Proceedings Paper:
Liu, T., Wang, X., Huang, H. et al. (2023) Weak regression enhanced lifelong learning for improved performance and reduced training data. In: CIKM '23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023), 21-25 Oct 2023, Birmingham, United Kingdom. Association for Computing Machinery, pp. 1587-1596. ISBN: 979-8-4007-0124-5.

https://doi.org/10.1145/3583780.3615108

Weak Regression Enhanced Lifelong Learning for Improved Performance and Reduced Training Data

ABSTRACT

As an emerging learning paradigm, lifelong learning intends to solve multiple consecutive tasks over long-time scales upon previously accumulated knowledge. When facing with a new task, existing lifelong learning approaches need first gather sufficient training data to identify task relationships before knowledge transfer can succeed. However, annotating large number of training data persistently for every coming task is time-consuming, which can be prohibitive for real-world lifelong regression problems. To reduce this burden, we propose to incorporate weak regression into lifelong learning so as to enhance training data and improve predictive performance. Specifically, the weak prediction is first produced by single-task predictor, which is encoded as feature vectors that contain essential prior output information. This weak regression is further linked with task model via coupled dictionary learning. The integration of weak regression and task model can facilitate both cross-task and inter-task knowledge transfer, thus improving the overall performance. More critically, the weak regression can backup the task model especially when there is insufficient training data to construct an accurate model. Three real-world datasets are used to evaluate the effectiveness of our proposed method. Results show that our method outperforms existing lifelong models and single-task models even if training data is minimal.

CCS CONCEPTS

• **Computing methodologies** → *Online learning settings*; **Learning paradigms**.

KEYWORDS

lifelong learning, weak regression, coupled dictionary learning, knowledge transfer

ACM Reference Format:

. 2023. Weak Regression Enhanced Lifelong Learning for Improved Performance and Reduced Training Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (CIKM '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Most machine learning methods take a "single-shot" approach in which knowledge is not retained between learning problems. However, we are easily encountered multiple consecutive learning tasks in real-life. This advances an emerging machine learning paradigm called *lifelong learning*, in which a lifelong learner continually learns to solve multiple tasks through knowledge transfer from previously learned models, and revision of stored source knowledge from new upcoming task [2, 22, 23]. Due to its high efficiency in handling massive tasks, lifelong learning has found wide-ranging data modeling applications, including regression, classification and clustering [3, 4, 6, 17, 21, 24, 26, 27]. For regression problem, existing lifelong learning approaches need sufficient training data to modeling task relationships before knowledge transfer can succeed. However, for most real-world regression applications, such as soft sensing [7–9, 34] and pervasive healthcare [12–14], high-quality annotations are difficult to obtain. Hence, it is practical prohibitive to labeling large number of training data for every coming task for a regression model to learn. This motivates our current work to develop an effective lifelong regression model that combats this limitation.

Among lifelong learning community, the efficient lifelong learning algorithm (ELLA) framework is one of the most popular approaches [20]. ELLA learns and maintains a knowledge repository as a shared basis for all tasks, supporting knowledge transfer among task models. When a new task arrives, it transfers knowledge through the shared basis to learn the new model, and refines the basis with knowledge learned from new task. By updating the basis over time, newly acquired knowledge is integrated into the knowledge repository, thereby improving previously learned model. To further improve its scalability, curriculum learning strategy is integrated into this framework, enabling the lifelong learner to actively select task order so as to maximize overall performance using as few tasks as possible [19, 25]. However, the active task selection requires the tasks candidate pool is known as a prior. This holds not true for many practical situations where only one task comes at each time step, and the learner requires to make quick response as it comes. Another interesting work is to extend this framework to reinforcement learning with policy gradient (PG-ELLA) method [1, 11]. With the ability to knowledge transfer between multiple sequential decision making tasks, PG-ELLA is capable of rapid learning of control policies for new system.

One critical issue regarding the ELLA-like methods is that when faced with a new task, the learner needs first gather sufficient training (labeled) data before bootstrapping a model via knowledge transfer. This need for training data becomes problematic for lifelong regression application, as labeling large number of training data persistently for every coming task is time-consuming, and often the learner is expected to rapidly predict new task without delay to wait for labeling task. To overcome this restriction, one famous early work of [5] uses high-level task features or descriptors to model the inter-task relationships in lifelong reinforcement learning, namely task descriptor lifelong learning (TaDeLL). Unlike the PG-ELLA that uses only one knowledge repository for task's

CIKM '23, October 21–25, 2023, Birminghan, UK

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

policy [1], TaDeLL employs a coupled repository to integrate high-level task descriptors with task policies. Results show that using task descriptors improves the performance of learned task policies, and more importantly, it enables accurately predicting the new task policy even without training data via zero-shot knowledge transfer [5]. TaDeLL is further extended for regression problem in [18], which eliminates the need to gather data for predicting new task. This 'learning without 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 [18] 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. 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 in lifelong learning communities, how to efficiently utilize both labeled and unlabeled data in characterizing and learning each consecutive task with improved performance is an important challenge. This motivates our current work to develop an effective lifelong regression model that enables to learn new task with reduced training set, thus reducing the burden for large number of annotations. We explore the use of weak prediction to enhance knowledge transfer between multiple regression tasks and improve the overall predictive performance. Our approach to incorporate weak prediction into lifelong regression learning is general, as it does not need domain-specific task descriptor that requires human expert. Instead, we use weak predictions that is easily provided by single-task predictors, encoding them as feature vectors and treating these prior predictions as side information to augment training data for individual tasks. In order to obtain a conceivable and robust weak prediction, we construct the single-task predictor by partial least square (PLS) algorithm. The PLS model is capable of handing data with high dimensionality and multicollinearity [16], enabling providing higher modeling accuracy than classic least squares regression used in existing lifelong learning methods. The idea of using weak regression to enhance predictive model is previously explored in [10], where weak prediction is incorporated into deep neural networks to extract better output-relevant features, thus improving the final predictive accuracy. In comparison, our method aims to learn a mapping from weak prediction onto the task model, enabling the leaner to achieve knowledge transfer with less training samples. Similar to [5, 18], we use coupled dictionary learning to model the inter-task relationships between weak prediction and task model, enabling them to complement to each other. Therefore, when there is insufficient training data to construct an accurate task model, the weak regression acts as a backup to supplement this shortage, hence to guarantee the achievable performance. Our novel contributions can be summarized as follows:

(1) We construct single-task predictor by the PLS algorithm, treating it as weak predictor to provide a rough prediction of target value using only unlabeled data for each task. This weak prediction provides essential and vital output-relevant information for knowledge transfer.
(2) We integrate weak regression and task model by coupled dictionary learning, so as to facilitate both cross-task and inter-task knowledge transfer. First, dual knowledge transfer from two spaces can better identify cross-task relationships, thus improving the overall performance. More importantly, the weak regression can make up for the inaccuracy of task model caused by insufficient training data. This capacity is very important in the online setting of lifelong regression process, as it reduces the burden of labeling large number of training data for consecutive tasks.
(3) We analysis the method theoretically, and use three real-world datasets to validate its effectiveness. Results show that our method outperforms existing lifelong learning models even if training data is minimal.

2 PRELIMINARIES

2.1 Problem definition

For lifelong regression problem, the lifelong learner faces a series of regression tasks $\{\mathbb{Z}^{(1)}, \mathbb{Z}^{(2)}, \ldots, \mathbb{Z}^{(T_{\max})}\}$. Each regression task $\mathbb{Z}^{(t)} = \left(f^{(t)}, \boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}\right)$ is specific by a function mapping $f^{(t)} : \boldsymbol{X}^{(t)} \mapsto \boldsymbol{y}^{(t)}$ from input space $\boldsymbol{X}^{(t)} \in \mathbb{R}^d$ to the output space $\boldsymbol{y}^{(t)} \in \mathbb{R}$. To learn $f^{(t)}$, the learner is given n_t training input data $\boldsymbol{X}^{(t)} \in \mathbb{R}^{n_t \times d}$ and output data $\boldsymbol{y}^{(t)} \in \mathbb{R}^{n_t}$. For brevity, $\left(\boldsymbol{x}_i^{(t)}, y_i^{(t)}\right)$ denotes the ith labeled training sample for task t. The lifelong learner does not know the total task number $T_{\max}$, task order or task distribution in a prior.

At each time step, the lifelong learner receives a batch of training data $\left(\boldsymbol{x}_i^{(t)}, y_i^{(t)}\right)_{i=1}^{n_t}$ for task t. Let T denote the number of tasks the learner has encountered so far. At anytime, the learner may be asked to make predictions on data from any previous task. Its goal is to consecutively construct a set of task models $\{\hat{f}^{(1)}, \ldots, \hat{f}^{(T)}\}$ such that each $\hat{f}^{(t)}$ will approximate $f^{(t)}$ to make accurate prediction on new unseen data, and new model $\hat{f}^{(t)}$ can be added efficiently when learner encountering new task. Ideally, knowledge learned from previous tasks $\{\mathbb{Z}^{(1)}, \ldots, \mathbb{Z}^{(T-1)}\}$ should accelerate training and improve performance on each new task $\mathbb{Z}^{(T)}$. Also, the lifelong learner should scale effectively to large number of tasks, learning new task rapidly with minimal data.

2.2 Efficient lifelong learning

The well-known ELLA is developed to operate in this lifelong learning setting. To be specific, ELLA learns and maintains a shared knowledge repository $\boldsymbol{L} \in \mathbb{R}^{d \times k}$, which forms a basis for all task models and facilitates knowledge transfer between tasks. For each task t, ELLA learns a model $\hat{f}^{(t)}(\boldsymbol{X}) = \hat{f}(\boldsymbol{X}; \boldsymbol{\theta}^{(t)})$ that is parameterized by a d-dimensional task-specific 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 repository $\boldsymbol{L}$ stores chunks of knowledge that are useful for multiple tasks, and

the sparse code $\boldsymbol{s}^{(t)}$ extracts relevant pieces of knowledge for a particular task. Hence, this model vector factorization enables effective knowledge transfer among tasks.

Given the training data for each task, ELLA minimizes the predictive error of each task model while encouraging shared task structure by optimizing this objective:

$$\min_{L,S} \frac{1}{T}\sum_{t=1}^{T}\left\{\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{J}\big(f(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}),y_i^{(t)}\big)+\mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right\}+\lambda\left\|\boldsymbol{L}\right\|_{\mathrm{F}}^2, \tag{1}$$

where $\boldsymbol{S}=[\boldsymbol{s}^{(1)},\ldots,\boldsymbol{s}^{(T)}]$ is the matrix of sparse vector, $\|\bullet\|_{\mathrm{F}}$ is the Frobenius norm, which regularizes the basis $\boldsymbol{L}$ complexity. The L_1 norm is used to control the sparsity of $\boldsymbol{s}^{(t)}$, and μ and λ are regularization parameters.

To solve this objective in a lifelong learning setting, ELLA tasks a second-order Taylor expansion to approximate the objective around an estimate $\hat{\boldsymbol{\theta}}^{(t)}=\arg\min_{\boldsymbol{\theta}}\frac{1}{n_t}\sum_{t=1}^{n_t}\mathcal{J}\big(f(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}),y_i^{(t)}\big)$ of the single-task model parameters for each task, and update only the coefficients $\boldsymbol{s}^{(t)}$ for the current task at each time step. This process enables solving $\boldsymbol{L}$ and $\boldsymbol{S}$ efficiently in an online manner, which yields the following recursive update equations that approximate the result of Eq. (1):

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

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

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

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

where $\|\boldsymbol{v}\|_{\boldsymbol{A}}^2=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}$, the symbol $\otimes$ denotes the Kronecker product, $\boldsymbol{\Upsilon}^{(t)}=\boldsymbol{\Upsilon}(\hat{\boldsymbol{\theta}}^{(t)})$ is the Hessian matrix of the loss $\mathcal{J}(\boldsymbol{\theta}^{(t)})$, $\boldsymbol{I}_{kd}$ is the $kd\times kd$ identity matrix, $\boldsymbol{A}_L$ is initialized to be a $kd\times kd$ zero matrix, and $\boldsymbol{b}_L\in\mathbb{R}^{kd}$ is initialized to zeros.

At each time step when encountering new task and receiving corresponding training data, the ELLA performs two-step model adaptation by updating $\boldsymbol{s}^{(t)}$ and $\boldsymbol{L}$. In order to compute $\boldsymbol{s}^{(t)}$, it first computes an optimal model vector $\hat{\boldsymbol{\theta}}^{(t)}$ using training data from task t. The task model is normally constructed to avoid huge computational complexity. The classic ELLA simply employs a linear regression model and its optimal model parameter $\hat{\boldsymbol{\theta}}^{(t)}$ is solved by the least squares (LS) estimator. However, data from real-world tasks are typically high-dimensional with strong co-linearities. A commonly used way of eliminating data co-linearity is to transform the original data onto the latent subspace by means of PLS. The following section will briefly introduce PLS as an alternative to LS regression under the lifelong learning framework.

2.3 Partial least square

The PLS aims to predict output $\boldsymbol{y}$ using latent variables in $\boldsymbol{X}$ instead of the original input. One major benefit of using PLS is to reduce dimensions, avoid co-linearity in input data, and shrink the variance of prediction. We use the same notations as in the previous section, where $\boldsymbol{X}^{(t)}\in\mathbb{R}^{n_t\times d}$ and $\boldsymbol{y}^{(t)}\in\mathbb{R}^{n_t}$ are the input and output data for task t. Assuming that the modeling data have been mean-centered and appropriate scaled, the PLS algorithm models the mapping relationship between $\boldsymbol{X}^{(t)}$ and $\boldsymbol{y}^{(t)}$ as

$$\boldsymbol{y}^{(t)}\approx\boldsymbol{X}^{(t)}\hat{\boldsymbol{\theta}}^{(t)}, \tag{6}$$

where $\hat{\boldsymbol{\theta}}^{(t)}=\big(\boldsymbol{X}^{(t)\mathrm{T}}\boldsymbol{X}^{(t)}\big)^{\dagger}\boldsymbol{X}^{(t)\mathrm{T}}\boldsymbol{y}^{(t)}\in\mathbb{R}^d$ is the regression coefficients, and $(\bullet)^{\dagger}$ denotes the generalized inverse operator. In the PLS algorithm, data matrices $\boldsymbol{X}^{(t)}$ and $\boldsymbol{y}^{(t)}$ are first decomposed respectively as

$$\boldsymbol{X}^{(t)}=\sum_{i=1}^{a}\boldsymbol{t}_i\boldsymbol{p}_i^{\mathrm{T}}+\boldsymbol{E}_x=\boldsymbol{T}\boldsymbol{P}^{\mathrm{T}}+\boldsymbol{E}_x, \tag{7}$$

$$\boldsymbol{y}^{(t)}=\sum_{i=1}^{a}\boldsymbol{u}_i q_i^{\mathrm{T}}+\boldsymbol{E}_y=\boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}}+\boldsymbol{E}_y, \tag{8}$$

where a denotes the number of latent variables, $\boldsymbol{T}=[\boldsymbol{t}_1,\ldots,\boldsymbol{t}_a]^{\mathrm{T}}\in\mathbb{R}^{n_t\times a}$ and $\boldsymbol{U}=[\boldsymbol{u}_1,\ldots,\boldsymbol{u}_a]^{\mathrm{T}}\in\mathbb{R}^{n_t\times a}$ represent the score matrices, and $\boldsymbol{P}=[\boldsymbol{p}_1,\ldots,\boldsymbol{p}_a]^{\mathrm{T}}\in\mathbb{R}^{d\times a}$ and $\boldsymbol{Q}=[q_1,\ldots,q_a]^{\mathrm{T}}\in\mathbb{R}^a$ denote the loading matrices of $\boldsymbol{X}^{(t)}$ and $\boldsymbol{y}^{(t)}$, respectively, while $\boldsymbol{E}_x\in\mathbb{R}^{n_t\times d}$ and $\boldsymbol{E}_y\in\mathbb{R}^{n_t}$ are the respective error matrices. The algorithm details can be found in [15].

3 WEAK REGRESSION ENHANCED LIFELONG LEARNING

Although the PLS-based task model can improve predictive accuracy by eliminating co-linearities reside in task data, it is still challenging when dealing with insufficient training data for consecutive tasks. In order to alleviate lifelong leaner's dependence on training data and improve the overall predictive accuracy, we propose to incorporate weak regression into lifelong learning via sparse coding with a coupled dictionary, thus enabling the weak regression and learned task model to augment each other. Specifically, the weak prediction is easily provided by the single-task predictor or we say 'weak predictor'. The lifelong learner then encodes these weak prediction results as feature vectors that identify each task, treating weak prediction results as side information to augment training data on individual tasks. In order to link weak regression's space with task model's space, we employ two dictionaries that act as knowledge repositories for two spaces, and they are coupled by a joint sparse representation. Because of the learned coupling, the weak regression can compensate the inaccuracy of task model due to insufficient training data. This capacity greatly reduce the burden of labeling large number of training samples for sequential tasks in the lifelong learning setting. The framework of our proposed weak regression enhanced lifelong learning can be seen in Fig. 1.

3.1 Weak regression by PLS

At each time step, the lifelong learner receives a batch of training data $\{\boldsymbol{X}^{(t)},\boldsymbol{y}^{(t)}\}$ for task t. It constructs a single task predictor $f(\boldsymbol{X}^{(t)};\boldsymbol{\theta}^{(t)})=\boldsymbol{X}^{(t)}\boldsymbol{\theta}^{(t)}$ upon the training data using the PLS algorithm, yielding the optimal model parameters:

$$\hat{\boldsymbol{\theta}}^{(t)}=\boldsymbol{X}^{(t)\mathrm{T}}\boldsymbol{U}^{(t)}(\boldsymbol{T}^{(t)\mathrm{T}}\boldsymbol{X}^{(t)}\boldsymbol{X}^{(t)\mathrm{T}}\boldsymbol{U}^{(t)})^{-1}\boldsymbol{T}^{(t)\mathrm{T}}\boldsymbol{y}^{(t)}, \tag{9}$$

where $\boldsymbol{U}^{(t)}$ and $\boldsymbol{T}^{(t)}$ denote the score matrices required to be calculated in PLS for task t. Given the squared-loss function of $\mathcal{J}(\boldsymbol{\theta}^{(t)})$,

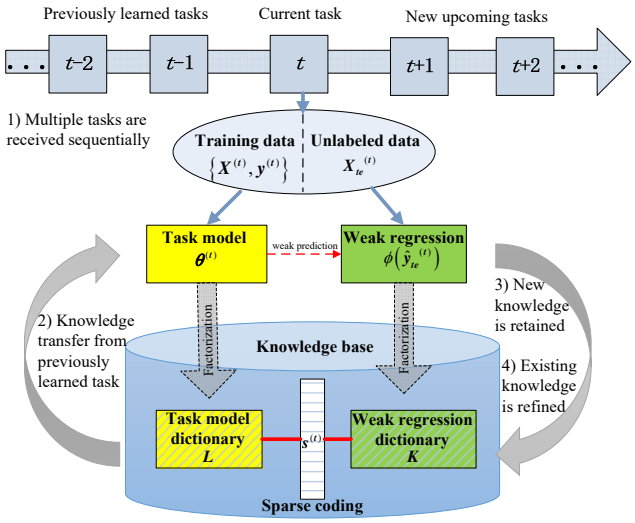


Figure 1: Weak regression enhanced lifelong learning system. Weak regression and task model are integrated to facilitate knowledge transfer in lifelong learning. Model parameters $\boldsymbol{\theta}^{(t)}$ are factored into $\boldsymbol{L}$ and $\boldsymbol{s}^{(t)}$ while weak predictions $\phi(\hat{\boldsymbol{y}}_{te}^{(t)})$ are factored into $\boldsymbol{K}$ and $\boldsymbol{s}^{(t)}$. Since both dictionaries share the same sparse coding $\boldsymbol{s}^{(t)}$, the weak regression is naturally coupled with the task model.

the Hessian $\boldsymbol{\Upsilon}^{(t)}$ of the corresponding loss function around the single task solution $\hat{\boldsymbol{\theta}}^{(t)}$ is:

$$\boldsymbol{\Upsilon}^{(t)} = \frac{1}{2n_t}\boldsymbol{X}^{(t)}\boldsymbol{X}^{(t)\mathrm{T}}, \tag{10}$$

Recalling that the capacity of lifelong learner is to make predictions on newly observed data from any previously learned tasks. Given the unlabeled newly observed input data $\boldsymbol{X}_{te}^{(t)}$ for task t, the single task predictor can provide a preliminary prediction of the target value:

$$\hat{\boldsymbol{y}}_{te}^{(t)} = \boldsymbol{X}_{te}^{(t)}\hat{\boldsymbol{\theta}}^{(t)}, \tag{11}$$

We call this 'weak' prediction for the reason that its prediction $\hat{\boldsymbol{y}}_{te}^{(t)}$ is not our final prediction for the true output $\boldsymbol{y}_{te}^{(t)}$. But this weak prediction provides the essential and vital prior output information for both cross-tasks and inter-task knowledge transfer. This prior output information is particularly important especially when the task model is less precision due to the shortage of training data, as it can be treated as side information to augment training data as well as task model. The predicted value is then encoded as feature vectors $\phi(\hat{\boldsymbol{y}}_{te}^{(t)})$ that is analogous to task model's space, representing weak regression's space. Due to the fact that the sample size for each task can be different, we use the minimal sample size of $\hat{\boldsymbol{y}}_{te}^{(t)}$ as the unified size, so as to guarantee the dimension of weak regression's space is identical for all tasks. In this case, the operator $\phi(\bullet)$ denotes tasking the shortest size of $\hat{\boldsymbol{y}}_{te}^{(t)}$ among all tasks. However, when the sample size for each task varies significantly, using the minimal size of $\hat{\boldsymbol{y}}_{te}^{(t)}$ as the feature vector can only provide limited and incomplete information for task with large sample size, thus failing to fully exploit the advantage of weak regression enhancement. To avoid this, another way is to use statistic features of $\hat{\boldsymbol{y}}_{te}^{(t)}$ rather than itself as the feature vector. In this case, the operator $\phi(\bullet)$ is used to transform original weak prediction $\hat{\boldsymbol{y}}_{te}^{(t)}$ into a set of statistic features:

$$\phi(\hat{\boldsymbol{y}}_{te}^{(t)}) = \begin{Bmatrix} \min(\hat{\boldsymbol{y}}_{te}^{(t)}), & \max(\hat{\boldsymbol{y}}_{te}^{(t)}), & \mathrm{mean}(\hat{\boldsymbol{y}}_{te}^{(t)}), \\ \mathrm{median}(\hat{\boldsymbol{y}}_{te}^{(t)}), & \mathrm{std}(\hat{\boldsymbol{y}}_{te}^{(t)}), & \mathrm{var}(\hat{\boldsymbol{y}}_{te}^{(t)}) \end{Bmatrix} \in \mathbb{R}^6 \tag{12}$$

where min(•), max(•), mean(•), median(•), and std(•) denote the operators of calculating minimum, maxmum, mean, median, and standard deviation, respectively. Although using statistic features can provide complete information about $\hat{\boldsymbol{y}}_{te}^{(t)}$, a potential problem is the dimension of statistic features' space is quite small, and it may have limited impact when the model parameter's space is very large. The choice of weak regression encoding is obviously problem independent, and one can always choice an appropriate encoding strategy according to task characteristics. We have thoroughly analyze this in our experiments.

3.2 Coupled dictionary learning

Most lifelong learning approaches factorize model parameters $\boldsymbol{\theta}^{(t)}$ for each task as a sparse linear combination over a shared basis $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$. Basically, each column of the shared basis $\boldsymbol{L}$ serves as a reusable model component representing a cohesive chunk of knowledge. During online operation, the basis $\boldsymbol{L}$ incrementally update as it learns more tasks. The sparse coefficients $\boldsymbol{S}$ encodes the task model in this shared basis, providing a platform for tasks to share knowledge.

Similar to this, the weak prediction's feature vector $\phi(\hat{\boldsymbol{y}}_{te}^{(t)})$ can also be linearly factorized using a shared basis $\boldsymbol{K} \in \mathbb{R}^{d_w \times k}$ over the weak regression's space. This basis captures relationships among weak predictions for multiple tasks. In order to link two spaces and make them complement to each other, the key is to find task embeddings that are consistent for both spaces. We enforce this by coupling two basis $\boldsymbol{L}$ and $\boldsymbol{K}$, sharing the same sparse coding $\boldsymbol{S}$ to reconstruct both the model parameters and weak predictions. Hence, for task t,

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}, \ \phi(\hat{\boldsymbol{y}}_{te}^{(t)}) = \boldsymbol{K}\boldsymbol{s}^{(t)}, \tag{13}$$

Because we enforce both dictionaries for two spaces to share the same sparse coding $\boldsymbol{s}^{(t)}$, the relevant pieces of information for a task model become coupled with its corresponding weak predictions. The idea of using coupled dictionary learning is originally to link the high-level task descriptions with the learned model to achieve zero-shot transfer for new tasks. Considering a very different aspect of zero-shot knowledge transfer, we use coupled dictionaries to seamlessly connect task model's space with the associated weak predictions' space, so as to achieve training data augmentation as well as improved modeling accuracy.

To optimize the coupled basis $\boldsymbol{L}$ and $\boldsymbol{K}$, we first reformulate the objective Eq. (1) for the coupled dictionaries as

$$\min_{L,K,S} \frac{1}{T}\sum_{t=1}^{T}\left\{\mathcal{J}(\boldsymbol{\theta}^{(t)}) + \beta\left\|\phi(\hat{\boldsymbol{y}}_{te}^{(t)}) - \boldsymbol{K}\boldsymbol{s}^{(t)}\right\|_2^2 + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right\} \tag{14}$$
$$+ \lambda\left(\|\boldsymbol{L}\|_{\mathrm{F}}^2 + \|\boldsymbol{K}\|_{\mathrm{F}}^2\right),$$

where $\mathcal{J}(\boldsymbol{\theta}^{(t)}) = \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{J}(f(\boldsymbol{x}_i^{(t)};\boldsymbol{L}\boldsymbol{s}^{(t)}), y_i^{(t)})$, parameter β balances the task model's fit to the weak regression's fit.

To solve (14) in a lifelong setting, we approximate $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ by a second-order Taylor expansion around $\hat{\boldsymbol{\theta}}^{(t)}$. The optimal parameter $\hat{\boldsymbol{\theta}}^{(t)}$ is easily obtained by the single task predictor in (9). Then we expand $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ for each task around $\hat{\boldsymbol{\theta}}^{(t)}$ as:

$$\mathcal{J}(\boldsymbol{\theta}^{(t)}) \approx \mathcal{J}(\hat{\boldsymbol{\theta}}^{(t)}) + \nabla\mathcal{J}(\hat{\boldsymbol{\theta}}^{(t)})(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}}^{(t)}) + \frac{1}{2}\left\|\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}}^{(t)}\right\|^2_{\Upsilon^{(t)}} \tag{15}$$

where ∇ denotes the gradient operator. The first constant term $\mathcal{J}(\hat{\boldsymbol{\theta}}^{(t)})$ can be suppressed for the purpose of optimization. Also note that $\boldsymbol{\theta}^{(t)}$ is the minimizer of the function $\mathcal{J}(\boldsymbol{\theta}^{(t)})$, $\nabla\mathcal{J}(\hat{\boldsymbol{\theta}}^{(t)})$ should be zero, and hence the second term can be removed. Considering that $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$, the last term of $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ can be rewritten as $\left\|\hat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\right\|^2_{\Upsilon^{(t)}}$.

Approximating $\mathcal{J}(\boldsymbol{\theta}^{(t)})$ leads to a simplified from of (14) as

$$\begin{aligned}\min_{\boldsymbol{L},\boldsymbol{K},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\Big\{\left\|\hat{\boldsymbol{\theta}}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\right\|^2_{\Upsilon^{(t)}} + \beta\left\|\phi(\hat{\boldsymbol{y}}_{te}^{(t)}) - \boldsymbol{K}\boldsymbol{s}^{(t)}\right\|_2^2 \\ + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\Big\} + \lambda\big(\|\boldsymbol{L}\|_{\mathrm{F}}^2 + \|\boldsymbol{K}\|_{\mathrm{F}}^2\big),\end{aligned} \tag{16}$$

we can merge pairs of terms in (16) by defining:

$$\boldsymbol{\Theta}^{(t)} = \begin{bmatrix}\hat{\boldsymbol{\theta}}^{(t)} \\ \phi(\hat{\boldsymbol{y}}_{te}^{(t)})\end{bmatrix},\ \boldsymbol{H} = \begin{bmatrix}\boldsymbol{L} \\ \boldsymbol{K}\end{bmatrix},\ \boldsymbol{\Psi}^{(t)} = \begin{bmatrix}\Upsilon^{(t)} & \boldsymbol{0} \\ \boldsymbol{0} & \beta\boldsymbol{I}_{d_w}\end{bmatrix} \tag{17}$$

where $\boldsymbol{0}$ is the zero matrix, d_w is the dimension of weak prediction $\phi(\hat{\boldsymbol{y}}_{te}^{(t)})$. Hence, the objective (16) can be rewritten in a concise form as

$$\min_{\boldsymbol{H},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left\{\left\|\boldsymbol{\Theta}^{(t)} - \boldsymbol{H}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Psi}^{(t)}} + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right\} + \lambda\|\boldsymbol{H}\|_{\mathrm{F}}^2, \tag{18}$$

This objective function has an identical form with the classic lifelong learning method, and it can be solved efficiently in an online manner. Note that the objective (18) can be decoupled into two optimization problems with a similar form on $\boldsymbol{L}$ and $\boldsymbol{K}$, hence two dictionaries can be updated independently.

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

$$\boldsymbol{s}^{(t)} = \arg\min_{\boldsymbol{s}}\left\|\boldsymbol{\Theta}^{(t)} - \boldsymbol{H}\boldsymbol{s}^{(t)}\right\|^2_{\boldsymbol{\Psi}^{(t)}} + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1, \tag{19}$$

After $\boldsymbol{s}^{(t)}$ is obtained, two dictionaries $\boldsymbol{L}$ and $\boldsymbol{K}$ are updated independently. Taking updating $\boldsymbol{K}$ as an example, the decoupled objective for weak regression can be written as:

$$\min_{\boldsymbol{K},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left\{\left\|\phi(\hat{\boldsymbol{y}}_{te}^{(t)}) - \boldsymbol{K}\boldsymbol{s}^{(t)}\right\|^2_{\beta\boldsymbol{I}_{d_w}} + \mu\left\|\boldsymbol{s}^{(t)}\right\|_1\right\} + \lambda\|\boldsymbol{K}\|_{\mathrm{F}}^2, \tag{20}$$

To update $\boldsymbol{K}$, we null the gradient of Eq. (20) and solve for $\boldsymbol{K}$. This procedure yields the updated column-wise vectorization of $\boldsymbol{K}$ as ${\boldsymbol{A}_K}^{-1}\boldsymbol{b}_K$, and we update $\boldsymbol{A}_K$, $\boldsymbol{b}_K$ and $\boldsymbol{K}$ incrementally as

$$\boldsymbol{A}_K = \boldsymbol{A}_K + \big(\boldsymbol{s}^{(t)}\boldsymbol{s}^{(t)\mathrm{T}}\big) \otimes \beta\boldsymbol{I}_{d_w}, \tag{21}$$

$$\boldsymbol{b}_K = \boldsymbol{b}_K + \mathrm{vec}\big[\boldsymbol{s}^{(t)\mathrm{T}} \otimes \big(\phi(\hat{\boldsymbol{y}}_{te}^{(t)})^{\mathrm{T}}\beta\boldsymbol{I}_{d_w}\big)\big], \tag{22}$$

$$\boldsymbol{K} = \boldsymbol{K} + \mathrm{mat}\big[\big(\frac{1}{T}\boldsymbol{A}_K + \lambda\boldsymbol{I}_{kd_w}\big)^{-1}\frac{1}{T}\boldsymbol{b}_K\big], \tag{23}$$

$\boldsymbol{A}_K$ is initialized to be a $kd_w \times kd_w$ zero matrix, and $\boldsymbol{b}_K \in \mathbb{R}^{kd_w}$ is initialized to zeros.

3.3 Algorithm summary

The proposed weak regression enhanced lifelong learning is summarized in Algorithm 1.

Algorithm 1 Weak regression enhanced lifelong learning

Parameters: Number of latent basis k, regularization parameters μ and λ, coefficient β.
Initialize: Randomly initialize $\boldsymbol{L}$ and $\boldsymbol{K}$. Set $T = 0$.
While some task $\mathbb{Z}^{(t)}$ is available **do**
Set $T = T + 1$.
Collect training input-output data $\{\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}\}$ and unlabeled input data $\boldsymbol{X}_{te}^{(t)}$ from task $\mathbb{Z}^{(t)}$.
Construct a single-task predictor upon training data $\{\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}\}$ using the PLS algorithm.
Compute the optimal model parameter $\hat{\boldsymbol{\theta}}^{(t)}$ and Hessian matrix $\Upsilon^{(t)}$ by Eq. (9) and (10), respectively.
Given the unlabeled input data $\boldsymbol{X}_{te}^{(t)}$, compute weak predictions $\hat{\boldsymbol{y}}_{te}^{(t)}$ based on learned task model by Eq. (11).
Encode weak predictions into feature vector $\phi(\hat{\boldsymbol{y}}_{te}^{(t)})$.
Construct matrices $\boldsymbol{\Theta}^{(t)}$, $\boldsymbol{H}$, and $\boldsymbol{\Psi}^{(t)}$ by Eq. (17).
Solve sparse coding $\boldsymbol{s}^{(t)}$ by objective function Eq. (19).
Update $\boldsymbol{A}_L$, $\boldsymbol{b}_L$, dictionary $\boldsymbol{L}$ by Eqs. (3)-(5), respectively.
Update $\boldsymbol{A}_K$, $\boldsymbol{b}_K$, dictionary $\boldsymbol{K}$ by Eqs. (21)-(23), respectively.
For: $t \in \{1, \ldots, T\}$ **do**: $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$
End while

Convergence analysis: In order to prove the convergence of our proposed method, we use theoretical results in [20]. These results can directly apply to our coupled dictionary learning with weak regression enhancement. The work [20] has proved that the learned dictionary $\boldsymbol{L}$ becomes increasingly stable as it learns more tasks. The result is based on two assumptions: 1) The tuples $(\Upsilon^{(t)}, \hat{\boldsymbol{\theta}}^{(t)})$ are drawn $i, i, d.$ from a distribution with compact support. 2) For all task t, let $\boldsymbol{L}_k$ be the subset of the current dictionary $\boldsymbol{L}_t$, where only columns corresponding to non-zero element of $\boldsymbol{s}^{(t)}$ are included. Then, all eigenvalues of the matrix $\boldsymbol{L}_k^{\mathrm{T}}\Upsilon^{(t)}\boldsymbol{L}_k$ need to be strictly positive.

We incorporate weak regression into lifelong learning framework by changing $\hat{\boldsymbol{\theta}}^{(t)}$ into $\boldsymbol{\Theta}^{(t)}$, $\boldsymbol{L}$ into $\boldsymbol{H}$, and $\Upsilon^{(t)}$ into $\boldsymbol{\Psi}^{(t)}$. Clearly, $\boldsymbol{\Theta}^{(t)}$ and $\boldsymbol{\Psi}^{(t)}$ are constructed by adding deterministic entries as shown in Eq. (17), and they should be drawn $i, i, d.$. Hence, Condition 1 holds for our method. For Condition 2, we can easily analogously form $\boldsymbol{H}_k$. The eigenvalues of $\boldsymbol{H}_k$ are either eigenvalues of $\boldsymbol{L}$ or the parameter β by definition, so they should also be strictly positive. Therefore, both two conditions are met for our proposed method and it should follow the same result as in [20].

Computational complexity: We further analysis the online computational complexity of our method for learning new task. First, the construction of single-task predictor has a cost of $O(\xi(d, n_t))$, where $\xi(\cdot)$ depends on the computation involved in PLS modeling. The PLS model can be computed by either nonlinear iterative PLS algorithm or recursive PLS algorithm, which has proved to be computationally efficiency for online monitoring purpose. The adaptation of single dictionary $\boldsymbol{L} \in \mathbb{R}^{d\times k}$ and sparse coefficient $\boldsymbol{s}^{(t)} \in \mathbb{R}^k$ is $O(k^2d^3)$. We incorporate weak regression into lifelong learning by altering $\boldsymbol{L} \in \mathbb{R}^{d\times k}$ into $\boldsymbol{H} \in \mathbb{R}^{(d+d_W)\times k}$, hence the coupled dictionary adaptation costs $O(k^2(d+d_w)^3)$. This yields an overall cost of adaptation per task $O(\xi(d, n_t) + k^2(d+d_w)^3)$. This is clearly efficiency for online operation, as it is independent of task number and the computation per-task is identical.

4 EXPERIMENTS

In this section, we conduct extensive experiments, including **School**, **Parkinson** and **Alzheimer** datasets, to verify the effectiveness of our proposed lifelong regression model.

4.1 Dataset description

The details of three real-world datasets are as follow:

(1) **School dataset**: the London school dataset has been widely used in lifelong learning of regression problem [20, 25]. It contains examination scores of 15362 students from 139 secondary schools and each school is considered as one regression task. The goal is to predict the scores for students according to their input features. Each student has 27 binary features (e.g., student-specific features, school-specific features), plus 1 basis feature, and the corresponding response is the examination score.

(2) **Parkinson dataset**: the Parkinson dataset consists of Parkinson disease symptom score of 5875 observations for 42 patients. The goal is to predict the symptom score for each patient according to their 16 biomedical features [22, 25]. Hence, the symptom score prediction for a patient is considered as one regression task and we have 42 tasks in total. The output of this dataset is a score consisting of Motor and Total, we establish two regression datasets in our experiment: **Parkinson-Motor** and **Parkinson-Total**.

(3) **Alzheimer dataset**: the Alzheimer disease progression prediction is a very popular multi-task learning problem [30, 31, 33]. The aim is to develop multiple regression models between the magnetic resonance imaging (MRI) features and the cognitive scores (e.g., ADAS-Cog) at consecutive time points (6-month or 1-year interval) [28, 29, 32]. Therefore, the ADAS-Cog score prediction upon MRI features at specific time point is considered as one regression task. In this study, we have 341 MRI features at multiple time points from baseline (M00) to 120-th month (M120), which stands for 12 tasks. To the first time, we consider the Alzheimer disease progression prediction as a lifelong regression problem and use it to evaluate our algorithm.

The statistics details of three datasets are summarised in Table. 1. For each dataset, we split 50% as training set and the rest 50% for testing. Before experiment, all the dataset has been normalized.

4.2 Experimental setup

Our proposed method has two forms, namely weak prediction (WP) and weak prediction features (WPF) enhanced-lifelong regression model. The former uses minimal size of raw prediction results as feature vectors while the latter uses statistical features of whole prediction results as in Eq. (12). We compare our method with two single-task predictors, including single-task learning with LS (STL-LS) base model and PLS (STL-PLS) base model [16]. Additionally, the classic ELLA [20] and ELLA with PLS (ELLA-PLS) base model are utilized as lifelong models for comparison. Note that the classic ELLA only employs LS model as its base predictor, and we simply replace the LS with PLS model to observe how performance changes. More importantly, introducing ELLA-PLS as a comparative method can demonstrate the superiority of our proposed learning framework over traditional lifelong learning framework regardless of replacing the base model. It should be noted that the TaDell [18] 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 lifelong models, each task is presented sequentially to the agent, following the online learning setting. For a fair comparison, the task order is fixed during all experiments.

The mean squared error (MSE) is used to evaluate the overall test performance on the whole tasks, while the averaged MSE per task (MSEpT) is used to evaluate the average performance for each task. 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 lifelong model. The performance of each method are presented by its mean and standard deviation (STD) of the test MSE and ACTpT over 10 independent realizations.

For all lifelong models, we chose dictionary size k and regularization parameters independently for each dataset using a grid search over ranges $\{10^{-n},\ n = 0, \ldots, 8\}$ for regularization parameters and $\{1, \ldots, 8\}$ for k, respectively. For proposed method, the parameter ρ is used to balance the model's fit to the weak prediction's fit, and we empirically set ρ to either 1 or 0.1 and find it works well for all datasets. Another parameter for all PLS-based methods is the latent variable number, we simply set it to 2 that is suitable for all datasets.

4.3 Results and analysis

Comparison on prediction accuracy: The mean and STD of test MSE for various models on three datasets are reported in Table. 2. It can be seen that our proposed method achieves best prediction performance among all models, as evidenced by its smallest MSE and MSEpT for three datasets. Not surprisingly, the STL-LS is the worst

Table 1: Statistics details of the School, Parkinson and Alzheimer datasets.

Dataset	Total tasks	Total samples	Samples for each task	Dimension
School	139	15362	25 to 251	28
Parkinson-Motor	42	5875	101 to 168	16
Parkinson-Total	42	5875	101 to 168	16
Alzheimer	12	6339	69 to 1074	314

Table 2: Test performance comparison of STL-LS, STL-PLS, ELLA, ELLA-PLS and proposed methods for School, Parkinson and Alzheimer dataset in terms of test MSE. Methods with the best and runner-up performances are colored with red and blue, respectively.

Dataset	Metric	STL-LS	STL-PLS	ELLA	ELLA-PLS	Proposed	
						WP	WPF
School	MSE	134.44	117.01	**112.40±0.48**	115.49±1.07	111.46±0.05	**109.40±0.08**
	MSEpT	134.81	118.89	**111.20±0.53**	114.34±1.14	109.63±0.05	**108.24±0.10**
Parkinson-Motor	MSE	8.96	6.86	6.31±0.01	**6.26±0.02**	5.75±0.15	**5.66±0.05**
	MSEpT	9.42	7.19	6.51±0.01	**6.46±0.02**	6.01±0.16	**5.90±0.06**
Parkinson-Total	MSE	11.72	9.37	9.46±0.02	**9.23±0.13**	**7.83±0.01**	7.99±0.08
	MSEpT	12.16	9.68	9.63±0.02	**9.39±0.14**	**8.04±0.01**	8.22±0.08
Alzheimer	MSE	336.63	88.72	103.06±0.22	**88.51±0.99**	**84.39±0.10**	85.83±0.30
	MSEpT	360.10	114.55	128.54±0.54	**113.59±2.37**	**103.85±0.09**	105.87±1.18

model, as it neither modeling relationships among multiple tasks nor has a powerful predictive capacity of its base predictor. The STL-LS model's shortcoming is magnified in the Alzheimer dataset, as each task is with high dimensions and strong co-linearities, and a simple LS model is difficult to handle such complex dataset. Compared to STL-LS, the STL-PLS achieves much better performance. This is because the PLS enables analyzing original features in a reduced-dimensional latent subspace and well addressing the data co-linearity problem. Its advantage is prominent, especially when the feature dimension is very high, as demonstrated in the Alzheimer dataset. Undoubtedly, the ELLA attains much smaller MSE than the STL-LS, as it can learn relationships among multiple tasks. However, its performance improvement for Alzheimer dataset is less prominent than the STL-PLS. This may indicate that the importance of establishing a highly accurate base learner for each task is not worse than learning the relationships among multiple tasks. Hence, the ELLA-PLS is adopted here to combine the advantages of both PLS model and lifelong learning framework. As can be seen that despite the school data, the ELLA-PLS achieves better MSE than both STL-PLS and traditional ELLA. Beyond the ELLA-PLS, our proposed method can further improve its performance with weak regression enhancement.

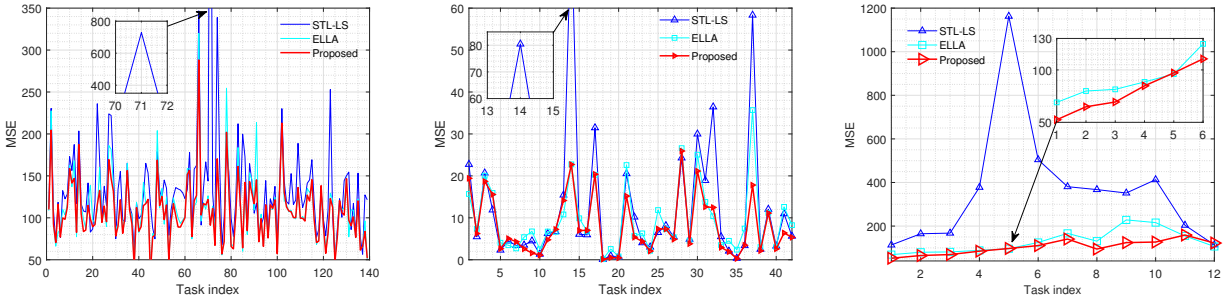
We further compare the test MSE on each task for three datasets (Parkinson Motor and Total have similar results, and we only plot Parkinson-Total here). For a clear presentation, we select three compared models, including baseline STL-LS, classic ELLA and our proposed method with best performance, the result is shown in Fig. 2. Clearly observe that the result is consistent with Table. 2 and our proposed method attains the smallest MSE on the most of tasks.

Analysis of weak regression strategy: We further analysis the difference between two strategies (WP and WPF) for proposed method. Specifically, for school data, the WPF strategy attains smaller MSE than the WP. The reason for this is that each task's sample size for this dataset varies dramatically, and there is a big gap between the minimal (25) and maximal (251) sample size for specific task. Hence, using the prediction features that contain complete task information could be more robust and informative than using a small-size incomplete prediction output (minimal sample size over whole tasks). For Alzheimer dataset, on the contrary, the WP strategy is superior to the WPF strategy. This is because for this dataset, the model space or we say feature dimension is dramatically high. If we use the WPF strategy, the weak regression space is too low (only 6) compared with the model space, and its effect of weak prediction enhancement is likely to be ignored. For Parkinson dataset, the WP strategy and WPF strategy have a comparable performance, as the sample size for each task is close and both strategies provide similar information. It can be seen that the chose of two strategies is problem independent, and we can always chose an appropriate strategy according to the task characteristics and above-mentioned criterion.

Comparison on the number of learned tasks: We further explore how the number of learned tasks influence the overall prediction accuracy of various models. Based on the fixed 50%-50% training-testing set for each task, the learned task number can be reconstructed from 1 to T_{max}. From the performance curves presented in Fig. 3 (only Parkinson-Total is presented here), we can see that among the increase of learned task number, the decline in test MSE of proposed method is more significant compared with other models. This is because through the incremental update of two dictionaries $\boldsymbol{L}$ and $\boldsymbol{K}$ over time, our method becomes more knowledgeable than the classic ELLA learning framework that employs only one dictionary. This clearly demonstrates the effectiveness of cross-task relationship learning in two spaces.

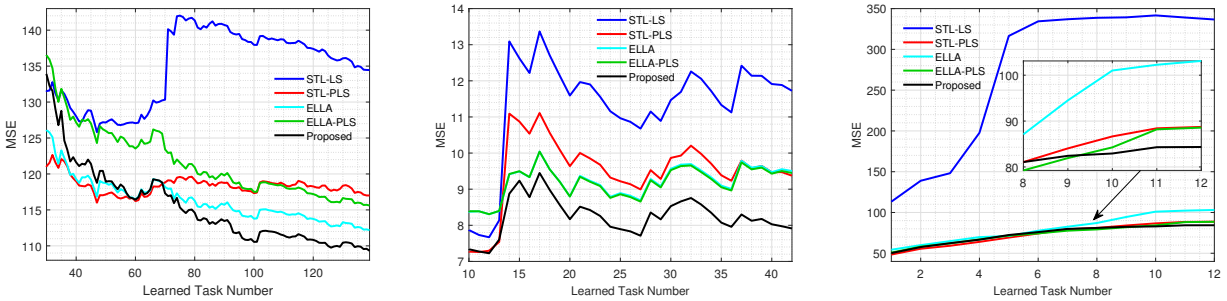
Comparison on the number of training samples: In order to demonstrate the effectiveness of inter-task relationship learning in our method, we further compare the influence of training set proportion on the prediction accuracy for various models over whole tasks. The result is shown in Fig. 4. Clearly observe that our method consistently outperforms other models over the whole training set proportion for both school and Parkinson datasets. For Alzheimer dataset, our method achieves much better performance than the classic ELLA while it has a comparable performance with the PLS-based models (STL-PLS and ELLA-PLS). This is because on the one hand, we incorporate weak regression with a coupled dictionary to alleviate the model's dependence on training data; on the other hand, we employ PLS as the base predictor to effectively handle tasks with high dimensionality and multicollinearity. Hence, our method take advantages of both part and is proven to be effective for different scenarios.

Comparison on the online computational complexity: We compares the ACTpT (ms) of various models on three datasets. The computer for carrying out the experiments has the following configuration: Windows 10, 16 GB of RAM, CPU i7-9750 (2.60 GHz). As can be seen from Table. 3 that all lifelong models attain higher



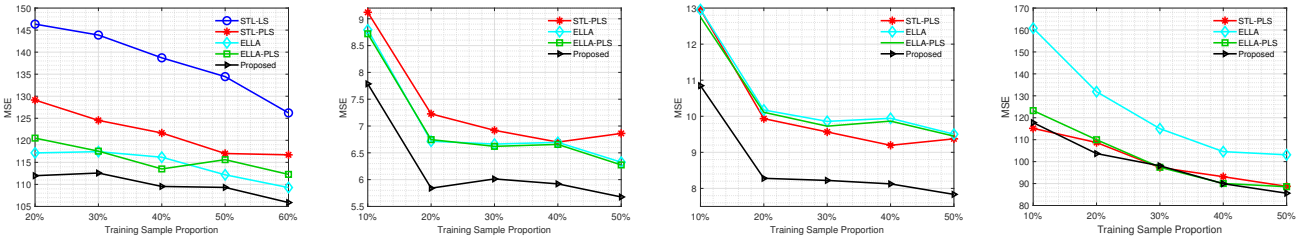
(a) (b) (c)

Figure 2: Comparison of test MSE on each learned task of the STL-LS, ELLA and proposed method for: (a) School, (b) Parkinson, and (b) Alzheimer datasets.



(a) (b) (c)

Figure 3: Comparison of test MSE among the increase of learned task number for: (a) School, (b) Parkinson, and (b) Alzheimer datasets.



(a) (b) (c) (d)

Figure 4: Impact of training sample proportion on the prediction accuracy for: (a) School, (b) Parkinson-Motor, (c) Parkinson-Total, and (b) Alzheimer datasets.

Table 3: Test performance comparison of STL-LS, STL-PLS, ELLA, ELLA-PLS and proposed methods for School, Parkinson and Alzheimer dataset in terms of computation time. Methods with the best and runner-up performances are colored with red and blue, respectively.

Dataset	Metric	STL-LS	STL-PLS	ELLA	ELLA-PLS	Proposed	
						WP	WPF
School	ACTpT (ms)	0.21±0.05	0.29±0.25	**0.76±0.23**	0.84±0.43	1.00±0.43	**1.00±0.41**
Parkinson-Motor	ACTpT (ms)	0.21±0.05	0.58±0.90	1.19±0.49	**1.48±0.90**	6.21±0.49	**3.42±0.56**
Parkinson-Total	ACTpT (ms)	0.21±0.12	0.87±1.38	1.25±0.57	**1.80±1.53**	4.23±1.51	**3.32±1.23**
Alzheimer	ACTpT (ms)	12.46±1.05	2.05±2.11	39.74±2.48	**32.17±2.91**	**32.46±3.32**	46.37±6.72

ACTpT than single task learners, as the former needs to modeling relationships among tasks while the latter not. Additionally, the PLS-based models (STL-PLS, ELLA-PLS and proposed method) have higher running time than the corresponding LS-based models (STL-LS and ELLA), and this phenomenon is opposite for Alzheimer dataset. This is reasonable because although PLS itself has higher computation, it enables modeling data in a lower subspace, and this capacity of PLS makes it particularly efficiency for handling high-dimensional tasks. Although our method attains slightly higher ACTpT than the second-best model, its online computational complexity is still acceptable, as it enables learning each tasks within a fraction of second.

5 CONCLUSION AND FUTURE WORKS

This paper proposes an effective lifelong regression model that integrates weak regression and task model via coupled dictionary learning. Specifically, at each time step we first construct single-task predictor by PLS algorithm that is capable of eliminating data co-linearities and providing higher modeling accuracy. The single-task predictor is used to generate a prior prediction of target value, which is called weak prediction. We further encode these weak prediction results as feature vectors, linking them with task model by two dictionaries that share a joint sparse representation. Since both weak prediction and model provide information about the task, each can augment the learning of the other, thus facilitating both cross-task and inter-task knowledge transfer. The superiority of our method is prominent especially when there is insufficient training data to build an accurate task model, as this weak regression can act as a substitute to fill a vacancy of task model. Finally, extensive experiments have demonstrated the effectiveness of our proposed method.

The proposed weak regression enhanced lifelong learning still requires a small amount of labeled training samples to supervised learning of new task models. In practical lifelong learning setting, tasks can arrive rapidly and often the learner is expected to learn new task model without delay to wait for labeling task. Therefore, unsupervised features, provided solely by task input data, can be incorporated into the lifelong learning framework to achieve learning new task model without output information. This could be promising future improvement for our proposed scheme. Another potential improvement is the extension to nonlinear modeling by replacing the PLS-based task model with some nonlinear base models, such as neural networks.

REFERENCES

[1] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. 2014. Online multi-task learning for policy gradient methods. In *International conference on machine learning*. PMLR, 1206–1214.

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

[3] Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Peng Chu, Quanzeng You, Jiang Wang, Zicheng Liu, and Zheng-jun Zha. 2022. Lifelong unsupervised domain adaptive person re-identification with coordinated anti-forgetting and adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14288–14297.

[4] Muhammad Irfan, Zheng Jiangbin, Muhammad Iqbal, and Muhammad Hassan Arif. 2021. A novel lifelong learning model based on cross domain knowledge extraction and transfer to classify underwater images. *Information Sciences* 552 (2021), 80–101.

[5] David Isele, Mohammad Rostami, and Eric Eaton. 2016. Using Task Features for Zero-Shot Knowledge Transfer in Lifelong Learning.. In *Ijcai*, Vol. 16. 1620–1626.

[6] Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. 2022. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5441–5454.

[7] Tong Liu, Sheng Chen, Shan Liang, Dajun Du, and Chris J Harris. 2020. Fast tunable gradient RBF networks for online modeling of nonlinear and nonstationary dynamic processes. *Journal of Process Control* 93 (2020), 53–65.

[8] Tong Liu, Sheng Chen, Shan Liang, Shaojun Gan, and Chris J Harris. 2020. Multi-output selective ensemble identification of nonlinear and nonstationary industrial processes. *IEEE Transactions on Neural Networks and Learning Systems* 33, 5 (2020), 1867–1880.

[9] Tong Liu, Sheng Chen, Po Yang, Yunpeng Zhu, and Chris J Harris. 2023. Efficient adaptive deep gradient RBF network for multi-output nonlinear and nonstationary industrial processes. *Journal of Process Control* 126 (2023), 1–11.

[10] Tong Liu, Zeyue Tian, Sheng Chen, Kai Wang, and Chris J Harris. 2022. Deep Cascade Gradient RBF Networks With Output-Relevant Feature Extraction and Adaptation for Nonlinear and Nonstationary Processes. *IEEE transactions on cybernetics* (2022).

[11] Jorge Mendez, Boyu Wang, and Eric Eaton. 2020. Lifelong policy gradient learning of factored policies for faster training without forgetting. *Advances in Neural Information Processing Systems* 33 (2020), 14398–14409.

[12] Y. Po. 2016. Lifelogging data validation model for internet of things enabled personalized healthcare. *IEEE Trans. on Systems, Man, and Cybernetics: Systems* 48, 1 (2016), 50–64.

[13] Jun Qi, Po Yang, Geyong Min, Oliver Amft, Feng Dong, and Lida Xu. 2017. Advanced internet of things for personalised healthcare systems: A survey. *Pervasive and mobile computing* 41 (2017), 132–149.

[14] Jun Qi, Po Yang, Lee Newcombe, Xiyang Peng, Yun Yang, and Zhong Zhao. 2020. An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure. *Information Fusion* 55 (2020), 269–280.

[15] S Joe Qin. 1998. Recursive PLS algorithms for adaptive data modeling. *Computers & Chemical Engineering* 22, 4-5 (1998), 503–514.

[16] S Joe Qin, Yining Dong, Qinqin Zhu, Jin Wang, and Qiang Liu. 2020. Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring. *Annual Reviews in Control* 50 (2020), 29–48.

[17] Mohammad Rostami. 2021. Lifelong domain adaptation via consolidated internal distribution. *Advances in neural information processing systems* 34 (2021), 11172–11183.

[18] Mohammad Rostami, David Isele, and Eric Eaton. 2020. Using task descriptions in lifelong machine learning for improved performance and zero-shot transfer. *Journal of Artificial Intelligence Research* 67 (2020), 673–704.

[19] Paul Ruvolo and Eric Eaton. 2013. Active task selection for lifelong machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 27. 862–868.

[20] Paul Ruvolo and Eric Eaton. 2013. ELLA: An efficient lifelong learning algorithm. In *International conference on machine learning*. PMLR, 507–515.

[21] Paul Ruvolo and Eric Eaton. 2014. Online multi-task learning via sparse dictionary optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[22] G Sun. 2022. Representative task self-selection for flexible clustered lifelong learning. *IEEE Trans. Neural Networks and Learning Systems* 33, 4 (2022), 1467–1481.

[23] Gan Sun, Yang Cong, Jiahua Dong, Yuyang Liu, Zhengming Ding, and Haibin Yu. 2021. What and how: generalized lifelong spectral clustering via dual memory. *IEEE transactions on pattern analysis and machine intelligence* 44, 7 (2021), 3895–3908.

[24] Gan Sun, Yang Cong, Qianqian Wang, Jun Li, and Yun Fu. 2020. Lifelong spectral clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5867–5874.

[25] Gan Sun, Yang Cong, and Xiaowei Xu. 2018. Active lifelong learning with" watchdog". In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[26] Shuojin Yang and Zhanchuan Cai. 2023. Cross Domain Lifelong Learning Based on Task Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[27] Fei Ye and Adrian G Bors. 2023. Compressing Cross-Domain Representation via Lifelong Knowledge Distillation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[28] Yu Zhang, Tong Liu, Vitaveska Lanfranchi, and Po Yang. 2022. Explainable Tensor Multi-Task Ensemble Learning Based on Brain Structure Variation for Alzheimer's Disease Dynamic Prediction. *IEEE Journal of Translational Engineering in Health and Medicine* 11 (2022), 1–12.

[29] Yu Zhang, Menghui Zhou, Tong Liu, Vitaveska Lanfranchi, and Po Yang. 2022. Spatio-temporal Tensor Multi-Task Learning for Predicting Alzheimer's Disease in a Longitudinal study. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 979–985.

[30] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, and Jieping Ye. 2012. Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1095–1103.

[31] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. 2013. Modeling disease progression via multi-task learning. *NeuroImage* 78 (2013), 233–248.

[32] Menghui Zhou, Yu Zhang, Tong Liu, Yun Yang, and Po Yang. 2022. Multi-task Learning with Adaptive Global Temporal Structure for Predicting Alzheimer's Disease Progression. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2743–2752.

[33] Menghui Zhou, Yu Zhang, Tong Liu, Yun Yang, and Po Yang. 2023. Efficient multi-task learning with adaptive temporal structure for progression prediction. *Neural Computing and Applications* (2023), 1–16.

[34] Zhonglin Zuo, Li Ma, Shan Liang, Jing Liang, Hao Zhang, and Tong Liu. 2022. A semi-supervised leakage detection method driven by multivariate time series for natural gas gathering pipeline. *Process Safety and Environmental Protection* 164 (2022), 468–478.