0.0025**	0.1563±0.0037	0.1243±0.0035	15.3414±1.0405

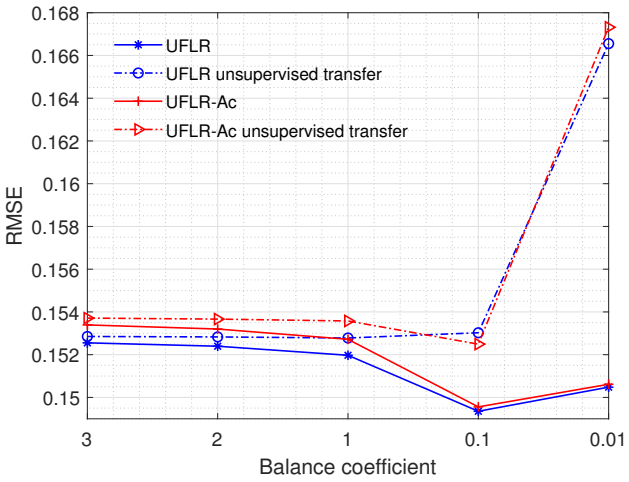

Fig. 3. Impact of ρ on the model prediction and unsupervised transfer performance of the proposed methods for school examination score dataset.

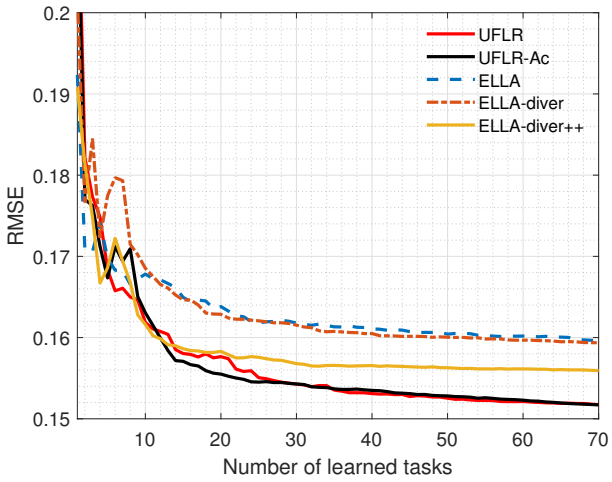

Fig. 4. Comparison of test RMSE performance versus number of tasks learned for school examination score dataset.

in Fig. 3. When the value of ρ is large ($\rho = 3, 2, 1$), the unsupervised feature plays the dominant role and the task model has less impact on the algorithm's performance. Hence the model prediction performance are similar to the unsupervised transfer performance. When $\rho = 0.1$, the model prediction accuracy improves while maintaining an acceptable unsupervised transfer accuracy, which indicates that this value of ρ balances well the task model's fit to the unsupervised feature's fit. When ρ decreases further to 0.01, the model prediction accuracy only decreases slightly but the unsupervised transfer performance degrades dramatically. This is because when ρ becomes very small, the unsupervised feature has little impact on the algorithm, which makes it unable to recover the task predictor via transfer. Hence, $\rho = 0.1$ is appropriate for this case study.

Table 2 presents the test performance comparison of various methods, where for the lifelong learning methods, the models with the best and runner-up performance are emphasized with boldface black and blue colours, respectively. Note that for our proposed approach, the model prediction is carried out by both the task predictor and the unsupervised feature, while the unsupervised transfer performance is obtained only by the unsupervised feature. Our approach is the only one that can carry out this unsupervised feature based prediction. Since EWC is implemented on PyTorch, it will be unfair to compare its computation time with other models that are implemented on Matlab. So we do not present the ACTpT of EWC. However, since EWC is based on neural networks, its training and testing are much more time costly than the other methods using linear base models. Clearly, although the STL imposes the least computational cost, it has the worst prediction performance compared with the lifelong models. Both our UFLR and UFLR-Ac attains the smallest model prediction RMSE and MAE, compared with the three ELLA-based methods and EWC model. Moreover, our UFLR imposes the lowest ACTpT among all the lifelong models. Most significantly, our proposed approach is able to use input data only to recover the task model, and achieves the unsupervised transfer performance that is similar to or slightly better than the ELLA-based methods. This clearly demonstrates the excellent unsupervised transfer performance of our method.

Clearly, after all the training tasks have been learned, the performance of an active task selection based lifelong model should be the same or similar to that of the non-active task selection based counterpart, and incorporating active task selection into a lifelong model significantly increases the algorithm's complexity. To investigate how task selection impacts on the model's prediction performance, we conduct the following experiment. After the lifelong model selects a training task to update its knowledge library, we measure its test performance on all the evaluation tasks using the current library. This procedure yields a learning curve that depicts the relationship between the prediction performance and the number of tasks learned, which is shown in Fig. 4

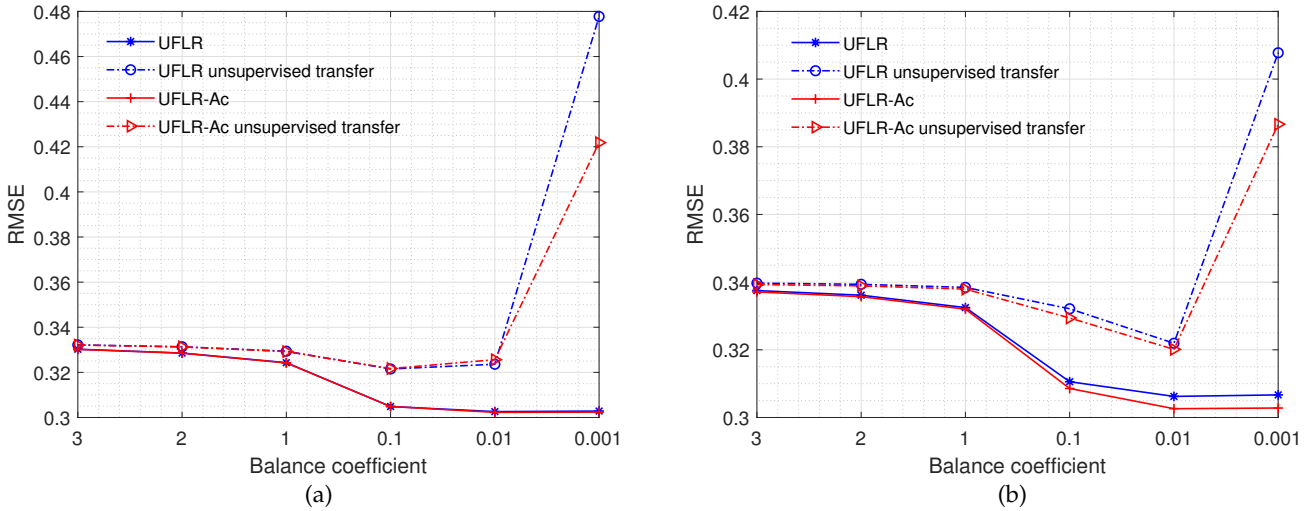

Fig. 5. Impact of ρ on the model prediction and unsupervised transfer performance of the proposed methods for (a) Parkinson-Motor, and (b) Parkinson-Total.

for the five lifelong model. The test RMSEs of all the models decrease as the number of tasks learned increases. This is because the lifelong models become more knowledgeable as their libraries capture more knowledge from more tasks. The learning curves of our UFLR and UFLR-Ac are very similar, and this is also the case for the ELLA and ELLA-diver. This indicates that active task selection only has minor impact on the lifelong model's generalization performance for this case study. The reason may be that the data distributions for the tasks of school examination score dataset are similar. Note that since the training pipeline of EWC is different from the ELLA-based methods, we do not conduct this experiment for EWC.

5.3 Parkinson Disease Symptom Score Prediction

This dataset is composed of a range of biomedical voice measurements from 42 patients with early-stage Parkinson's disease [53], and it has been used to evaluate lifelong models [13], [34]. The dataset contains 5875 voice recordings from these 42 patients, with the observations for each patient vary from $n_t = 101$ to 168. The aim is to predict the Motor and Total UPDRS scores from the 16 voice measures. The symptom score prediction using $d = 16$ biomedical features for a patient is considered as a regression task and we have 42 tasks in total. Since the UPDRS scores consist of Motor and Total, we establish two regression datasets in our experiment: **Parkinson-Motor** and **Parkinson-Total**, each containing 20 training tasks and 21 evaluation tasks.

TABLE 3
Performance comparison of STL, ELLA, ELLA-diver, ELLA-diver++, EWC as well as proposed UFLR and UFLR-Ac for Parkinson-Motor dataset.

Methods	Model prediction		Unsupervised transfer		ACTpT (ms)
	RMSE	MAE	RMSE	MAE	
STL	0.3658±0.0134	0.2874±0.0091	-	-	0.1235±0.0553
ELLA	**0.3125±0.0066**	**0.2663±0.0053**	-	-	**1.2792±0.9694**
ELLA-diver	0.3171±0.0041	0.2698±0.0040	-	-	7.2818±1.7257
ELLA-diver++	0.3148±0.0054	0.2681±0.0052	-	-	8.5184±1.4950
EWC	0.3388±0.0009	0.2989±0.0007	-	-	-
UFLR	**0.3067±0.0028**	**0.2602±0.0043**	0.3506±0.0158	0.3039±0.0129	**1.3134±1.1294**
UFLR-Ac	**0.3072±0.0038**	**0.2609±0.0046**	0.3493±0.0099	0.3032±0.0097	8.2228±2.5817

TABLE 4
Performance comparison of STL, ELLA, ELLA-diver, ELLA-diver++, EWC as well as proposed UFLR and UFLR-Ac for Parkinson-Total dataset.

Methods	Model prediction		Unsupervised transfer		ACTpT (ms)
	RMSE	MAE	RMSE	MAE	
STL	0.3718±0.0078	0.2935±0.0064	-	-	0.1525±0.0574
ELLA	**0.3149±0.0057**	**0.2701±0.0050**	-	-	**1.3578±1.0468**
ELLA-diver	0.3168±0.0055	0.2716±0.0048	-	-	9.0110±3.0789
ELLA-diver++	0.3157±0.0058	0.2707±0.0051	-	-	8.8302±2.5508
EWC	0.3487±0.0042	0.2958±0.0018	-	-	-
UFLR	**0.3076±0.0044**	**0.2633±0.0021**	0.3398±0.0124	0.2936±0.0089	**1.3175±1.1060**
UFLR-Ac	**0.3066±0.0039**	**0.2625±0.0031**	0.3377±0.0108	0.2927±0.0088	5.2056±1.6684

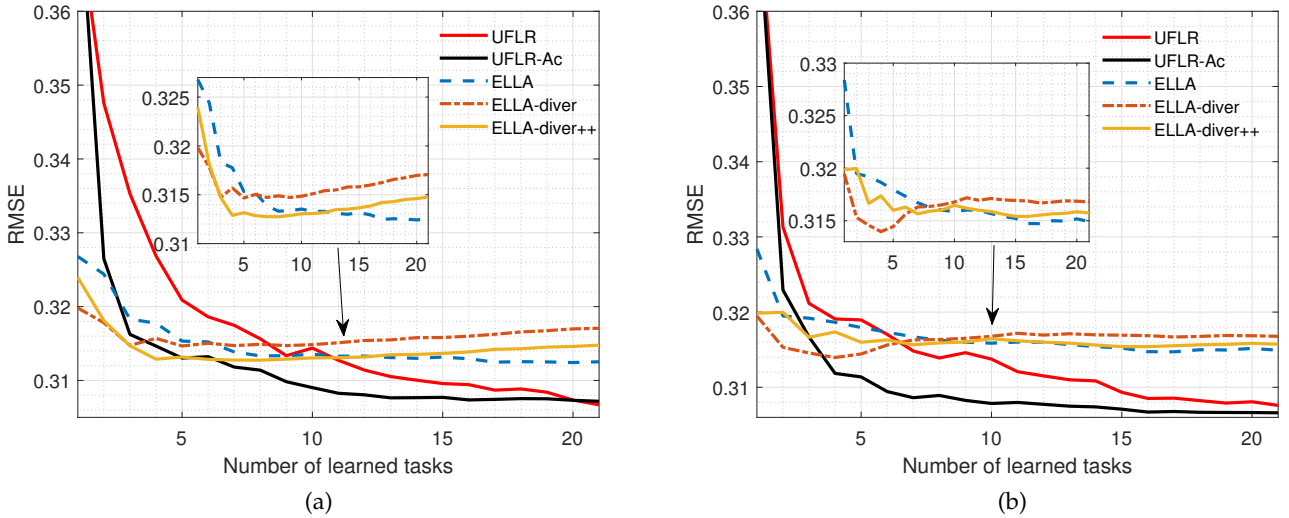

Fig. 6. Comparison of test RMSE performance versus number of tasks learned for (a) Parkinson-Motor, and (b) Parkinson-Total.

Based on the results of Fig. 5, we set $\rho = 0.01$ for our method, as this value best trades off the model prediction and unsupervised transfer.

The test performance of various models for Parkinson-Motor and Parkinson-Total datasets are compared in Tables 3 and 4, respectively. Again, our methods achieves the best prediction performance with the smallest test RMSE and MAE. Furthermore, our UFLR attains the second-lowest ACTpT and the lowest ACTpT for the two datasets, respectively. Also our models can recover the task model using input data only. The unsupervised transfer performance of our models, although not as accurate as the model prediction accuracy of ELLA, are better than that of STL. Fig. 6 depicts the test learning curves as the functions of the number of tasks learned for various lifelong models. It can be seen that although our models begin with the larger test RMSEs than the ELLA-based models, their prediction errors decreases dramatically after learning more tasks. This is because our methods have two libraries to initialize, thus having higher error at the beginning, and as the number of tasks learned increases, the two libraries can capture more knowledge, leading to higher prediction accuracy. Observe that the RMSE learning curve of UFLR-Ac decreases more quickly than that of UFLR. For Parkinson-Motor and Parkinson-Total datasets, the active task selection mechanism seems to enable the model to learn faster.

5.4 Alzheimer Disease Progression Modeling

This dataset is from Alzheimer's Disease Neuroimaging Initiative (ADNI) [54]. The ADNI project is a longitudinal study, which collects various measurements repeatedly over a 6-month or 1-year interval from patients. The first time patients receiving screening in hospital to obtain magnetic resonance imaging (MRI) is called baseline, and the time point for the follow-up visits is denoted by the duration starting from the baseline. The latest ADNI has up to 120 months' follow-up data available for some patients, which are divided as baseline (M00), 6-th month (M06), 12-th month (M12), 24-th month (M24), 36-th month (M36), 48-th month (M48), 60-th month (M60), 72-th month (M72), 84-th month (M84), 96-th month (M96), 108-th month (M108) and 120-th month (M120). The aim is to predict patients' cognitive scores at multiple time points using their MRI features. Hence, the cognitive score prediction at one time point is considered as a regression task, and we have 12 tasks in total. The number of samples for each task varies from $n_t = 69$ to 1074, and the dimension of features is $d = 314$. Alzheimer disease progression prediction is a very popular multi-task regression problem [55], [56], [57], [58], [59]. Because the patient's data can be received at consecutive time points over a long-time scale, for the first time we consider it as a lifelong regression problem and use this dataset to evaluate lifelong models. In this study, we have two cognitive measurements, including Mini Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale Cognitive Subscale (ADAS-cog). Hence we establish two regression datasets in our experiment: **Alzheimer-ADAS** and **Alzheimer-MMSE**. Each dataset contains 6 training tasks and 6 evaluation tasks.

The impact of ρ on the model prediction and unsupervised transfer performance of our methods for Alzheimer-ADAS dataset is shown in Fig. 7. For Alzheimer-MMSE, the results are similar and therefore they are omitted. Unlike the previous two case studies, the unsupervised feature plays a

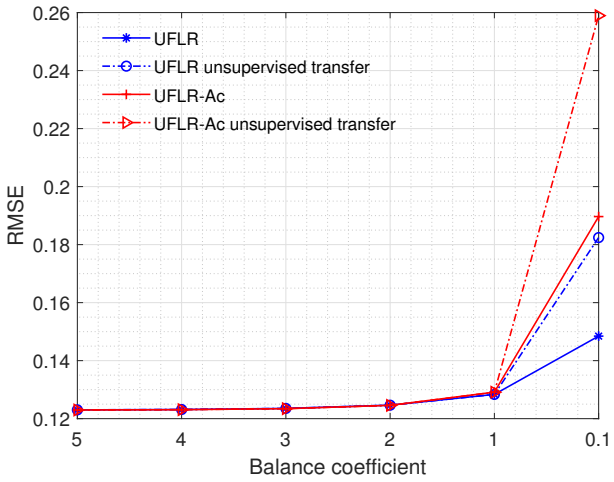

Fig. 7. Impact of ρ on the model prediction and unsupervised transfer performance of the proposed methods for Alzheimer-ADAS dataset.

TABLE 5
Performance comparison of STL, ELLA, ELLA-diver, ELLA-diver++, EWC as well as proposed UFLR and UFLR-Ac for Alzheimer-ADAS dataset.

Methods	Model prediction		Unsupervised transfer		ACTpT (ms)
	RMSE	MAE	RMSE	MAE	
STL	0.2220±0.0371	0.1648±0.0265	-	-	11.6719±0.8549
ELLA	0.1376±0.0080	0.1121±0.0072	-	-	**25.8865±6.8074**
ELLA-diver	0.1379±0.0080	0.1123±0.0072	-	-	73.4803±7.7040
ELLA-diver++	0.1378±0.0080	0.1122±0.0072	-	-	71.9221±8.2020
EWC	**0.1211±0.0011**	**0.0926±0.0013**	-	-	-
UFLR	**0.1108±0.0065**	**0.0816±0.0055**	0.1109±0.0065	0.0816±0.0055	**23.3615±4.2824**
UFLR-Ac	**0.1109±0.0066**	**0.0817±0.0056**	0.1109±0.0066	0.0817±0.0056	73.7116±9.1377

TABLE 6
Performance comparison of STL, ELLA, ELLA-diver, ELLA-diver++, EWC as well as proposed UFLR and UFLR-Ac for Alzheimer-MMSE dataset.

Methods	Model prediction		Unsupervised transfer		ACTpT (ms)
	RMSE	MAE	RMSE	MAE	
STL	0.2773±0.0427	0.1959±0.0333	-	-	12.0095±1.0083
ELLA	0.1503±0.0070	0.1056±0.0028	-	-	**23.1054±3.2531**
ELLA-diver	0.1506±0.0069	0.1058±0.0027	-	-	75.7961±4.8401
ELLA-diver++	0.1505±0.0069	0.1057±0.0027	-	-	75.2880±5.3311
EWC	**0.1379±0.0021**	**0.1014±0.0027**	-	-	-
UFLR	**0.1301±0.0064**	**0.0909±0.0064**	0.1307±0.0065	0.0915±0.0068	**22.2767±3.1511**
UFLR-Ac	**0.1298±0.0064**	**0.0905±0.0062**	0.1304±0.0064	0.0911±0.0066	76.3195±5.7989

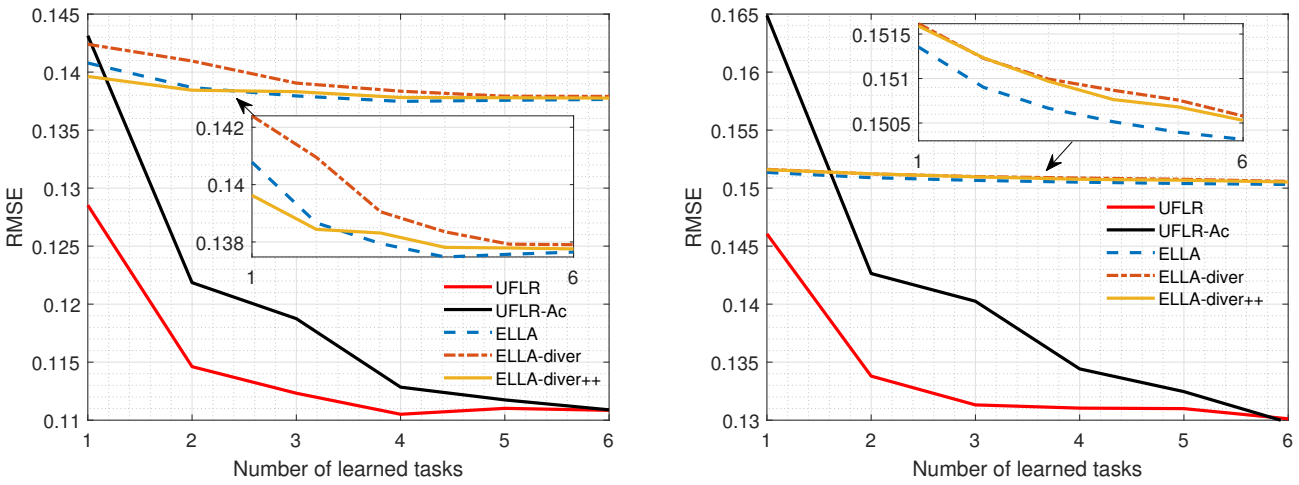

Fig. 8. Comparison of test RMSE performance versus number of tasks learned for (a) Alzheimer-ADAS, and (b) Alzheimer-MMSE.

more important role than the task model in this case. This may be because the dimension of input in this case is much larger. According to Fig. 7, we set $\rho = 5$ for our models to achieve the best model prediction and unsupervised transfer accuracy.

The test performance comparison of various models for Alzheimer-ADAS and Alzheimer-MMSE datasets are presented in Table 5 and Table 6, respectively. It can be seen that EWC attains the second best prediction accuracy. Again, our methods achieve the best prediction accuracy, and our UFLR imposes the lowest ACTpT. More significantly, the unsupervised transfer accuracies of our models are similar to their model prediction accuracies. This makes sense because in this case prediction is mainly contributed by unsupervised feature, and the unsupervised transfer accuracy should be comparable to the model prediction. The test RMSE learning curves as the functions of the number of tasks learned are depicted in Fig. 8 for various lifelong models. Observe that the test RMSEs of our methods decrease much more rapidly as more tasks are learned compared with the ELLA-based methods, which again demonstrates the superior learning capability of our models. Also observe that the active task selection does not seem to help to speed up learning. This is may be because of very limited number of training tasks.

5.5 Discussion of the Algorithm

Three real-world regression datasets from different application scenarios demonstrate the superiority of the proposed unsupervised transfer aided lifelong regression framework. Our proposed method not only consistently attains the best modeling accuracy compared with existing lifelong regression methods, but also provides important capacity of learn-

ing and prediction of new tasks with only input data. Note that the best existing unsupervised transfer aided strategy TaDell [36] can not be applied to our real-world regression benchmarks datasets. TaDell relies on the zero-shot transfer based on the so-called task descriptors. For it to work, these domain-specific task descriptors, which must characterize the underlying dynamics of data in individual tasks well, need to be hand crafted first. For simple engineering systems, some basic system parameters, such as length, mass, damping constant, etc., may be used as task descriptors because they define the system's underlying dynamics and have a close relation to the data characteristics. However, for most real-world tasks, seeking such appropriate and unified descriptors to identify different tasks requires in-depth cross-domain knowledge, which is generally impossible to achieve. By contrast, our proposed method learns new task and performs the unsupervised knowledge transfer with only input data, which is generally applicable to many applications. In terms of computational efficiency, the experimental results have demonstrated that our method has lower online time cost than the efficient ELLA. Most importantly, the computation cost is independent with the number of tasks, which is clearly affordable when massive tasks are received over long-time scales.

This work mainly focus on lifelong regression problem, hence we only use regression datasets to evaluate our algorithm. To our best knowledge, most existing lifelong/continual learning algorithms, such as the works of [47], [48], [49], [50], [51], only focus on object recolonization or classification, and they are not applicable to regression learning. By contrast, our proposed UFLR algorithm is specifically designed for regression problem, which fills a gap in the field. It is also worth mentioning that our method has a similar flexible structure with ELLA, and it can also be extended to address classification problem. In this case, we can compare the proposed framework with more state-of-the-art lifelong learning algorithms on classification benchmarks. However, we emphasize again that this research is devoted specifically for lifelong regression problems with consecutive unlabeled tasks, and for such challenging application area, our proposed framework shows considerable advantages over the existing state-of-the-art, as evidenced by the experimental results.

6 Conclusions and Future Works

This paper has proposed an effective lifelong regression framework capable of learning new consecutive tasks without desired output data. Specifically, during training phase, the input data for each task are encoded as feature vectors while both the input and output data are used to construct a single-task predictor using LS estimator. The unsupervised features and task predictor's parameters are factorized into two dictionaries that are coupled by a joint sparse coding. The leaner can also actively choose next training task to learn based on how poorly the current dictionary encodes the selected tasks. When new task arrives, the learner can perform either model prediction or unsupervised transfer depending on whether the task's data are labeled or not. Even if the new task is not labeled, the learner can still recover the task predictor using unsupervised features via knowledge transfer. This novel capability has ensured that our proposed lifelong regression framework has better generalization performance over the existing state-of-the-art lifelong regression models., which has been validated with applications to three real-world lifelong regression problems.

Defining an appropriate unsupervised feature for unlabeled task remains an open question. In our proposed framework, we simply use the mean values of input data as the feature vectors. In future, we will explore alternative more advanced features provided by various unsupervised learning methods to further improve the unsupervised transfer accuracy. Another interesting future direction is to extend this lifelong regression framework to nonlinear regression, since many real-life tasks are very complex and have strong nonlinearities. Hence, nonlinear models, such as neural networks, can potentially be used in the proposed scheme to replace linear regression. Note that our method has a flexible model structure, and it can also be extended to address classification problem by replacing the base linear regression predictor with simple classifier. This future extension enables our framework to be applicable to wider range of lifelong learning scenarios where labeling new tasks is challenging.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[3] S. Bickel, C. Sawade, and T. Scheffer, "Transfer learning by distribution matching for targeted advertising," in *Proc. NIPS 2008* (Vancouver, BC, Canada), Dec. 8-10, 2008, pp. 145–152.

[4] Y. Xu, *et al.*, "A unified framework for metric transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1158–1171, Jun. 2017.

[5] M. Long, *et al.*, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076 –1089, May 2014.

[6] M. Rostami, S. Kolouri, E. Eaton, and K. Kim, "Deep transfer learning for few-shot SAR image classification," *Remote Sensing*, vol. 11, no. 11, pp. 1374–1388, Jun. 2019.

[7] X. Liao and L. Carin, "Radial basis function network for multi-task learning," in *Proc. NIPS 2005* (Vancouver, BC, Canada), Dec. 5-8, 2005, pp. 792–802.

[8] L. Jacob, J. Vert, and F. Bach, "Clustered multi-task learning: A convex formulation," in *Proc. NIPS 2008* (Vancouver, BC, Canada), Dec. 8-10, 2008, pp. 745–752.

[9] A. Jalali, S. Sanghavi, and C. Ruan, and P. D. Ravikumar, "A dirty model for multi-task learning," in *Proc. NIPS 2010* (Vancouver, BC, Canada), Dec. 6-11, 2010, pp. 964–972.

[10] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Proc. NIPS 2011* (Granada, Spain), Dec. 12-17, 2011, pp. 702–710.

[11] Q. Zhou and Q. Zhao, "Flexible clustered multi-task learning by learning representative tasks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 266–278, Feb. 2016.

[12] Z. Chen and B. Liu, *Lifelong Machine Learning* (2nd edition), Morgan & Claypool Publishers, 2018.

[13] G. Sun, *et al.*, "Representative task self-selection for flexible clustered lifelong learning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1467–1481, Apr. 2022.

[14] G. Sun, *et al.*, "Hierarchical lifelong machine learning with watchdog," *IEEE Trans. Big Data*, vol. 9, no. 1, pp. 63–74, Jan./Feb. 2023.

[15] G. Sun, *et al.*, "What and how: Generalized lifelong spectral clustering via dual memory," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3895-3908, Jul. 2023.

[16] F. Ye and A. G. Bors, "Lifelong mixture of variational autoencoders," *IEEE Trans. Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 461–474, Jan. 2023.

[17] G. Sun, *et al.*, "Lifelong metric learning," *IEEE Trans. Cybernetics*, vol. 49, no. 8, pp. 3168–3179, Aug. 2019.

[18] S. Chen, X. Hong, and C. J. Harris, "Grey-box radial basis function modelling: The art of incorporating prior knowledge," in *Proc. SSP09* (Cardiff, UK), Aug. 31–3 Sep 3, 2009, pp. 377–380.

[19] X. Hong and S. Chen, "A new RBF neural network with boundary value constraints," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 39, no. 1, pp. 298–303, Feb. 2009.

[20] S. Chen, X. Hong, and C. J. Harris, "Grey-box radial basis function modelling," *Neurocomputing*, vol. 74, no. 10, pp. 1564–1571, 2011.

[21] P. Ruvolo and E. Eaton, "Online multi-task learning via sparse dictionary optimization," in *Proc. AAAI 2014* (Québec City, Québec, Canada), Jul. 27-32, 2014, pp. 2062–2068.

[22] P. Ruvolo and E. Eaton, "ELLA: An efficient lifelong learning algorithm," in *Proc. ICML 2013* (Atlanta, GA, USA), Jun. 16-21, 2013, pp. 507–515.

[23] H. B. Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, "Online multi-task learning for policy gradient methods," in *Proc. ICML 2014* (Beijing, China), Jun. 21-26, 2014, pp. 1206–1214.

[24] H. B. Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, "Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment," in *Proc. AAAI 2015* (Austin, TX, USA), Jan. 25-30, 2015, pp. 2504–2510.

[25] H. B. Ammar, R. Tutunov, and E. Eaton, "Safe policy search for lifelong reinforcement learning with sublinear regret," in *Proc. ICML 2015* (Lille, France), Jul. 6-11, 2015, pp. 2361–2369.

[26] H. B. Ammar, E. Eaton, J. M. Luna, and E. Eaton, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *Proc. IJCAI 2015* (Buenos Aires, Argentina), Jul. 25-31, 2015, pp. 3345–3351.

[27] J. Mendez, B. Wang, and E. Eaton, "Lifelong policy gradient learning of factored policies for faster training without forgetting," in *Proc. NIPS 2020* (Vancouver, BC, Canada), Dec. 6-12, 2020, pp. 1–12.

[28] M. Rostami, S. Kolouri, K. Kim, and E. Eaton, "Multi-agent distributed lifelong learning for collective knowledge acquisition," in *Proc. AAMAS 2018* (Stockholm, Sweden), Jul. 10-15, 2018, pp. 712–720.

[29] J. Kirkpatrick, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[30] L. Liu, *et al.*, "IncDet: In defense of elastic weight consolidation for incremental object detection," *IEEE Trans. Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2306–2319, Jun. 2021.

[31] D. Variš, O. Bojar, "Unsupervised pretraining for neural machine translation using elastic weight consolidation," in *Proc. 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (Florence, Italy), Jul. 28-Aug. 2, 2019, pp. 130–135.

[32] Y. Li, R. Zhang, J. Lu, and E. Shechtman, "Few-shot image generation with elastic weight consolidation," in *Proc. NIPS 2020* (Vancouver, BC, Canada), Dec. 6-12, 2020, pp. 15885–15896.

[33] P. Ruvolo and E. Eaton, "Active task selection for lifelong machine learning," in *Proc. AAAI 2013* (Bellevue, Washington, USA), Jul. 14-18, 2013, pp. 862–868.

[34] G. Sun, Y. Cong, and X. Xu, "Active lifelong learning with 'watchdog'," in *Proc. AAAI 2018* (New Orleans, LA, USA), Feb. 2-7, 2018, pp. 4107–4114.

[35] D. Isele, M. Rostami, and E. Eaton, "Using task features for zero-shot knowledge transfer in lifelong learning," in *Proc. IJCAI 2016* (New York, USA), Jul. 9-15, 2016, pp. 1620–1626.

[36] M. Rostami, D. Isele, and E. Eaton, "Using task descriptions in lifelong machine learning for improved performance and zero-shot transfer," *J. Artificial Intelligence Research*, vol. 67, pp. 673–704, Mar. 2020.

[37] M. Rostami, *et al.*, "Zero-shot image classification using coupled dictionary embedding," *Machine Learning with Applications*, vol. 8, Article No.100278, pp. 1–11, Jun. 2022.

[38] M. Long, H. Zhu, J. Wang, and M. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. NIPS 2016* (Barcelona, Spain), Dec. 5-10, 2016, pp. 136–144.

[39] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. CVPR 2018* (Salt Lake City UT, USA), Jun. 18-22, 2018, pp. 3723–3732.

[40] P. Morerio, J. Cavazza, and V. Murino, "Minimal-entropy correlation alignment for unsupervised deep domain adaptation," in *Proc. ICLR 2018* (Vancouver, BC, Canada), Apr. 30-May 3, 2018, pp. 1–15.

[41] B. Gholami, *et al.*, "Unsupervised multi-target domain adaptation: An information theoretic approach," *IEEE Trans. Image Processing*, vol. 29, pp. 3993–4002, Feb. 2020.

[42] T. Isobe, *et al.*, "Multi-target domain adaptation with collaborative consistency learning," in *Proc. CVPR 2021*, Jun.19-25, 2021, pp. 8187–8196.

[43] C. H. Yao, *et al.*, "Federated multi-target domain adaptation," *Proc. WACV 2022* (Waikoloa, HI, USA), Jan. 3-8, 2022, pp. 1424–1433.

[44] S. Roy, *et al.*, "Curriculum graph co-teaching for multi-target domain adaptation," *Proc. CVPR 2021*, Jun.19-25, pp. 5351–5360.

[45] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. ICLR 2017* (Toulon, France), Apr. 24–26, 2017, pp. 3987–3995.

[46] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," arXiv:1511.05952, 2015.

[47] S. Tang, P. Su, D. Chen, and W. Ouyang, "Gradient regularized contrastive learning for continual domain adaptation," in *Proc. AAAI 2021*, Feb. 2-9, 2021, pp. 2–13.

[48] A. Taufique, C. S. Jahan, and A. Savakis, "ConDA: continual unsupervised domain adaptation," arXiv:2103.11056, 2021.

[49] D. Li, *et al.*, "Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation," *Proc. NAACL HLT 2019* (Seattle, WA, USA), Jun. 2-7, 2022, pp. 5441–5454.

[50] M. Rostami, "Lifelong domain adaptation via consolidated internal distribution," *Proc. NeurIPS 2021*, Dec.6-14, 2021, pp. 11172–11183.

[51] Z. Huang, *et al.*, "Lifelong unsupervised domain adaptive person re-identification with coordinated anti-forgetting and adaptation," *Proc. CVPR 2022* (New Orleans, LA, USA), Jun. 19-24, 2022, pp. 14288–14297.

[52] A. Kumar and H. Daume, "Learning task grouping and overlap in multi-task learning," in *Proc. ICML 2012* (Edinburgh, Scotland, UK), Jun. 26-Jul. 1, 2012, pp. 1723–1730.

[53] A. Tsanas, M. Little, P. McSharry and L. Ramig, "Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests," *Nature Precedings*, vol. 67, pp. 673–704, Oct. 2009.

[54] M. W. Weiner, *et al.*, "The Alzheimer's disease neuroimaging initiative: Progress report and future plans," *Alzheimer's & Dementia*, vol. 6, no. 3, pp. 202–211, Oct. 2010.

[55] J. Zhou, J. Liu, V. A. Narayan, and J. Ye, "Modeling disease progression via fused sparse group Lasso," in *Proc. SIGKDD 2012* (Beijing, China), Aug. 12-16, 2012, pp. 1095–1103.

[56] J. Zhou, J. Liu, V. A. Narayan, and J. Ye, "Modeling disease progression via multi-task learning," *NeuroImage*, vol. 78, pp. 233–248, Sep. 2013.

[57] X. Wang, *et al.*, "Cognitive assessment prediction in Alzheimer's disease by multi-layer multi-target regression," *Neuroinformatics*, vol. 16, no. 3, pp. 285–294, May 2018.

[58] P. Cao, *et al.*, "Sparse shared structure based multi-task learning for MRI based cognitive performance prediction of Alzheimer's disease," *Pattern Recognition*, vol. 72, pp. 219–235, Dec. 2017.

[59] X. Liu, *et al.*, "Modeling Alzheimer's disease cognitive scores using multi-task sparse group Lasso," *Computerized Medical Imaging and Graphics*, vol. 66, pp. 100–114, Jun. 2018.

Tong Liu (IEEE Member) received the B.S. degree in Automation, and the Ph.D. degree in Control Theory and Engineering in 2016 and 2021, respectively, from Chongqing University, China. He is currently a Lecturer in Pervasive Data Science with the Department of Computer Science, University of Sheffield, U.K. He was a Postdoctoral Research Associate with the Sargent Centre for Process Systems Engineering, Imperial College London. He also partly worked as an AI researcher with the Shell. Before he was with the University of Sheffield, he was a visiting researcher with the School of Electronics and Computer Science, University of Southampton. His primary research interest is to develop sustainable and lifelong learning machines for large-scale data analytics, so as to support intelligent decision-making. Application areas include industrial automation, precision agriculture and healthcare informatics.

Xulong Wang received the B.Sc. degree in computer science from XI'an University of Finance and Economics, Xi'an, China, in 2018, the M.Sc. degree in the National Pilot School of Software, Yunnan University, Kunming, China, in 2021. He is currently working toward the Ph.D. degree in the Department of Computer Science, University of Sheffield, Sheffield, U.K. His research interests include artificial intelligence, lifelong learning, continual learning, multi-task learning and pervasive data science.

Po Yang (IEEE Senior Member) is currently an Associate Professor in Large-scale Data Fusion with the Department of Computer Science, University of Sheffield, Sheffield, U.K. He received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2004, the M.Sc. degree in computer science from Bristol University, Bristol, U.K., in 2006, and the Ph.D. degree in electronic engineering from the University of Staffordshire, Stafford, U.K., in 2011. He was a Postdoctoral Research Fellow with the Department of Computing, Bedfordshire University, Luton, U.K. Before he was with the Bedfordshire University, he was a Research Assistant with the University of Salford. His main research interests include pervasive computing, data science, mobile intelligence, computer vision, GPU, and parallel computing.

Sheng Chen (IEEE Life Fellow) received his BEng degree from the East China Petroleum Institute, Dongying, China, in 1982, and his PhD degree from the City University, London, in 1986, both in control engineering. In 2005, he was awarded the higher doctoral degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK. From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with the School of Electronics and Computer Science, the University of Southampton, UK, where he holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, evolutionary computation methods and optimization. He has published over 700 research papers. Professor Chen has 19,800+ Web of Science citations with h-index 62 and 38,600+ Google Scholar citations with h-index 83. Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of Asia-Pacific Artificial Intelligence Association and a Fellow of IET. He is one of the original ISI highly cited researcher in engineering (March 2004).

Chris J. Harris received the B.Sc. degree from the University of Leicester, Leicester, U.K., in 1967, the M.A. degree from the University of Oxford, Oxford, U.K., in 1976, and the Ph.D. and D.Sc. degrees from the University of Southampton, Southampton, U.K., in 1972 and 2001, respectively. He held senior academic appointments with the Imperial College London, London, U.K., the University of Oxford, and the University of Manchester, Manchester, U.K. He was a Deputy Chief Scientist with the U.K. Government. He is currently an Emeritus Research Professor with the University of Southampton. Professor Harris was awarded the IEE senior Achievement Medal for Data Fusion research and the IEE Faraday Medal for distinguished international research in Machine Learning. He was elected to Fellow of the UK Royal Academy of Engineering in 1996. He is the co-author of over 600 scientific research papers during a 60 year research career.