Evaluating Multi-task Curriculum Learning for Forecasting Energy Consumption in Electric Heavy-duty Vehicles

Yuantao Fan[1,*], Sławomir Nowaczyk[1], Zhenkan Wang[2] and Sepideh Pashami[1,3]

[1]Kristian IV:s väg 3, 301 18 Halmstad, Sweden, Center for Applied Intelligent Systems Research (CAISR), Halmstad University

[2]Gropegårdsgatan 2, 417 15 Göteborg, Sweden, Volvo Group

[3]Isafjordsgatan 28 A, 164 40 Kista, Sweden, Research Institutes of Sweden (RISE)

Abstract

Accurate energy consumption prediction is crucial for optimising the operation of electric commercial heavy-duty vehicles, particularly for efficient route planning, refining charging strategies, and ensuring optimal truck configuration for specific tasks. This study investigates the application of multi-task curriculum learning to enhance machine learning models for forecasting the energy consumption of various onboard systems in electric vehicles. Multi-task learning, unlike traditional training approaches, leverages auxiliary tasks to provide additional training signals, which has been shown to enhance predictive performance in many domains. By further incorporating curriculum learning, where simpler tasks are learned before progressing to more complex ones, neural network training becomes more efficient and effective.

We evaluate the suitability of these methodologies in the context of electric vehicle energy forecasting, examining whether the combination of multi-task learning and curriculum learning enhances algorithm generalisation, even with limited training data. We primarily focus on understanding the efficacy of different curriculum learning strategies, including sequential learning and progressive continual learning, using complex, real-world industrial data.

Our research further explores a set of auxiliary tasks designed to facilitate the learning process by targeting key consumption characteristics projected into future time frames. The findings illustrate the potential of multi-task curriculum learning to advance energy consumption forecasting, significantly contributing to the optimisation of electric heavy-duty vehicle operations. This work offers a novel perspective on integrating advanced machine learning techniques to enhance energy efficiency in the exciting field of electromobility.

Keywords

Energy Consumption Forecasting, Curriculum Learning, Multi-task Learning, Electric Vehicles

1. Introduction

Predicting energy consumption for electric vehicles (EVs), especially those used in commercial heavy-duty contexts, is paramount for improving their operational efficiency and promoting

sustainability. Effective energy consumption forecasts are indispensable for strategic route planning, optimising charging protocols, and ensuring that vehicle configurations align well with specific operational demands. As electric vehicles gain traction as a viable and eco-friendly alternative to internal combustion engine vehicles, the importance of precise energy consumption predictions becomes increasingly pronounced. The challenges in this domain are multifaceted, stemming from the inherent variability in driving conditions, vehicle load, and diverse environmental factors, which collectively complicate the development of accurate predictive models. Overcoming these obstacles is essential not only for enhancing the reliability and performance of EVs but also for minimising operational costs and boosting the overall efficiency of electric transport systems.

The transition to electric vehicles is a significant step towards reducing greenhouse gas emissions and achieving sustainable transportation goals. However, since limited energy storage puts unique constraints on which operations are feasible, the benefits of EVs can only be fully realised through the development of specified forecasting methods that accurately anticipate energy needs. In this context, AI and ML emerge as transformative tools. AI-driven models can analyse vast amounts of data to uncover patterns and relationships that are not immediately apparent, providing more accurate and reliable energy consumption forecasts. These models can adapt to new data, continuously improving their predictions over time.

Nevertheless, energy consumption forecasting for EVs faces critical challenges, such as dynamic driving conditions and fluctuating loads, which makes even state-of-the-art methods struggle to handle complex real-world data effectively. While the potential to learn from historical data and identify trends that influence energy consumption is the biggest strength of ML-based approaches, it is crucial to develop robust models that can generalise well across different scenarios and vehicle types.

The complexity and variability inherent in forecasting energy consumption for electric vehicles make it a relevant testing ground for cutting-edge modelling techniques that promise to handle diverse and dynamic data inputs. In particular, Multi-Task Learning (MTL) presents a compelling solution by enabling simultaneous training across multiple related tasks, thereby leveraging shared information to improve the predictive performance of each task. In contrast, training machine learning models in a traditional setting only utilize the target task. MCL is particularly beneficial in scenarios with limited training data, as it enhances generalisation by incorporating auxiliary tasks that provide additional training signals. Moreover, the efficacy of MTL can be further amplified by integrating curriculum learning (CL), which structures the learning process in a progressive manner. Curriculum learning organises tasks from simple to complex, allowing the model to build a robust foundation before tackling more challenging problems. By combining these methodologies into multi-task curriculum learning (MCL), we can efficiently train neural networks that not only perform better on individual tasks but also generalise more effectively across different contexts. MCL optimises the learning trajectory, ensuring that simpler tasks enhance the model's capability to learn more complex ones, ultimately leading to more accurate and reliable energy consumption forecasts for electric heavy-duty vehicles. This integrative approach has been shown to be a potent strategy to address the multifaceted challenges in several domains but has not been applied to EV auxiliary energy forecasting before. Thus, this paper aims to evaluate the suitability of MCL in this real-world, complex scenario.

Generating a set of auxiliary tasks is a critical step in the implementation of MCL – and how to do it for forecasting energy consumption in EVs requires experimental evaluation. To create auxiliary tasks, one must first obtain an understanding of the primary task, identifying key factors and variables that influence energy consumption and the types of patterns that are indicative of future behaviour. These factors often include vehicle load, driving speed, route characteristics, weather conditions, and driver behaviour. Each of these variables can serve as the basis for an auxiliary task. For instance, an auxiliary task might involve predicting the impact of vehicle load on energy consumption under different traffic conditions or estimating the effect of varying driving speeds on battery usage. Historical data from real-world vehicle operations can be mined to extract relevant patterns and correlations, which can then be used to define these auxiliary tasks. In this paper, we have decided to focus on the patterns within the forecasted value itself instead of exploiting multivariate vehicle signals. In particular, we define several types of energy consumption characteristics as targets for the auxiliary tasks, such as questioning whether the consumption in the next time frame exceeds the global mean, whether the consumption will be higher in the next time frame compared to the current consumption, or predicting the consumption difference between the start and the end of the next time frame. These tasks are general enough to be suitable for any forecasting task, while at the same time being sufficiently closely related to the actual primary task to, hopefully, provide useful information to boost the training process.

The core contribution of this paper is the evaluation of applying several multi-task curriculum learning techniques for forecasting the energy consumption of heavy-duty electric vehicles, including the proposition of utilising key consumption characteristics as targets for generating auxiliary tasks for MCL. Comparison of MCL variations, with combinations of curriculum learning strategy (sequential learning and progressive continual learning) and auxiliary tasks, illustrates the improvements in the performance on a real-world data collected from normal operations of commercial transportation electric vehicles. The experimental results show progressive continual learning, with a logistic growth weighting function governing the learning balance between the primary and the auxiliary task, achieves the best performance; the result also shows that the first auxiliary task is the most helpful task for subsystems 1 and 4; the third auxiliary task is the most helpful task for subsystems 2 and 3. Furthermore, it is observed that MCL with the proposed auxiliary tasks can improve the learning efficiency of the model, achieving faster convergence to a point beyond which the gain from further training is limited.

2. Related Work

Curriculum learning enables the training of machine learning models in a meaning order, from easy samples to sets of difficult and complex samples [1]. A Common approach for CL introduces easy-to-hard ordering of samples for the training process, e.g., vanilla CL, self-paced CL, balanced CL, etc. When multiple tasks are available, the easy-to-hard ordering of the tasks to be learned can be applied as well. Multi-task learning can be applied, by sharing information across a set of related tasks in the training process, and the performance can be further improved [2] via, e.g. Gradnorm [3] balancing the losses between multiple tasks. While most multi-task learning approaches aim at learning multi-tasks simultaneously, progressive curriculum learning

allows determining the best order to learn multiple tasks to maximise the final result. Study [4] presented by Pentina et al. finds the best order of tasks to be learned in a sequence based on a generalisation bound criterion to optimise the average expected classification performance over all the tasks. Work [5] by Siahpour et al. introduced a penalty coefficient, as a function of the epoch step, to govern the training process by suppressing the loss, and noise respectively, from the domain discrimination task in the early stage, to ensure the efficient training of neural networks. Shi et al. proposed progressive contrastive learning [6] based on multi-prototypes in the dataset, the training process is ordered to learn the centroid prototype first, followed by the hard prototype, and finally the dynamic prototype. In this work, we explore sequential learning and progressive continual learning with a set of auxiliary tasks generated based on key characteristics of target signal.

3. Problem Formulation

For a given primary learning task $\mathcal{T}_i$, we create a set of auxiliary tasks $\mathcal{T}_i^j$, where $\mathcal{T}_i$ corresponds to the primary task (in our case, the forecasting of energy consumption for the i-th auxiliary subsystem in an electric truck), and $\mathcal{T}_i^j$ corresponds to the j-th type of auxiliary task.

The majority of the multi-task learning studies aim to learn all relevant tasks together to improve the performance for each task $\mathcal{T}_i$. In our study, we are only interested in improving the energy forecasting tasks $\mathcal{T}_i$, not the generated auxiliary tasks $\mathcal{T}_i^j$. All energy forecasting tasks and the auxiliary tasks are learned from the same dataset, multi-variate time series sensor readings were collected from the normal operations of several heavy-duty electric vehicles.

Let us denote data of the multivariate time series $\mathbf{x}$ of each vehicle v by $X = \{ x_{v,t}^k \mid t = 1, 2, ..., T_e(v), \ k = 1, 2, ..., K \}$, where $x_{v,t}^k$ is the value of the k-th feature $\mathbf{x}$ given a vehicle/trajectory v at time t, and $T_e(v)$ corresponds to the end of the recording. A subset of the features $u_{v,t}^i$ reflects the energy consumption of subsystem i at time t. The target energy consumption y_{v,t_0}^i in a future time frame τ_{ph} can be approximated by summing up the energy consumed over this time frame $y_{v,t_0}^i = \sum_{t \in [t_0, t_0 + \tau_{ph}]} p^i(t) \cdot \Delta t$, where $p^i(t)$ is the power consumption at time t, and Δt is the time interval between two samples.

In this study, we set τ_{ph} equal to 10 minutes. For a given forecasting task $\mathcal{T}_i$, a regression model $f_i^j(\cdot)$ is trained together with one of the auxiliary tasks $\mathcal{T}_i^j$ to estimate consumption $y_{v,t}^i$. In this study, neural networks, a shared feature extractor, with multiple heads, each corresponding to one task, were trained under different settings and evaluated for their performance after 200 training epochs. We explore different multi-task curriculum learning settings and auxiliary tasks for forecasting energy consumption. The MCL methods were compared to the traditional approach

4. Method

4.1. Auxiliary Tasks

For a given regression task $\mathcal{T}_i$ (forecasting energy consumption for one of the subsystems), a set of auxiliary tasks was generated to assist the learning progress. We explore the use of five types

of consumption characteristics as targets for creating the auxiliary tasks: i) $\mathcal{T}_i^1$: classifying whether the consumption in the next time frame exceeds the global mean for that subsystem i; ii) $\mathcal{T}_i^2$: classifying whether the consumption will increase in the next time frame, compared with the current consumption; iii) $\mathcal{T}_i^3$: classifying whether the consumption at the end of the next time frame exceeds the starting point; iv) $\mathcal{T}_i^4$: predicting the consumption difference between the start and the end of the next time frame; v) $\mathcal{T}_i^5$ predicting the difference between the peak consumption and the lowest consumption in the next time frame. The first three auxiliary tasks are classification task, the other two tasks are regression task. Learning to predict these key consumption characteristics in these auxiliary tasks $\mathcal{T}_i^j$, along with the primary tasks $\mathcal{T}_i$, under MCL, are evaluated for their usefulness.

4.2. Network Architecture

The regression model evaluated for MCL in this study builds on a multi-layer perceptron. The model is comprised of a shared feature extractor and two heads, one head carries out the main task $\mathcal{T}_i$, and the other corresponds to one of the five auxiliary tasks $\mathcal{T}_i^j$. The network architecture is illustrated in Figure 1. For auxiliary tasks that are classification tasks, a sigmoid function was applied to the output of the corresponding head.

4.3. Curriculum Learning Strategy

The two curriculum learning strategies evaluated in this work are sequential learning (SeqL) and progressive continual learning (PCL). The overall optimisation loss $\mathcal{L}$ can be defined as:

$$\mathcal{L} = \lambda \mathcal{L}_{\mathcal{T}_i} + (1 - \lambda)\mathcal{L}_{\mathcal{T}_i^j} \tag{1}$$

where $\mathcal{L}_{\mathcal{T}_i}$ denotes the loss for the primary tasks, while $\mathcal{L}_{\mathcal{T}_i^j}$ denotes the loss for the auxiliary task j. The SeqL employed imposes a fixed ordering of the tasks, e.g. learning the auxiliary task first, before a predetermined epoch, and the primary task afterwards:

$$\lambda_{SeqL} = \begin{cases} 0 & \text{if } \eta < \mathcal{N}_{ep} \\ 1 & \text{if } \eta \geq \mathcal{N}_{ep} \end{cases} \tag{2}$$

where η is the current training epochs, and $\mathcal{N}_{ep}$ is the number of epochs predetermined to switch to another task.

The PCL employs a weighting mechanism, a function of training epochs, to govern the learning process and gradually increases the weights on the loss corresponding to the primary task:

$$\lambda_{PCL} = \frac{2}{1 + exp(-10\alpha\eta/\mathcal{N}_{tot})} - 1 \tag{3}$$

where α is a coefficient governing the change rate (see Figure 2 for an illustration), η is the current training epochs, and $\mathcal{N}_{tot}$ is the total amount of training epochs. The two curriculum learning strategies were compared with MTL without any special curriculum learning and learning only on the primary task $\mathcal{T}_i$
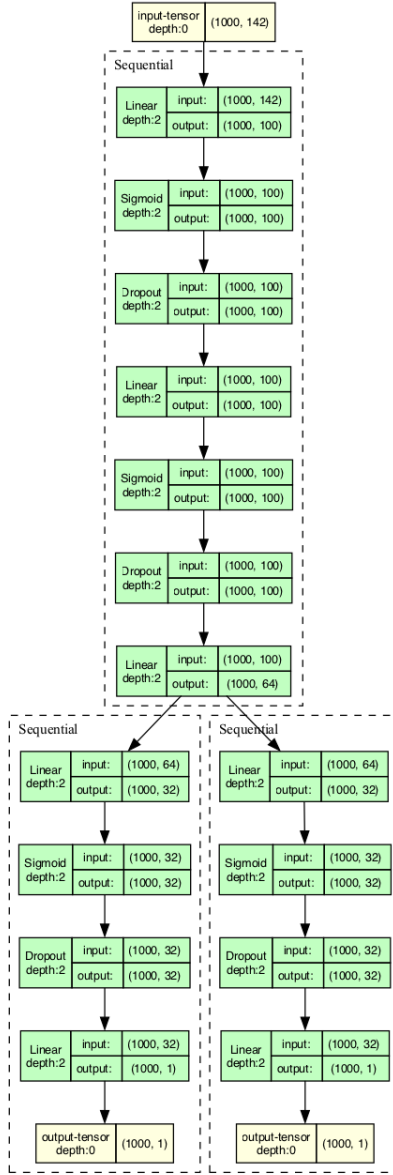
Figure 1: MLP network architecture

The two evaluation criteria in this study are (i) the test loss (Mean Absolute Error, MAE) after training converged, i.e., N_{tot} epochs, and (2) whether the proposed learning strategy achieves a faster convergence time, i.e., the epoch at which the test loss has reached a saturation point (no further significant decrease in the loss afterwards). In each case, different variants of MCL are compared against a learning process without any multi-task curriculum learning. The saturation point is detected using a knee point detection algorithm [7], proposed by Satopaa et al.
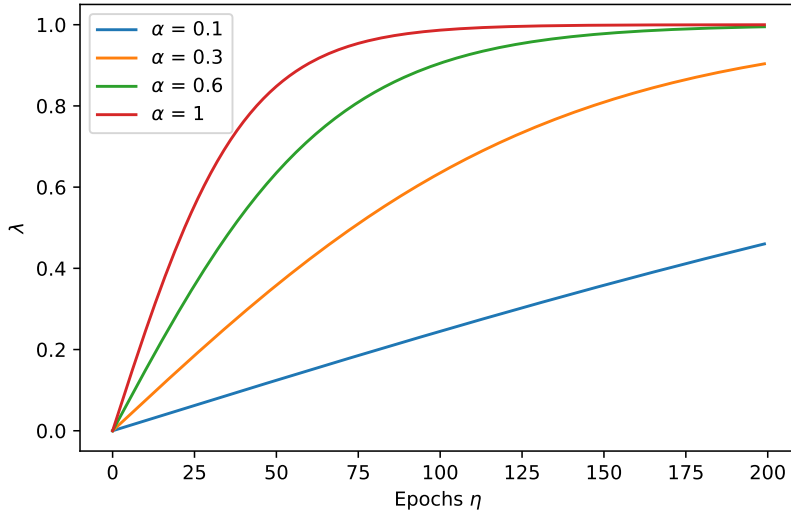
Figure 2: PCL weighting function

5. Experiment Result

The energy consumption dataset was collected from several electric trucks operating in different countries for a couple of months, including sensor readings of mileage, speed, ambient temperature, and energy consumed for auxiliary subsystems, etc., from sessions of driving. The four subsystems we forecast energy consumption for are the air compressor $\mathcal{T}_1$, the air conditioner $\mathcal{T}_2$, the cabin heater $\mathcal{T}_3$, and the heater of the energy storage system $\mathcal{T}_4$.

For the experiment conducted in this paper, the neural networks were implemented via Pytorch library [8], using an ADAM optimiser with a learning rate of 0.001. The loss function for the regression tasks is mean absolute error (MSE), and binary cross-entropy (BCE) was employed as the loss function for the classification tasks. The total number of training epochs $\mathcal{N}_{tot}$ is set to 200. For the sequential learning strategy, $\mathcal{N}_{ep}$ is set to 100, and for the progressive continual learning, α value of 0.1 (i.e. a linear function) and 0.3 are tested. The experiments were conducted using 4-fold cross-validation driving session-wise, i.e. data from the same driving session would never appear in the training and the testing population together.

Table 1 and Table 2 show the training and testing losses after 200 epochs of training of the neural networks using multi-task learning with any curriculum learning (MTL), sequential learning (SeqL), progressive continual learning with an α of 0.1 (PCL-lin), and an α of 0.3 (PCL-exp). The baseline performance, single task learning (STL), is produced with learning only on the primary task $\mathcal{T}_i$ for each subsystem, shown in the parenthesis. It is shown in both tables that the lowest averaged MSE is achieved using PCL-exp. As a sanity check, Table 1 demonstrated that the training losses, after 200 epochs of training, of most MCL methods did converge to a level comparable to STL. For the testing losses shown in Table 2, applying PCL-exp on task sets $\{\mathcal{T}_1, \mathcal{T}_1^1\}$ and $\{\mathcal{T}_4, \mathcal{T}_4^1\}$ achieved lowest averaged MSE for forecasting energy consumption of subsystem 1 and 4 (i.e., the first auxiliary task appears to be the most

Table 1

Comparison of training loss after the 200 epochs for different MCL approaches using different auxiliary tasks. The reference performances (using STL) are placed in parentheses. MCL results outperforming the baseline are highlighted in bold, and the best performance for each subsystem is underlined.

Task1 (0.6202 ± 0.0163)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**0.6125 ± 0.0141**	**0.6054 ± 0.0167**	**0.609 ± 0.0185**	<u>**0.6013 ± 0.0156**</u>
AuxTask2	0.6374 ± 0.0147	0.6502 ± 0.0138	0.6802 ± 0.0096	0.6435 ± 0.0156
AuxTask3	**0.6165 ± 0.0166**	**0.6131 ± 0.0118**	**0.6076 ± 0.0154**	**0.6033 ± 0.0155**
AuxTask4	0.6239 ± 0.0132	0.626 ± 0.0119	0.6567 ± 0.0161	**0.6182 ± 0.0121**
AuxTask5	0.6256 ± 0.0113	**0.6152 ± 0.0121**	0.625 ± 0.0147	**0.614 ± 0.0117**

Task2 (0.2617 ± 0.0245)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	0.2681 ± 0.023	0.276 ± 0.0128	0.2959 ± 0.0157	**0.2541 ± 0.0171**
AuxTask2	0.2619 ± 0.0158	0.3016 ± 0.0308	0.2939 ± 0.0204	**0.2475 ± 0.037**
AuxTask3	0.2662 ± 0.0395	0.2862 ± 0.0324	**0.2379 ± 0.0245**	<u>**0.2158 ± 0.0186**</u>
AuxTask4	**0.2534 ± 0.011**	0.2866 ± 0.0255	0.2795 ± 0.0202	**0.2366 ± 0.0432**
AuxTask5	0.2638 ± 0.0168	0.2691 ± 0.0361	0.2971 ± 0.0167	**0.2436 ± 0.0285**

Task3 (0.3173 ± 0.0115)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	0.3223 ± 0.0132	**0.3138 ± 0.0084**	0.3248 ± 0.012	**0.3116 ± 0.0111**
AuxTask2	0.3217 ± 0.0109	0.3222 ± 0.0116	0.3423 ± 0.0113	**0.317 ± 0.0096**
AuxTask3	**0.3148 ± 0.0074**	**0.3117 ± 0.0151**	0.3229 ± 0.0121	<u>**0.3018 ± 0.0116**</u>
AuxTask4	0.333 ± 0.0126	0.3272 ± 0.0137	0.356 ± 0.0214	0.3188 ± 0.0152
AuxTask5	0.3213 ± 0.0103	0.3188 ± 0.0143	0.3412 ± 0.0091	**0.3171 ± 0.0122**

Task4 (0.2936 ± 0.0129)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**0.2903 ± 0.0186**	0.3248 ± 0.0159	**0.2684 ± 0.0208**	**0.2646 ± 0.0124**
AuxTask2	0.2941 ± 0.0123	0.3511 ± 0.0171	0.3565 ± 0.0866	<u>**0.2583 ± 0.0156**</u>
AuxTask3	0.2979 ± 0.0136	0.3064 ± 0.0165	**0.2624 ± 0.0081**	0.344 ± 0.152
AuxTask4	0.3269 ± 0.0127	0.3712 ± 0.0117	0.425 ± 0.035	0.329 ± 0.0275
AuxTask5	0.3145 ± 0.0142	0.3142 ± 0.0311	0.3334 ± 0.0076	0.3036 ± 0.0182

helpful auxiliary task for subsystem 1 and 4); similarly, applying PCL-exp on task sets $\{\mathcal{T}_2, \mathcal{T}_2^3\}$ and $\{\mathcal{T}_3, \mathcal{T}_3^3\}$ achieved lowest averaged MSE for forecasting energy consumption of subsystem 2 and 3.

Figure 3 illustrates the differences between several multi-task curriculum learning strategies, focusing on the convergence speed. Specifically, we identify a reference point (epoch) beyond which the gain from further training is limited. This reference point is computed using a knee point detection method (algorithm [7] by Satopaa et al.) on the mean STL test losses (shown as grey dots and the corresponding dash line). The four plots in Figure. 3 illustrate the testing loss for learning the four primary tasks, along with their 5-th auxiliary task, i.e. $\mathcal{T}_i^5$. It is observed in Figure 3: i) there is no significant difference between the four approaches for $\mathcal{T}_1$; ii) MTL and PCL-lin drop slightly slower compared to STL and PCL-exp for $\mathcal{T}_2$; iii) both PCL approach drop slower compared with STL and MTL for $\mathcal{T}_3$; iv) MTL, PCL-lin, and PCL-exp drops faster
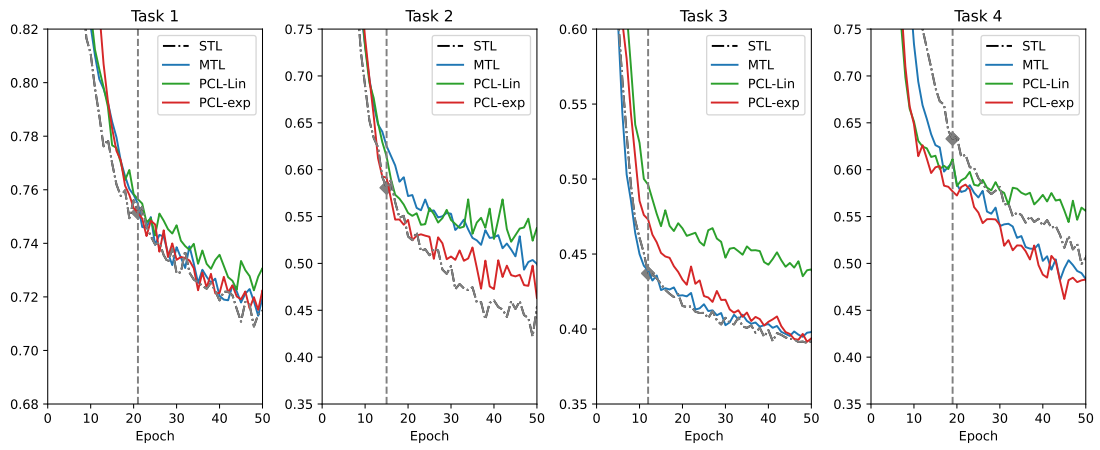
Table 2

Comparison of test loss after the 200 epochs for different MCL approaches using different auxiliary tasks. The reference performances (using STL) are placed in parentheses. MCL results outperforming the baseline are highlighted in bold, and the best performance for each subsystem is underlined.

Task1 (0.6861 ± 0.0713)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**0.6829 ± 0.0707**	0.692 ± 0.072	**0.6827 ± 0.0727**	**0.6784 ± 0.0672**
AuxTask2	0.6948 ± 0.0736	0.7037 ± 0.0749	0.7248 ± 0.0702	0.7076 ± 0.0709
AuxTask3	**0.6812 ± 0.0744**	0.6969 ± 0.0634	0.698 ± 0.0762	0.6943 ± 0.0774
AuxTask4	0.6917 ± 0.0775	0.6934 ± 0.069	0.7058 ± 0.0634	0.6881 ± 0.0684
AuxTask5	0.6968 ± 0.0712	0.6894 ± 0.0735	0.6913 ± 0.074	0.6862 ± 0.073
Task2 (0.4374 ± 0.1821)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	0.4553 ± 0.2035	**0.4199 ± 0.1608**	**0.4277 ± 0.1788**	**0.4285 ± 0.177**
AuxTask2	**0.4322 ± 0.1671**	0.4448 ± 0.1766	**0.4319 ± 0.1676**	**0.4329 ± 0.1678**
AuxTask3	**0.4109 ± 0.1556**	0.4427 ± 0.1782	**0.4105 ± 0.1398**	**0.3929 ± 0.1602**
AuxTask4	**0.4105 ± 0.1632**	0.4436 ± 0.1673	0.4699 ± 0.1928	**0.4362 ± 0.1752**
AuxTask5	0.4702 ± 0.2093	0.4734 ± 0.2158	0.4874 ± 0.2344	0.4632 ± 0.2307
Task3 (0.3827 ± 0.0551)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**0.3777 ± 0.0593**	**0.377 ± 0.0627**	0.3829 ± 0.0609	**0.3774 ± 0.0577**
AuxTask2	0.387 ± 0.0503	0.3901 ± 0.0618	0.401 ± 0.0534	0.3892 ± 0.0659
AuxTask3	0.3859 ± 0.0534	0.3857 ± 0.0578	0.3868 ± 0.0588	**0.3766 ± 0.0684**
AuxTask4	0.3847 ± 0.0683	0.3828 ± 0.0563	0.4021 ± 0.0724	0.3865 ± 0.0602
AuxTask5	0.3874 ± 0.0587	0.3868 ± 0.0605	0.4016 ± 0.0563	**0.38 ± 0.0549**
Task4 (0.4166 ± 0.0679)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**0.4155 ± 0.065**	0.4783 ± 0.0698	0.4332 ± 0.0801	**0.3986 ± 0.0567**
AuxTask2	**0.4085 ± 0.0745**	0.4786 ± 0.0853	0.5225 ± 0.1217	0.4339 ± 0.0813
AuxTask3	0.424 ± 0.0492	0.5029 ± 0.0554	0.4253 ± 0.054	0.4874 ± 0.1068
AuxTask4	0.4251 ± 0.0774	0.4548 ± 0.0767	0.4936 ± 0.0718	0.4375 ± 0.0664
AuxTask5	0.4291 ± 0.0613	0.4442 ± 0.0662	0.4535 ± 0.0651	0.4347 ± 0.0716

compared to STL.

Table 3 shows a comparison between MCL methods on the convergence time to the reference point (computed based on STL mean testing losses over the four folds). It is observed that: i) MTL outperforms STL in all four primary tasks, and converged to the reference point faster than other approaches in three out of four primary tasks; ii) the performance of PCL-lin achieved converged fast for two of the tasks; iii) PCL-exp achieved better performance compared to PCL-lin, with overall short convergence time. The result corresponding to SeqL is particularly interesting. Although a $\mathcal{N}_{ep}$ of 100 epochs is adopted for SeqL (i.e. trained on one of the auxiliary tasks for the first 100 epochs before learning the primary task), the testing loss converges to the reference point within 10 epochs in the majority of the cases. From an empirical perspective, the proposed auxiliary tasks assisted the learning (of the models) for the primary task, resulting

Figure 3: Comparison of convergence speed for different MCL approaches and auxiliary tasks.

in a faster convergence time to the reference point.

Table 3
Comparison of convergence speed to reach a point beyond which the gain from further training is limited. The reference point is given by STL loss sequences averaged over 4 folds. MCL results outperforming the baseline are highlighted in bold, and the best performance for each subsystem is underlined.

Task1 (20 Ep.)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**13.0 ± 1.4142**	102.25 ± 0.433	**16.75 ± 2.2776**	**16.25 ± 8.9268**
AuxTask2	32.5 ± 4.0311	125.75 ± 2.2776	103.5 ± 28.8141	45.75 ± 26.6962
AuxTask3	**15.75 ± 3.8971**	106.5 ± 1.118	29.0 ± 3.3912	22.25 ± 8.1968
AuxTask4	<u>**10.5 ± 1.5**</u>	106.5 ± 2.2913	70.5 ± 32.7605	29.0 ± 15.1493
AuxTask5	**15.25 ± 6.7961**	102.5 ± 0.5	**18.0 ± 4.3012**	**19.0 ± 9.083**

Task2 (19 Ep.)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**15.625 ± 7.9047**	58.875 ± 50.5827	34.0 ± 22.7211	20.125 ± 14.4606
AuxTask2	**18.0 ± 9.2736**	108.25 ± 9.8075	44.5 ± 14.239	23.5 ± 11.4127
AuxTask3	26.0 ± 16.4773	106.75 ± 7.9804	42.5 ± 12.0312	22.75 ± 7.1545
AuxTask4	32.0 ± 33.2039	111.25 ± 13.3112	60.3333 ± 21.6384	32.0 ± 18.8149
AuxTask5	<u>**17.0 ± 11.2916**</u>	108.5 ± 8.6168	100.0 ± 86.0145	28.0 ± 24.8697

Task3 (14 Ep.)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**9.25 ± 5.4025**	104.5 ± 3.5707	22.5 ± 20.6458	**10.25 ± 4.8153**
AuxTask2	<u>**8.5 ± 7.7298**</u>	58.75 ± 50.5118	36.125 ± 38.6505	21.875 ± 19.3354
AuxTask3	**9.75 ± 7.9175**	105.75 ± 3.6997	37.25 ± 20.216	18.0 ± 8.2765
AuxTask4	**11.25 ± 7.5291**	105.75 ± 3.2692	55.75 ± 36.8536	24.5 ± 11.9478
AuxTask5	**10.0 ± 8.6891**	103.75 ± 3.0311	36.25 ± 25.4497	22.25 ± 12.8331

Task4 (19 Ep.)	MTL	SeqL	PCL-lin	PCL-exp
AuxTask1	**14.75 ± 2.1651**	107.5 ± 3.3541	25.5 ± 2.958	21.5 ± 1.118
AuxTask2	**15.75 ± 4.867**	106.0 ± 1.5811	28.5 ± 3.8406	20.75 ± 2.3848
AuxTask3	**14.75 ± 2.9896**	61.375 ± 47.7178	**18.875 ± 8.1**	**17.1429 ± 4.4538**
AuxTask4	**16.5 ± 5.6789**	107.5 ± 1.5	41.5 ± 22.0057	24.0 ± 13.6015
AuxTask5	**15.0 ± 4.3589**	105.5 ± 3.2016	**14.75 ± 3.562**	<u>**14.0 ± 4.3589**</u>

6. Conclusion and Future Work

In this work-in-progress paper, several multi-task curriculum learning strategies were evaluated for forecasting the energy consumption of auxiliary subsystems in heavy-duty electric vehicles. The preliminary results show that progressive continual learning has achieved the best performance (lowest averaged MSE) compared to multi-task learning with any CL, Sequential CL, and the traditional approach (STL). Moreover, the proposed auxiliary tasks, based on key consumption characteristics, for CL have shown to be useful for solving all four primary tasks, both in terms of regression error as well as convergence speed.

Future works include: (i) developing methods to rank/order the auxiliary tasks based on relevancy and select the top k number of tasks for CL; (ii) proposing adaptive methods for governing the learning process, e.g. based on weighting the losses of auxiliary tasks using

learning dynamics; (iii) enabling CL across primary tasks, based on task relevancy.

Acknowledgments

References

[1] P. Soviany, R. T. Ionescu, P. Rota, N. Sebe, Curriculum learning: A survey, International Journal of Computer Vision 130 (2022) 1526–1565.

[2] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Transactions on Knowledge and Data Engineering 34 (2021) 5586–5609.

[3] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: International conference on machine learning, PMLR, 2018, pp. 794–803.

[4] A. Pentina, V. Sharmanska, C. H. Lampert, Curriculum learning of multiple tasks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5492–5500.

[5] S. Siahpour, X. Li, J. Lee, A novel transfer learning approach in remaining useful life prediction for incomplete dataset, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–11.

[6] J. Shi, X. Yin, Y. Wang, X. Liu, Y. Xie, Y. Qu, Progressive contrastive learning with multi-prototype for unsupervised visible-infrared person re-identification, arXiv preprint arXiv:2402.19026 (2024).

[7] V. Satopaa, J. Albrecht, D. Irwin, B. Raghavan, Finding a" kneedle" in a haystack: Detecting knee points in system behavior, in: 2011 31st international conference on distributed computing systems workshops, IEEE, 2011, pp. 166–171.

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).