Lifelong Learning Meets Industrial Processes: An Enabling Adaptive Process Modeling Framework with Delayed Process Output Measurement

Professor Sheng Chen

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ, United Kingdom

Joint work with Dr Tong Liu and Dr Po Yang, University of Sheffield, U.K.

Motivation

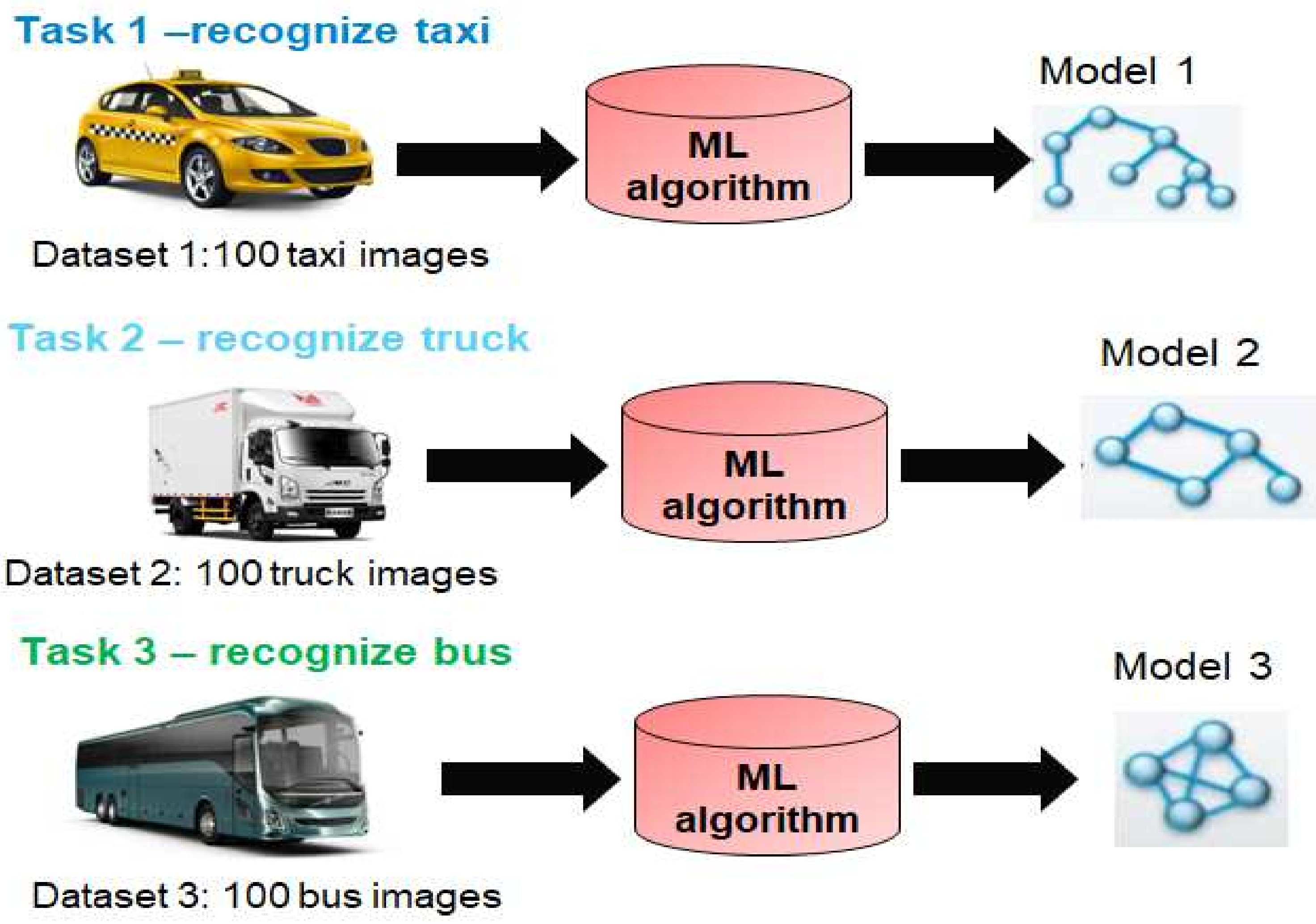
- Many industrial processes operate **continuously** **batch by batch**
  - **Predicting** plant output is needed for monitoring, decision making and control
  - Predictor model is constructed from historical plant operational data

- During plant operation, underlying process characteristics **change**
  - Predictor model must **adapt**, which requires actual process output as desired target

- **Process output measurement** for these batch-by-batch streaming processes is typically seriously **delayed**
  - Without **timely** process output **measurement**, adapting predictor model is **impossible**

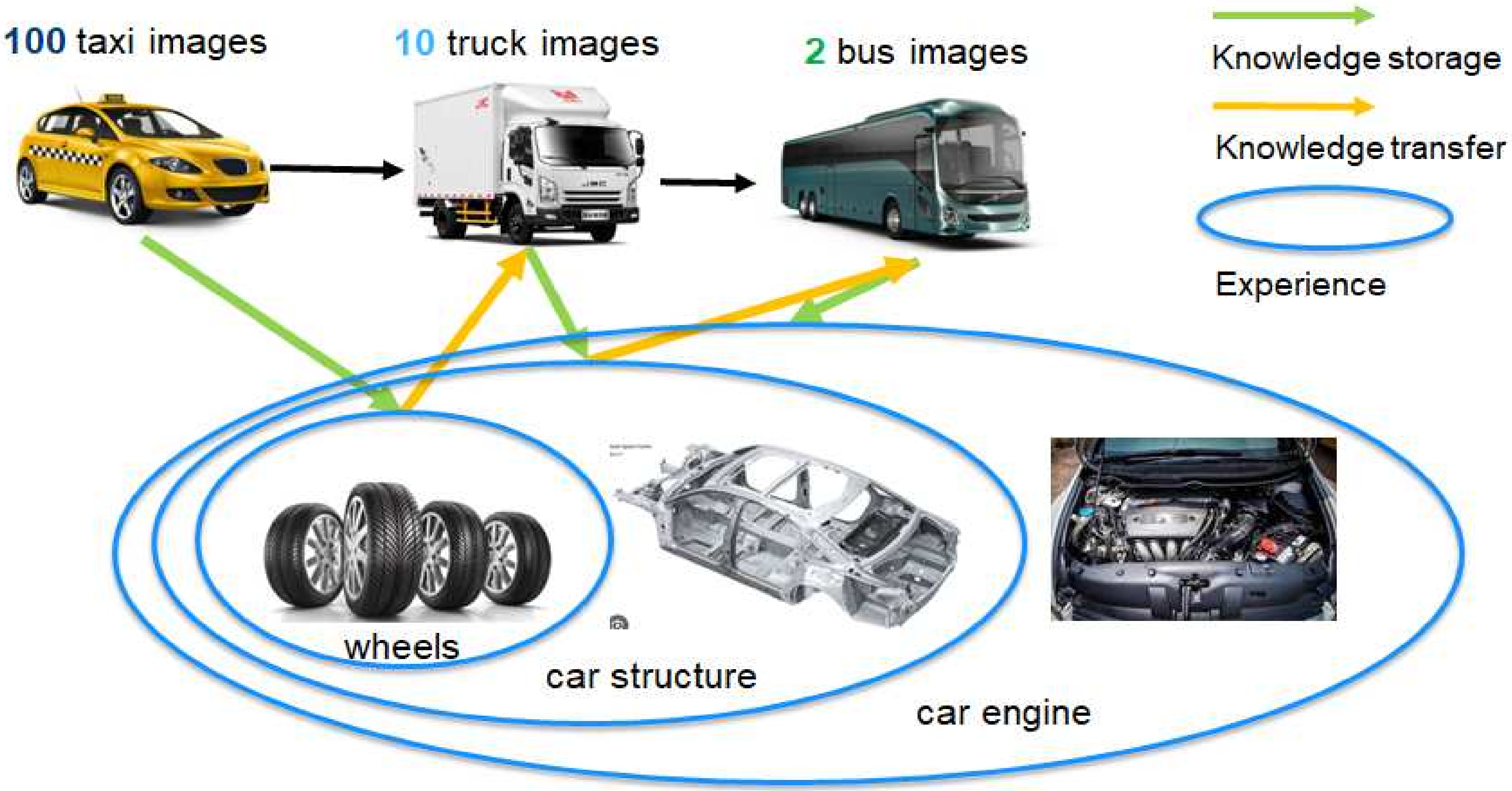
- 'Old' predictor is used **without** adaptation $\Rightarrow$ **degrade** prediction accuracy
  - **How to tackle this problem?**

Classic Machine Learning



- **Isolated learning**: Do not retain knowledge learned in the past and use it in future, requring large number of training data to learn effectively

How Human Learns



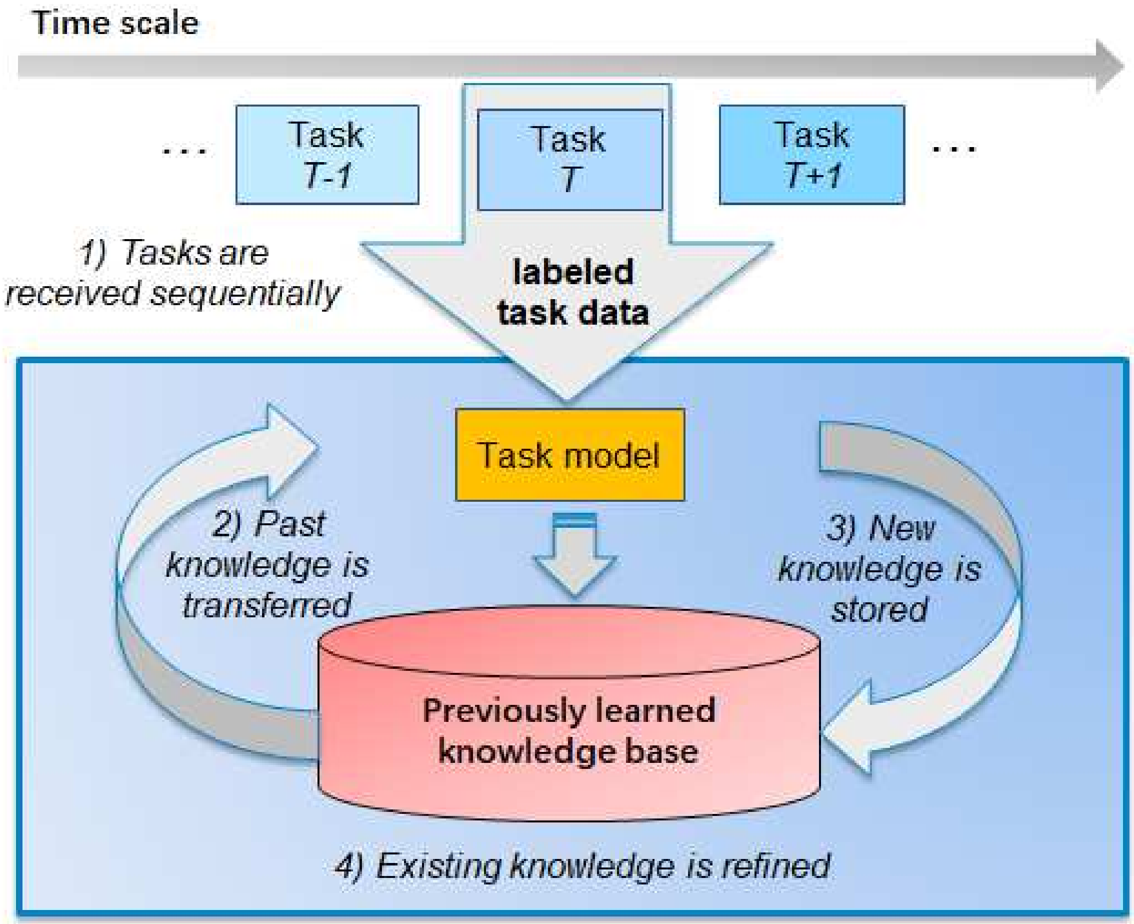
- **Human learning**: Learn continually with experience
  - Maintain **knowledge base** (brain), use past knowledge (**knowledge transfer**) to aid new task, and **store new knowledge** learned for future
  - Quickly learn new tasks with small dataset

Lifelong Maching Learning

- **Lifelong** maching **learning** **imitates** **human learning**
- Efficient lifelong learning algorithm

ELLA:

- Learning tasks consecutively
  - Maintain **knowledge base** for past learned tasks
  - **Transfer knowledge** from previous tasks to learn new task
  - **Store new knowledge learned** in new task to knowledge base
- To build a **task model** required labeled training data (**both inputs and desired outputs**)



ELLA

Efficient Lifelong Learning Algorithm

- ELLA fits a **parametric** model for each task

$$f^{(t)}\left(\boldsymbol{x};\boldsymbol{\theta}^{(t)}\right) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\theta}^{(t)} \quad \boldsymbol{\theta}^{(t)} \in \mathbb{R}^d$$

Base model: **linear**, **ELLA**: actually **nonlinear**

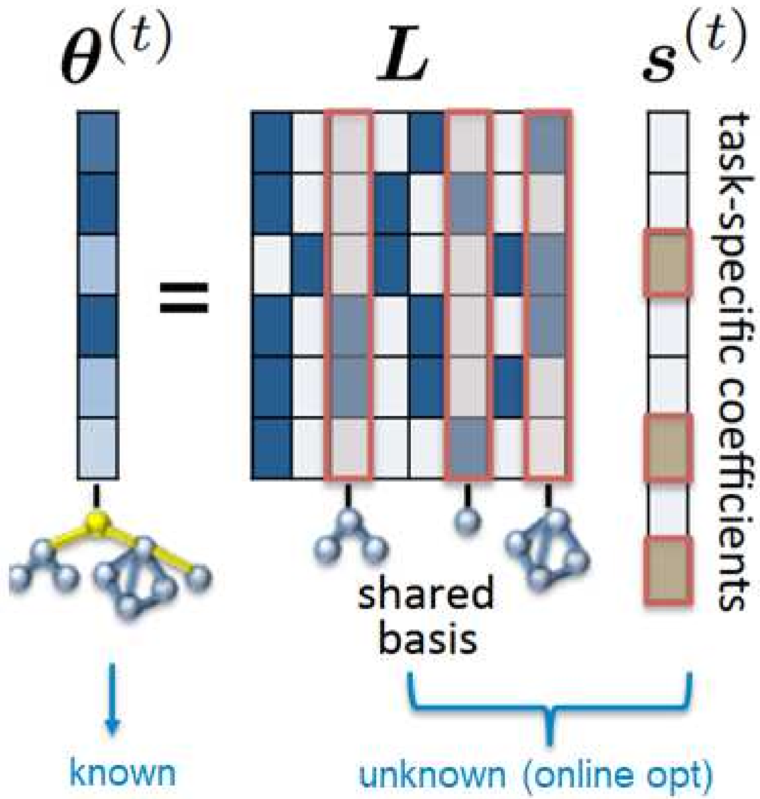
- $\boldsymbol{\theta}^{(t)}$: linear combinations of **knowledge base** $\boldsymbol{L}$ via **sparse encoding** $\boldsymbol{s}^{(t)} \in \mathbb{R}^k$

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\,\boldsymbol{s}^{(t)} \quad \boldsymbol{L} \in \mathbb{R}^{d\times k}$$

- Objective function:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\underbrace{J\left(\boldsymbol{\theta}^{(t)}\right)}_{\text{model fits to each task}} + \mu \underbrace{\|\boldsymbol{s}^{(t)}\|_1}_{\text{sparsity of coding coefficient}}\right) + \lambda \underbrace{\|\boldsymbol{L}\|_F^2}_{\text{regularize base complexity}}$$

$\boldsymbol{S} = \left[\boldsymbol{s}^{(1)}\cdots\boldsymbol{s}^{(T)}\right]$, T: tasks seen so far



- Online optimization: tasks arrive consecutively, update 'recursively' task by task

ELLA (continue)

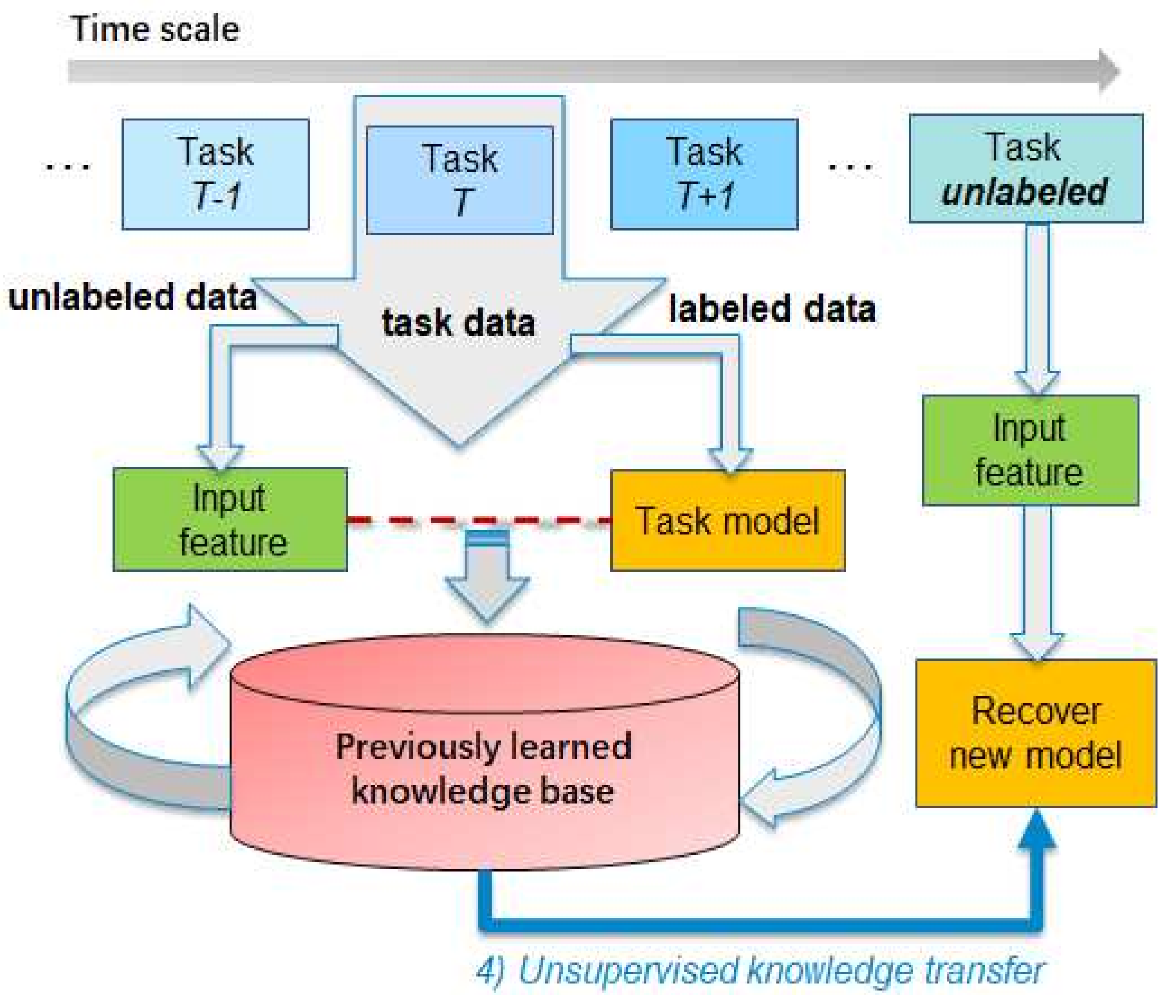
ELLA: given new task (batch) t

1. Train single-task model $\widehat{\boldsymbol{\theta}}^{(t)}$ for task t
   - Estimate $\widehat{\boldsymbol{\theta}}^{(t)}$ requires **labeled training data** (both input and desired output)
2. With $\widehat{\boldsymbol{\theta}}^{(t)}$, **solve sparse coding** coefficient $\boldsymbol{s}^{(t)}$ in current knowledge base or dictionary $\boldsymbol{L}$ via LASSO
   - **Knowledge transfer** from past tasks
3. **Update dictionary** $\boldsymbol{L}$ with $\widehat{\boldsymbol{\theta}}^{(t)}$, $\boldsymbol{s}^{(t)}$ and old $\boldsymbol{L}$ via efficient ELLA unpdate rules
   - **Store knowledge** learned for task t
4. **Current task model** is given by $\boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$

- **Build task model $\boldsymbol{\theta}^{(t)}$ requires labeled training data**
- **Not for streaming batch-by-batch industrial processes with delayed output measurement**

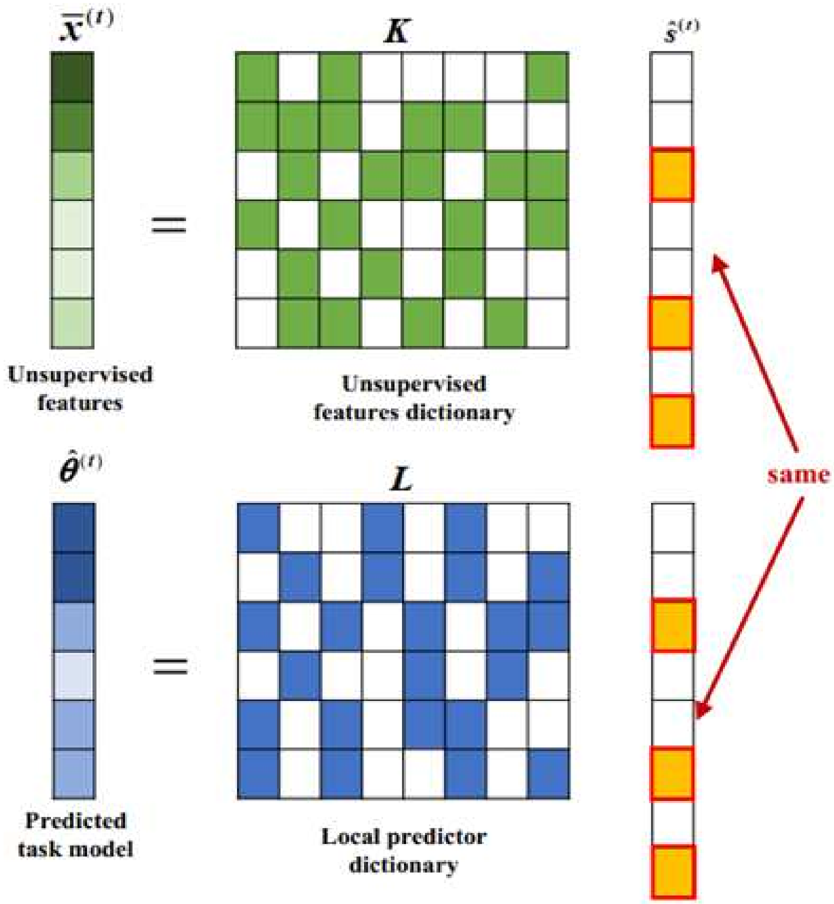
Unsupervised Transfer Aided Lifelong Learning

- We develop UTaLL: **unsupervised transfer** aided **lifelong learning**
- Improve performance by knowledge sharing in two spaces:
  - **Task model space** (as in ELLA)
  - New **input feature space**
- Learn new tasks without labeled data by unsupervised transfer



Liu, Wang, Yang, Chen, Harris, "Unsupervised transfer aided lifelong regression for learning without target output," *IEEE Trans. Knowledge and Data Engineering* **(under second review)**

UTaLL Idea



- **Coupled dictionaries** relate **task model** parameters and **input features**

$$\text{task model } \boldsymbol{\theta}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$$

$$\text{input feature } \overline{\boldsymbol{x}}^{(t)} = \boldsymbol{K}\boldsymbol{s}^{(t)}$$

  - $\overline{\boldsymbol{x}}^{(t)}$: averaging inputs of batch t
  - **Input feature** also reflects **underlying plant characteristics**
  - $\boldsymbol{K}$: **knowledge base** for input feature space
- ELLA objective function

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

- UTaLL objective function

$$\min_{\boldsymbol{L},\boldsymbol{K},\boldsymbol{S}} \frac{1}{T}\sum_{t=1}^{T}\left(\underbrace{J\left(\boldsymbol{\theta}^{(t)}\right)}_{\text{model fit}} + \rho\underbrace{\|\overline{\boldsymbol{x}}^{(t)} - \boldsymbol{K}\boldsymbol{s}^{(t)}\|_2^2}_{\text{input feature fit}} + \mu\underbrace{\|\boldsymbol{s}^{(t)}\|_1}_{\text{sparsity}}\right) + \lambda\underbrace{\left(\|\boldsymbol{L}\|_F^2 + \|\boldsymbol{K}\|_F^2\right)}_{\text{complexity}}$$

- For **labeled task**, learning steps are same as ELLA

Unsupervised Knowledge Transfer

Given new task t with **input data only**

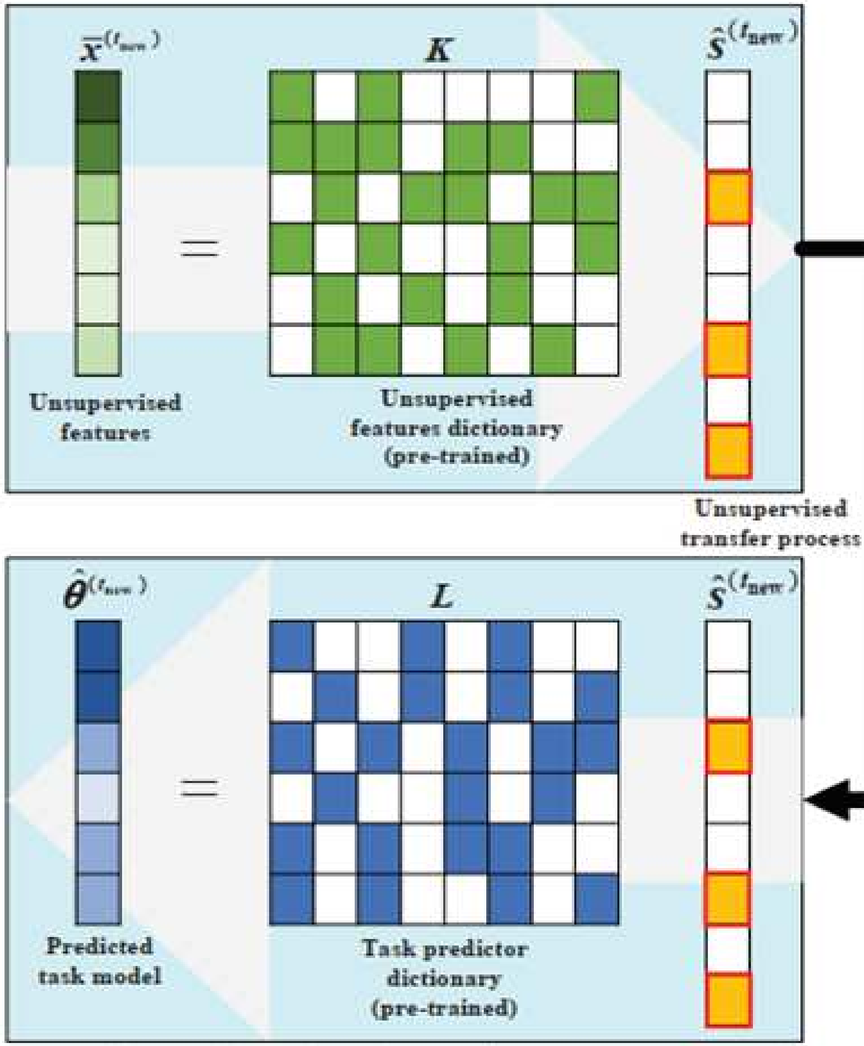
1. Use input feature $\overline{\boldsymbol{x}}$ and dictionary $\boldsymbol{K}$ to **recover sparse code** $\boldsymbol{s}$

$$\boldsymbol{s}^{(t)} = \arg\min_{\boldsymbol{s}} \left\| \overline{\boldsymbol{x}}^{(t)} - \boldsymbol{K}\boldsymbol{s} \right\|_2^2 + \mu \|\boldsymbol{s}\|_1$$

2. Use recovered $\boldsymbol{s}^{(t)}$ and dictionary $\boldsymbol{L}$ to **recover model** parameters $\boldsymbol{\theta}$

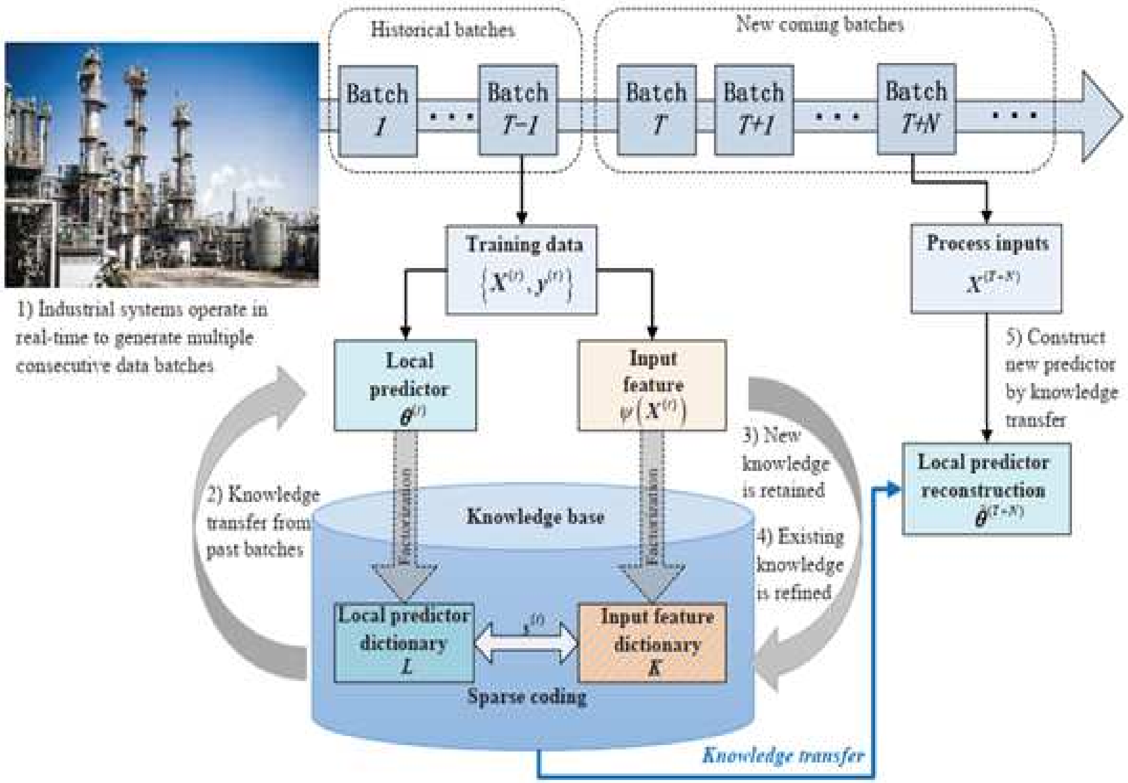
$$\widetilde{\boldsymbol{\theta}}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)}$$

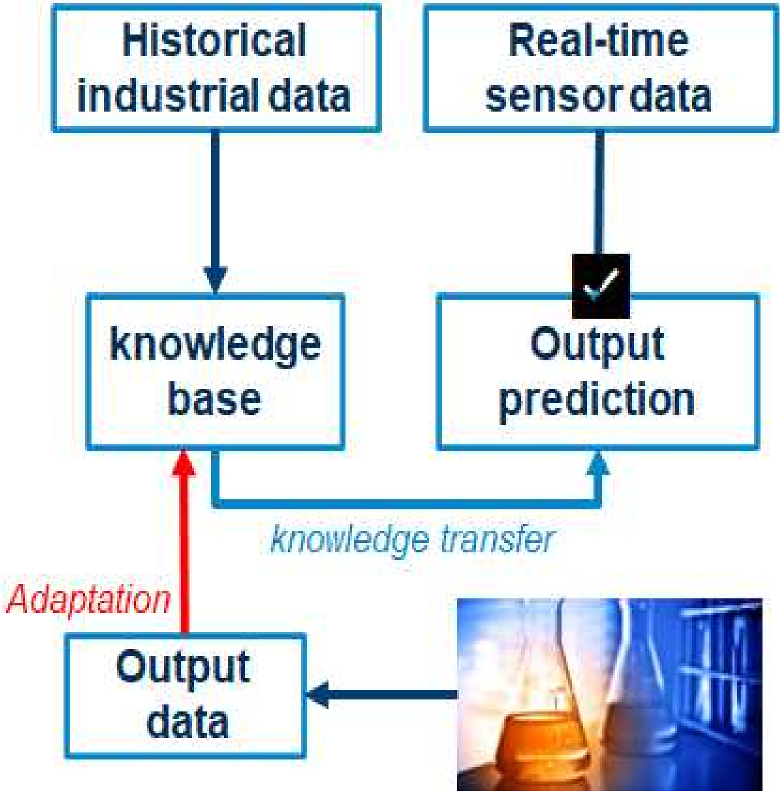
yielding **task model** for new task t



This UTaLL can be apply to online prediction and adaptation of streaming batch-by-batch industrial processes with delayed output measurement

UTaLL Meets Industrial Processes





- Use historical labeled data to build task model dictionary $\boldsymbol{L}$ and input feature dictionary $\boldsymbol{K}$
- During online operation of plant, when a new batch of input data only arrives
  - Use unsupervised knowledge transfer to recover predictor for predicting process output
  - Later if process output measurements arrive, update $\boldsymbol{L}$ and $\boldsymbol{K}$

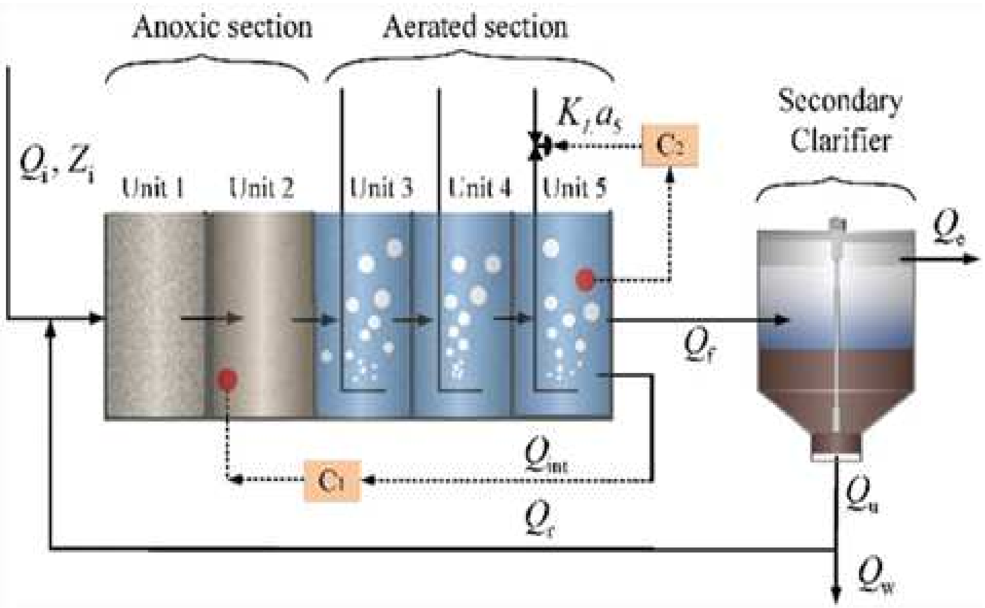
Liu, Chen, Yang, Zhu, Mercangoz, Harris, "Lifelong learning meets dynamic processes: An emerging streaming process prediction framework with delayed quality measurement," *IEEE Trans. Control Systems Technology* **(early access)**

Comparison Schemes

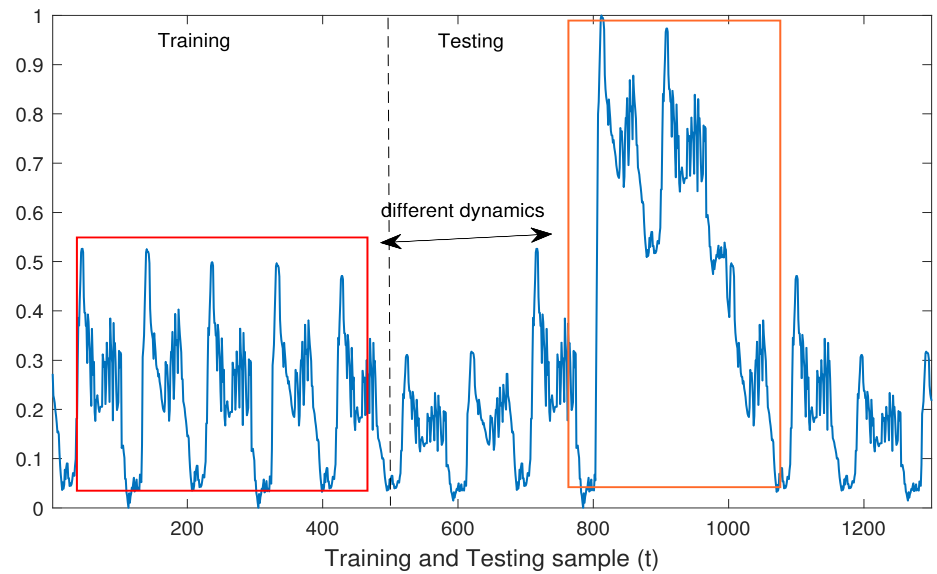
- **LS**, **BAL** (Bayesian augmented Lagrangian), **PLS**: **Nonadaptive**
  - Trained models are fixed during online operation
- **CLR** (clustering-based locally linear regression), **CLR-ensemble**: **Nonadaptive**
  - Trained local model set fixed during online operation
- **RLS-batch**: Model from previous batch is used to predict new batch. After true process output data for this new batch are acquired, it updates model over the batch
- Proposed UTaLL
  - **Pro-nonadaptive**: During online operation, trained KB is fixed, and model is recovered by unsupervised transfer to predict new batch data
  - **Pro-adaptive**: after unsupervised transfer based prediction of new batch data, KB is adapted
- **RLS-idealized**: after predicting output for a given input sample, output measurement is available to adapt model with input-output sample pair
  - **Cannot be used in online modeling and prediction for streaming processes with delayed output measurement**
- **Pro-idealized**: UTaLL continuously operates in training mode
  - **Cannot be used in online modeling and prediction for streaming processes with delayed output measurement**

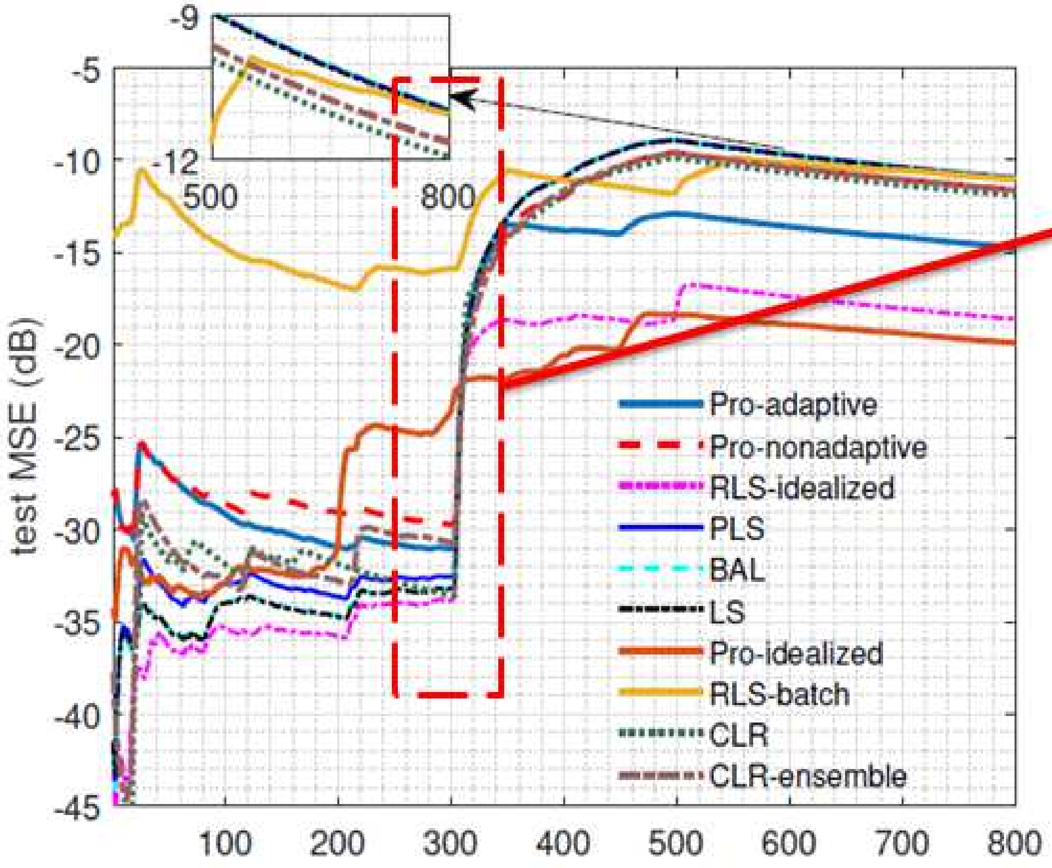
Wastewater Treatment Plant

Input/output	Description
x_1	Readily biodegradable substrate
x_2	Particulate inert organic matter
x_3	Slowly biodegradable substrate
x_4	Active heterotrophic biomass
x_5	$NH_4^+ + NH_3$ nitrogen
y	Flow rate

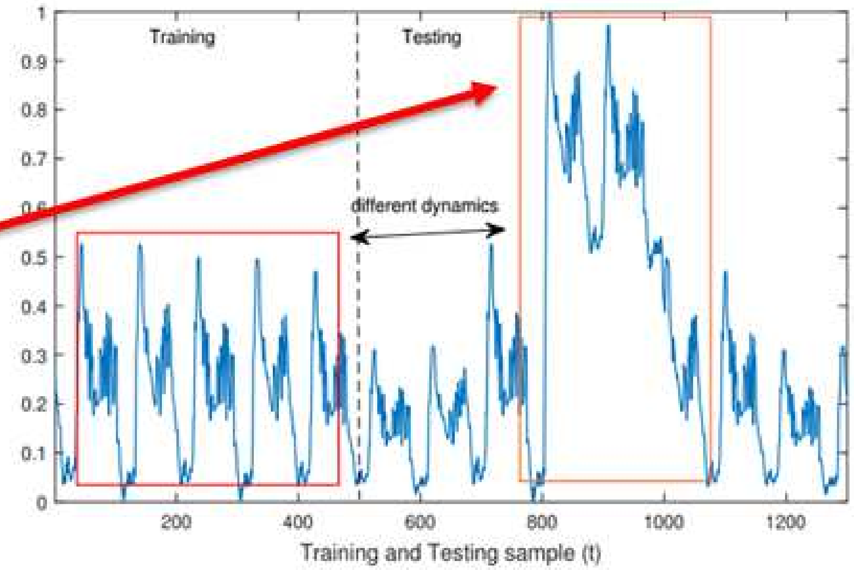


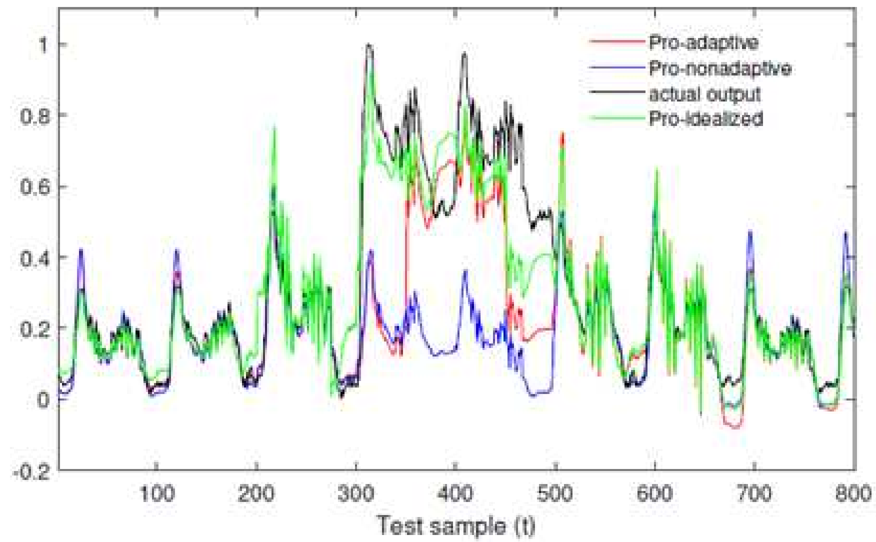
- 1300 samples are collected
- 40% training, 60% online testing
- Delayed output measurement time: 50
- Highly **time-varying** data: combination of **dry** weather and long **rainy** period





Online MSE comparison

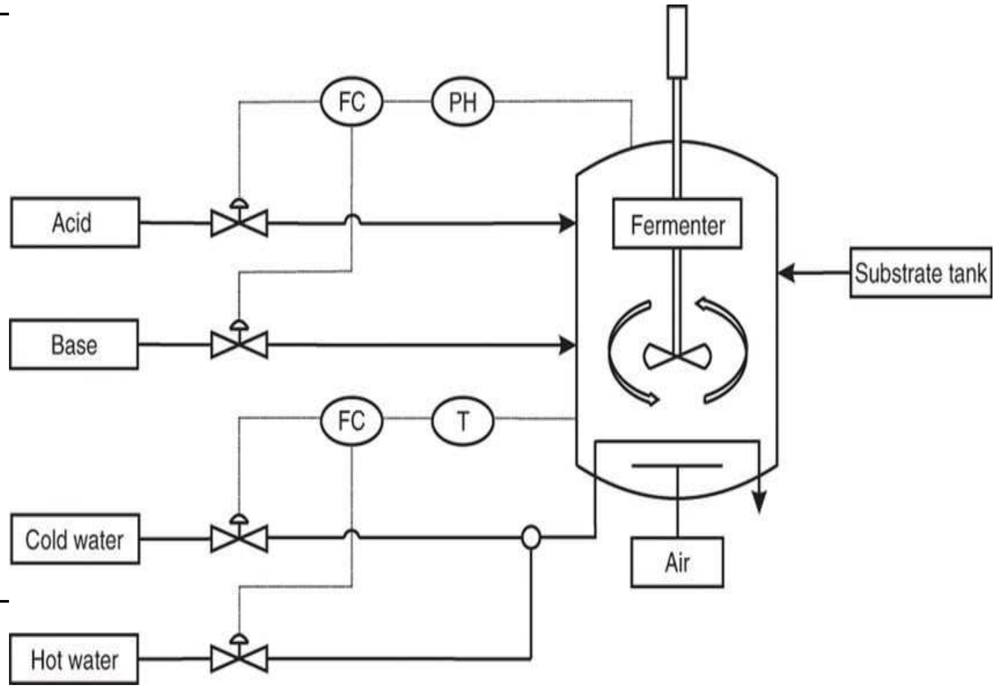




Model prediction by lifelong learner

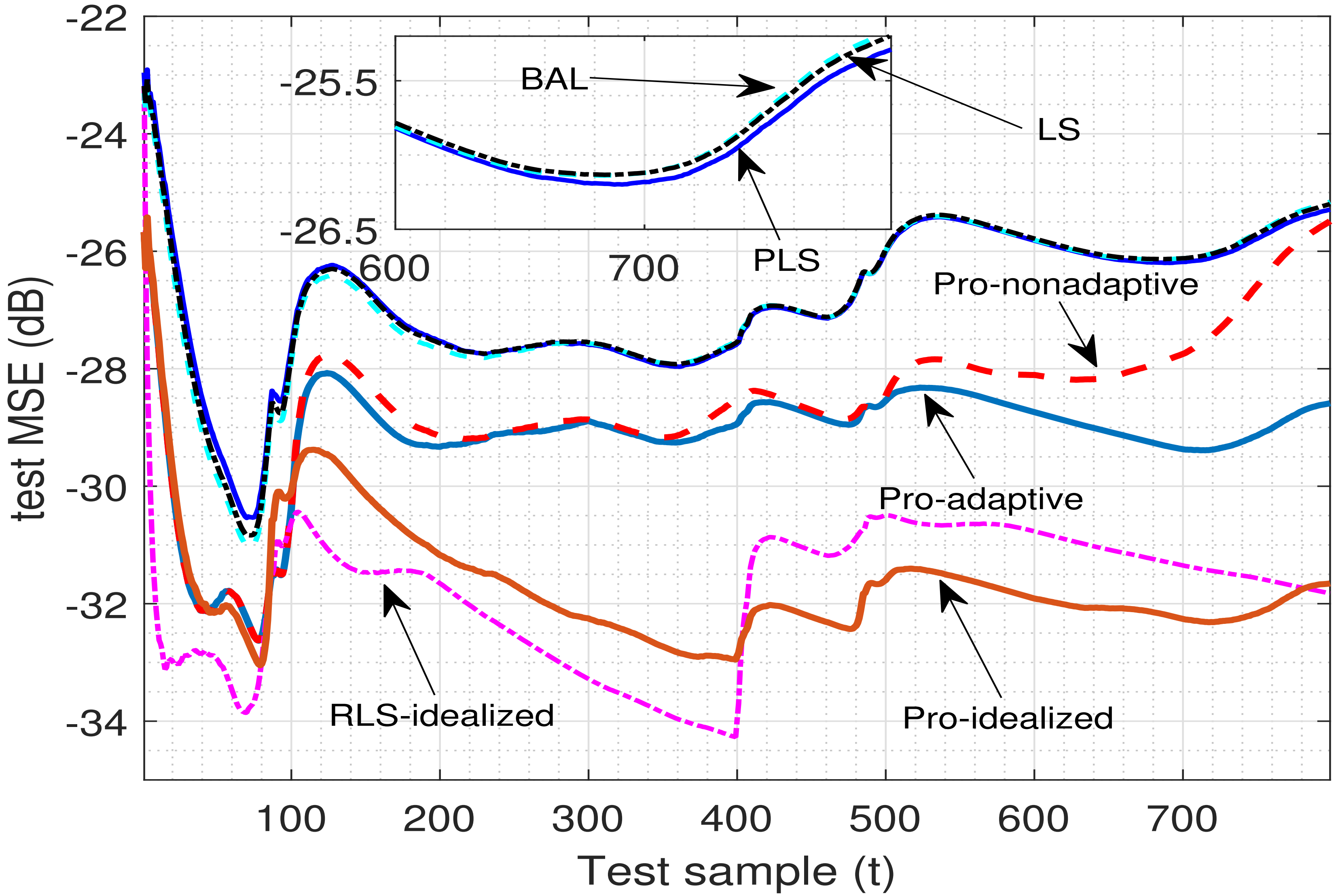
Penicillin Fermentation Process

Input/Output	Description
x_1	Aeration rate
x_2	Agitator power
x_3	Substrate feed rate
x_4	Substrate feed temperature
x_5	Dissolved oxygen concentration
x_6	Culture volume
x_7	Carbon dioxide concentration
x_8	pH
x_9	Fermentor temperature
x_{10}	Generated heat
y_1	Penicillin concentration
y_2	Substrate concentration



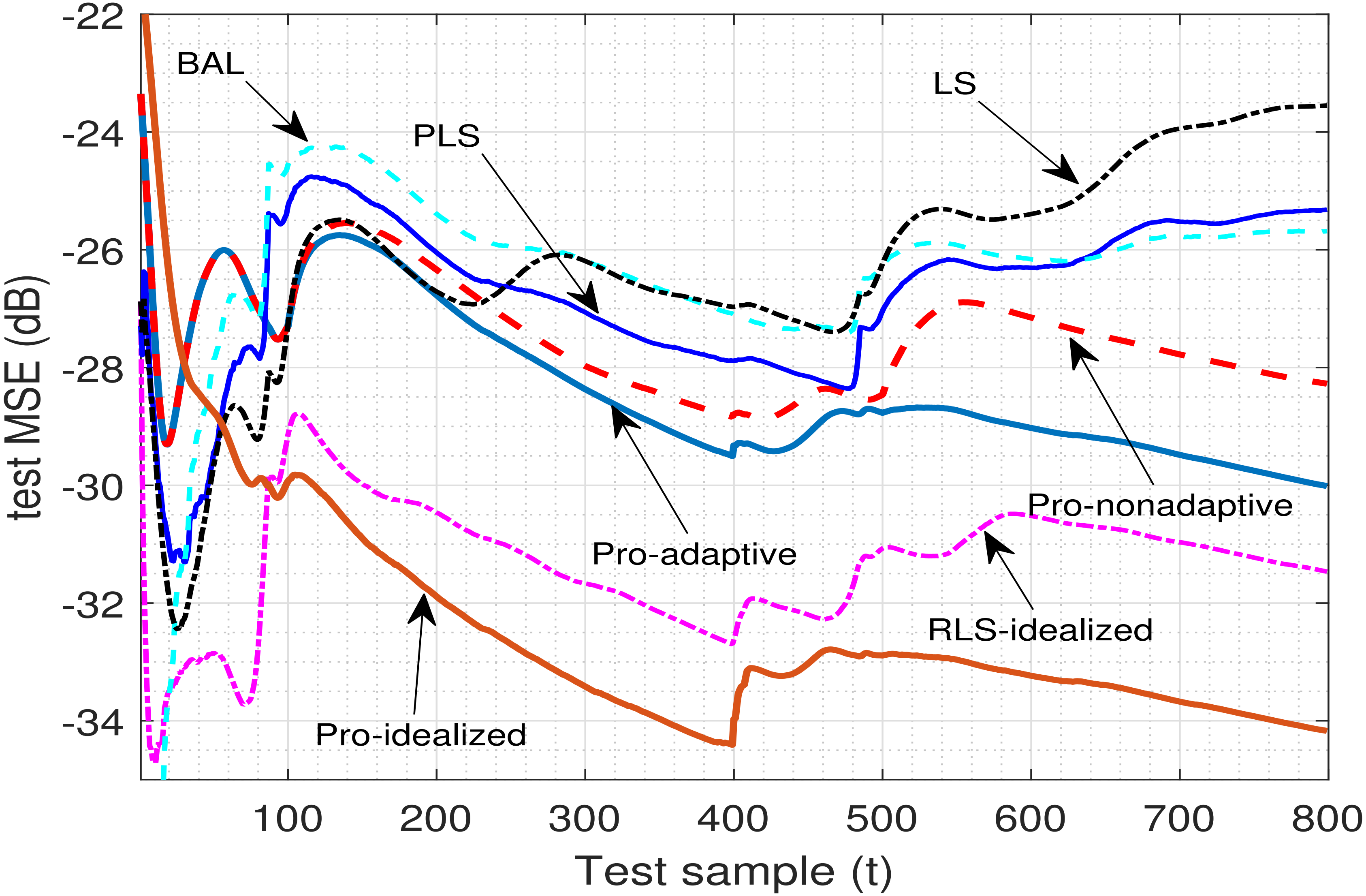
- 1600 samples are collected
- 50% training, 50% online testing
- Delayed output measurement time: 100
- **Multi-mode time-varying nonlinear** characteristics

Predicting Penicillin Concentration



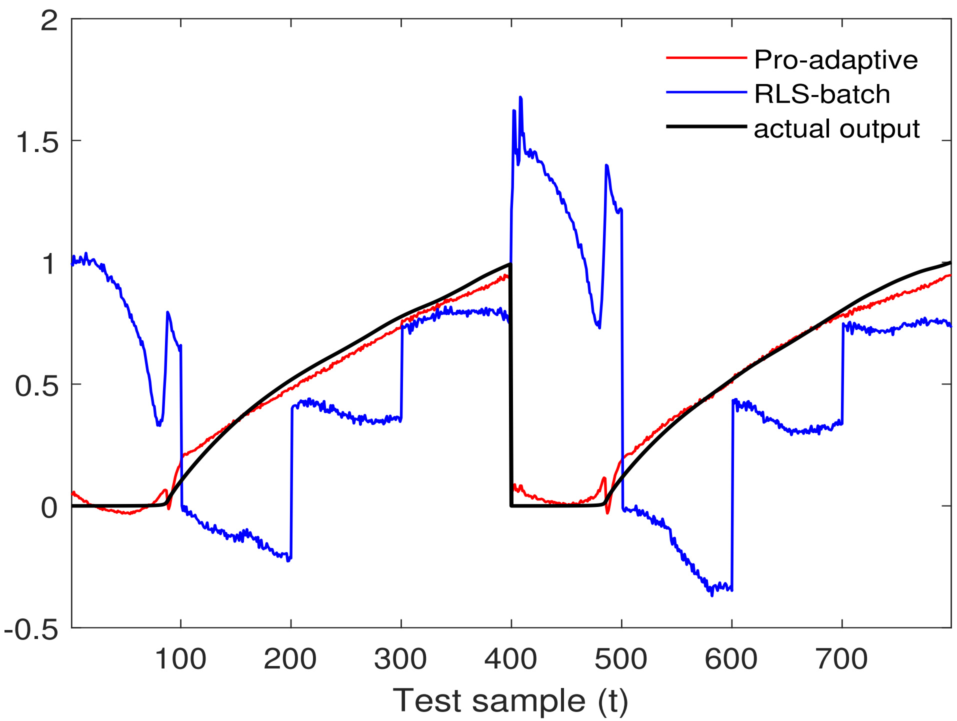
Prediction performance of RLS-batch is extremely poor, off the scale

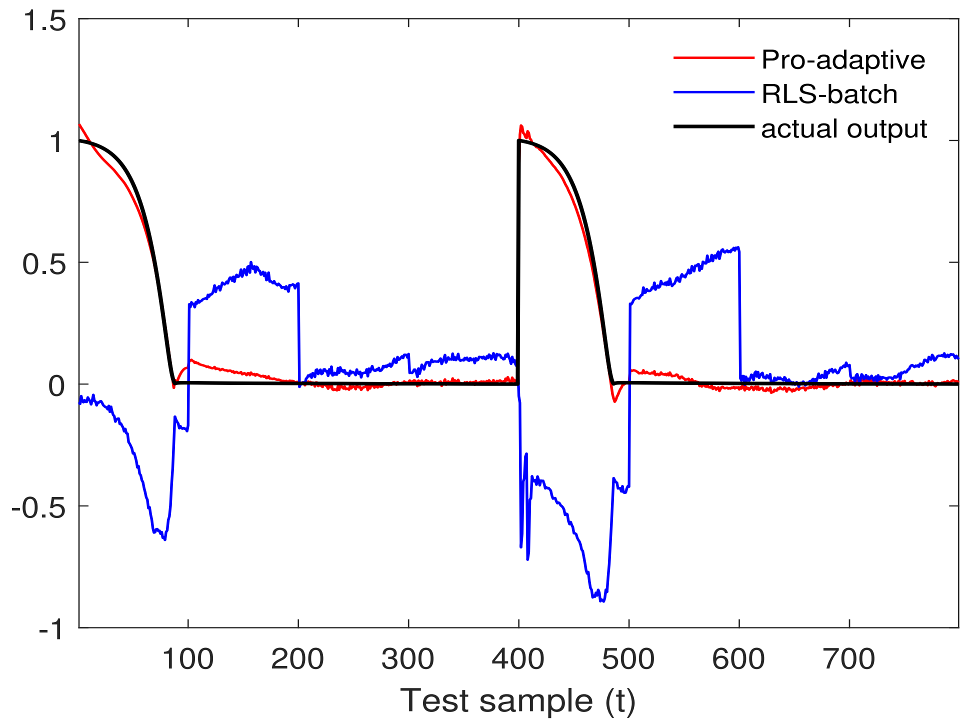
Predicting Substrate Concentration



Prediction performance of RLS-batch is extremely poor, off the scale

Why RLS-batch So Poor





- RLS has short memory, allowing it to forget past data and concentrate on current data
- After batch adaptation, model forgets most of past knowledge and captures characteristics of current batch
- This model is then used to predict next batch
- For highly time-varying nonlinear process, characteristics of next new batch can be very different from those captured in model

Conclusions

- Many industrial processes operate **continuously** **batch by batch**
  - **Predicting** plant output from plant input is essential during online operation
  - Online **adapting model** is vital to track plant time-varying characteristics
  - Process output measurements of these streaming batch-by-batch processes are seriously **delayed**, making it **challenging** for online adapting model

- **Lifelong** machine **learning** imitates **human learning** and has many advantages
  - Require **input-output data** to construct predictor, and cannot be applied to streaming batch-by-batch processes with delayed output measurement

- We have developed novel **unsupervised transfer aided lifelong learning**, capable of construct predictor from **input data only**
  - **Ideal for online prediction and model adaptation of streaming batch-by-batch processes with delayed output measurement**