A Novel Mechanism for Continual Learning based Predictive Quality Inspection in Smart Manufacturing

Garima Nain[1], K. K. Pattanaik[*,1], G. K. Sharma[1], Himanshu Gauttam[1] and Wattana Viriyasitavat[2]

Abstract— **Edge-enabled Deep Learning (DL) solutions for *Predictive Quality Inspection (PQI)* of products in Industry 4.0 are mostly designed for static manufacturing environments. In general, modern manufacturing processes are dynamic in nature. In this context, continual learning-based model retraining accommodates the dynamism for *PQI* of multiple processes (tasks) using a single DL model. However, the impact of the task ordering in sequentially arriving tasks and solution to reduce this impact on the overall *PQI* is yet to be solved. To this end, a novel mechanism using a light-weight similarity analysis module is introduced in the quality prediction system at the resource-limited edge. Sequential training of tasks above a similarity threshold (γ) is preferred, and dissimilar tasks are overlooked to train a separate model. This enables a *PQI* system to hover over training efficiency and model sustainability. The experimental results validate the impact of task order and the effectiveness of the proposed similarity-based analysis to reduce this impact by 70% on the model's overall performance in the real-world use case of plastic bricks.**

I. INTRODUCTION

Edge-enabled DL solutions act as key enabler for real-time data-driven quality prognosis of manufactured products, i.e., *Predictive Quality Inspection (PQI)*, in Industry 4.0 driven smart manufacturing. *PQI* follows a Digital-Physical-Digital (D2D) loop, which involves sensing and data collection (physical-to-digital conversion), data analysis (digital-to-digital conversion), and actuation (digital-to-physical conversion) for real-time quality predictions and provoking corrective measures based on it [1], [2]. DL-based *PQI* methods are much trumpeted in literature for quality inspection and prediction utilizing process and machine data [3], vision-based data [4], [5], acoustic signals [6], etc. Additionally, these solutions are also attaining attention in manufacturing processes like laser machining [7], additive manufacturing or 3D printing [8], and many more.

Most DL-based *PQI* mechanisms are traditionally designed for static manufacturing environments under surplus data availability [9], [10]. However, current industries are adopting flexible production processes and manufacturing solutions to chase and fulfil customers' demands. These solutions are subject to continuous changes in the process data, intending varying relationships among the process parameters and predictive quality parameters. Thus, these static DL models become obsolete and usher in poor-quality prediction over the new process data that in turn hampers application performance.

To this end, one possible solution is to train the model with all the new and old process data, but training from scratch becomes infeasible under the constraint of inaccessible old process data due to industry regulations or corporate policies [11] and limited storage availability (at the edge) [3]. Another solution is collecting enormous process data to train new quality inspection models for each possible production process. Nonetheless, it turns expensive, time-consuming and limits the sustainability of DL-based *PQI* of manufactured products. Propagating to the sequential training of a single DL model on new data, i.e., finetuning, leads to forgetting old task knowledge (catastrophic forgetting).

In this context, Continual Learning (CL) schemes address given research gaps and efficiently train a signal neural network across varying production tasks. The use of CL schemes in industrial applications is in its infancy; for instance, Lesort et al. [12] and Dehghan et al. [13] adopted these strategies to incrementally acquire knowledge in robotic systems, and Maschler et al. [14] intended their use for fault prognosis of engines. Regularization-based CL schemes are more prevalent in the literature, as need not to access old data, have frozen storage requirements not growing with tasks, and accommodates significant number of tasks [15]. Therefore, Tercan et al. [3] presented a regularization-based CL scheme, Memory Aware Synapses (MAS), to ensure the trained model acclimates the current training task as well as maintains performance across all the previous tasks.

However, the proposed solutions [3], [12], [13], [14] in the literature don't discuss the impact of task order on the sequential training of multiple tasks in the MAS scheme for the quality prediction of manufactured products. This gives rise to two Research Questions (RQs):

- *RQ-1: Does a task order impact overall model training, and if it does, how to reduce it?*
- *RQ-2: Should all newly arriving tasks be accommodated in a single DL model in continual training?*

In light of the above discussion, this paper first analyses the impact of task order or sequence on the performance of a CL scheme like MAS. Analysis indicates that different task orders have different performances (refer to Section VI-A for detailed description), and some tasks influence the overall sequence more than others. Figure 1 illustrates the impact of task ordering on the overall quality inspection system. The final DL-model learns different final input-output relationships for different sequences. Additionally, RQ2, motivates to rethink a solution to maintain trade-off between continually training a single DL model for *N* tasks

*K. K. Pattanaik is the corresponding author

[1]Garima Nain, K. K. Pattanaik, G. K. Sharma and Himanshu Gauttam are with the Department of Information Technology, ABV-IIITM, Gwalior, India garimanain96@gmail.com, kkpatnaik@iiitm.ac.in, gksharma@iiitm.ac.in, hgauttam262@gmail.com

[2]Wattana Viriyasitavat with the Chulalongkorn Business School, Pathumwan, Thailand hardgolf@gmail.com

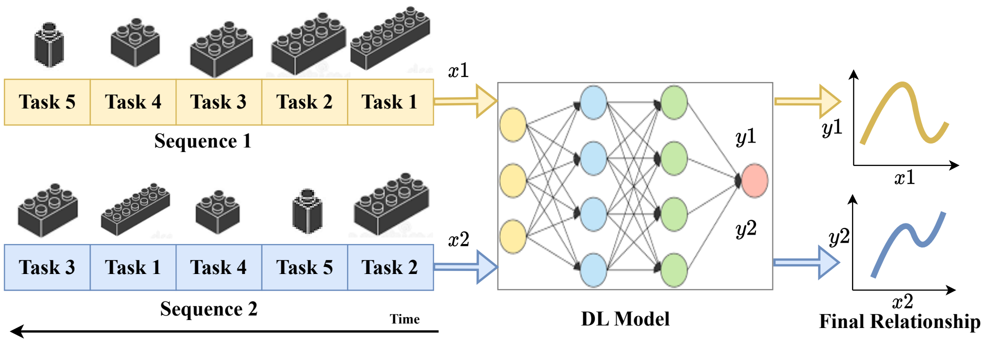

Fig. 1. Illustration of the impact of task ordering on the final input-output relationship learned by DL model for PQI

over traditionally training N models for N incrementally arriving tasks at the cost of storage.

The key component of any manufacturing industry is to ensure pre-defined quality measures and tolerances through efficient inspection and maintenance of manufactured products [16], [17]. Therefore, this work proposes a mechanism to improve the overall performance of continual training tasks by determining whether to accommodate or not any sequential task within the existing model and to diminish the impact of the task order. Consequently, a similarity module is proposed to minimize the impact of task orders and better maintain the trade-off between training efficiency and model sustainability. Following are the three key contributions of the paper:

- Analyses the impact of task ordering on the performance of the MAS scheme for incorporating new *PQI* tasks within the same DL model.
- Proposes a solution to minimize the impact of task orders on MAS and to better maintain the tradeoff between training efficiency and model sustainability at resource-limited edge.
- Presents practical application of the proposed solution in the real-world use case of injection molding for *PQI*.

The remaining manuscript is arranged as follows: Section II describes the two-tier system model for *PQI*. Section III briefly introduces MAS based CL scheme and similarity measures. The proposed methodology is provided in Section IV. Experimental background, use-case description followed by results are recapitulated in Section V and VI. Finally, Section VII briefs the conclusion and an outlook on future research.

II. SYSTEM MODEL

The following work adopts a two-tier architecture for the data-driven quality prediction of products using in-process manufacturing data. The overall *PQI* system consists of a sensory device layer, communication links, and a computing layer [11], [18]. Fig. 2 depicts the two-layer predictive quality inspection system in manufacturing industries.

- Device Layer: This layer captures and collects the process parameters from various sensory assets (such as, temperature, pressure, vibration sensors) during the manufacturing process. Additionally, some quality prediction parameters are collected from the finished products. These devices fuse and transmit the data to the computing layer over a wireless communication channel.

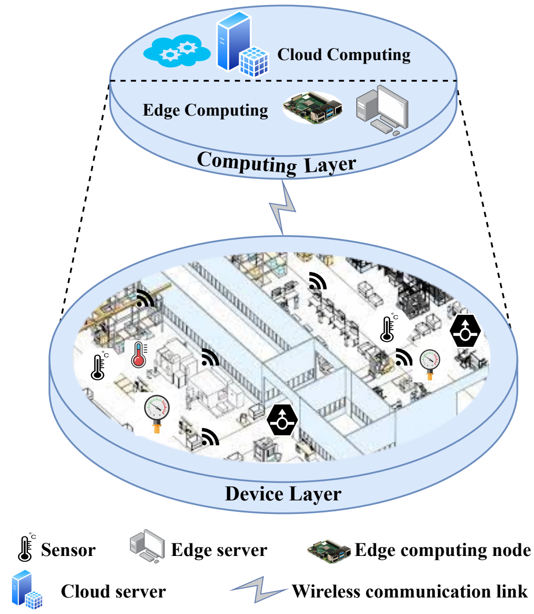

Fig. 2. A two-layer edge-enabled *PQI* system in smart manufacturing

- Computing Layer: This layer performs the data cleaning and processing of the process parameters collected from the device layer. It comprises two computing platforms: edge computing and cloud computing. The edge computing [18] platform encloses on-site devices such as single-board computers, industrial PCs, etc., whereas the cloud computing platform contains remote data centres. Using the DL-based *PQI* models, this layer produces the quality prediction of the manufactured product. Edge devices are preferred as introduces enhanced security and latency but have resource-limited devices.

III. PRELIMINARIES

A. Memory Aware Synapses (MAS)

MAS [19] is a regularization-based CL scheme wherein the importance values are determined for each parameter of the neural network. The importance value, $\Omega_{P,i}$ of a model parameter, $\theta_{P,i}$ is calculated by accumulating the sensitivity of the estimated output value to change in this parameter over given u data samples X_P^j, $j = 1, 2, ..., u$ of Task P as given in (1).

$$\Omega_{P,i} = \frac{1}{u}\sum_{j=1}^{u}\left|\frac{\delta l_2^2(O_P(X_P^j), 0)}{\delta \theta_{P,i}}\right| \quad (1)$$

where, O_P is the output of the model for X_P^j. For the continual training of Task Q, with training data X_Q^k, $k = 1, 2, 3, ..., v$, the overall loss function for the MAS strategy is presented in (2).

$$L(\theta_{Q,i}) = L_Q(\theta_{Q,i}) + \lambda \sum_i \Omega_{P,i}(\theta_{Q,i} - \theta_{P,i})^2 \quad (2)$$

where, λ is a real positive hyperparameter to control regularization, and $\theta_{Q,i}$ are updated model parameters when training Task Q. At the time of training of Task Q, the parameters change, i.e., $(\theta_{Q,i} - \theta_{P,i})$, is penalized according to the importance value $\Omega_{P,i}$. It ensures averting important knowledge being forgotten corresponding to previous Task P after the training of Task Q. Similarly, it can accommodate new tasks sequentially within the same model.

B. *Similarity Measures*

In order to determine the similarity among the two tasks, first, a similarity measure [20] is to be selected. The literature is flooded with tremendous similarity techniques, however computationally lightweight schemes are preferred for edge devices. Two lightweight similarity measures, i.e., Cosine similarity and Euclidean distance are introduced. Let $P = (p_1, p_2, p_3, ..., p_n)$ and $Q = (q_1, q_2, q_3, ..., q_n)$ be two n-dimensional data features, then the Cosine similarity and Euclidean distance are calculated as follows:

1) Cosine Similarity: The Cosine similarity (refer to (3)) for real-valued vector spaces P and Q is $C_{sim}(P,Q)$.

$$C_{sim}(P,Q) = \frac{P \cdot Q}{||P||\,||Q||} \quad (3)$$

where, $||P||$ and $||Q||$ represent the Euclidean norm of data features P and Q, respectively. The Cosine similarity is between [0, 1], where higher values (close to 1) validate that the vectors are more similar, and lower values (close to 0) express dissimilarity.

2) Euclidean Distance: Euclidean distance is the most common distance measure functional for similarity analysis of the numeric attributes/features. Its numerical formulation is given in (4).

$$d(P,Q) = ||P - Q||_0 = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (4)$$

$d(P,Q)$ is same as the L2 norm and a positive integer value, where high values denote higher dissimilarity and vice-versa.

IV. PROPOSED METHODOLOGY

As aforementioned, Tercan et al. [3] proposed a MAS-based CL scheme for learning multiple tasks in a single quality inspection DL model. However, the impact of the task order is not conferred when sequential tasks arrive. Any task order can significantly impacts the model's overall performance when trained over sequentially arriving tasks (refer to Section VI-A). Finding the best task order also, may not solve the problem, since storing the data of any tasks for later training violates the underlying assumption (no access to old data) of CL. Therefore, in this context, a Cosine similarity-based module is added before training any new task sequentially in a DL model. The methodology of the proposed similarity-based model maintenance mechanism is shown in Fig. 3 for the *PQI*.

Let us assume Task 1, Task 2, and Task 3 arrive sequentially at any time instance, t_0, t_1, and t_2. At the arrival of any new Task, a similarity analysis module determines the similarity value, α, between new and old tasks. The similarity value is determined from the mean feature vector of the data samples of each task using a Cosine similarity scheme. A Cosine-based similar measure is adopted as it gives a similarity value between [0, 1]. As shown in Fig. 3, given tasks with similarities above a similarity threshold (γ) are allowed to be trained continually using the MAS scheme. This similarity threshold depends upon the required quality standards of the industry and the application. In this study, we determined

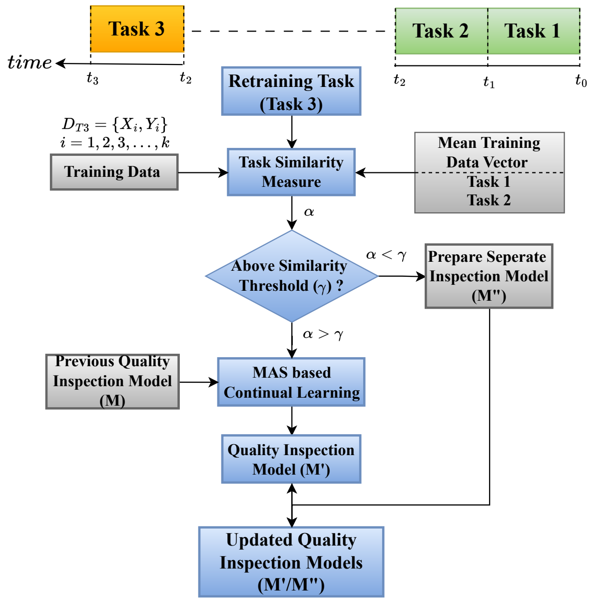

Fig. 3. The proposed similarity-based model maintenance mechanism

that similar sequential tasks (based on similarity measures) have better overall performance after training multiple tasks in comparison to varied tasks (similar and dissimilar tasks). Consequently, tasks with low similarity measures from all other previous tasks are trained separately in a new DL model for maintaining the better overall performance of all the quality inspection tasks. In this manner, the impact of the task ordering on the overall performance is reduced as all tasks are analysed (based on similarity) before training a single DL model. Thus, this work attempts to provide a tradeoff between model's sustainability and performance.

V. EXPERIMENTAL BACKGROUND

Injection molding is an important production process for manufacturing products from thermoplastics within a single manufacturing step [21]. It follows pouring or injecting the viscous plastic melt into the closed mold and extracting the finished product from the mold after cooling. This process involves complex relationships among various process variables (such as temperature, pressure, melt flow, etc.), and the relationship deviates when encompassing varying product designs. This work employs the plastic bricks dataset[1] [3] for the experimental analysis. The ten important features are demarcated using the correlation analysis among the available features, and the maximum deformation under the load is the target quality variable to be determined.

Five task formulation is accomplished from the available dataset based on the height (namely, *Flach*, *Fach*, *Lego*, *Hoch*, and *Gedrittelt* represented as *Fl*, *Fa*, *Le*, *Ho*, and *Ge*, respectively throughout the article) of the plastic brick for the continual training of tasks. Therefore, five quality inspection tasks with varying process parameters are considered. For better assurance, consecutive training of tasks was performed for five different task sequences and 25 different data shuffles. Any task order is represented by the sequence of task arrival denoted by their representation; for instance, *LeGeFaHoFl* has a task order as *Lego*, *Gedrittelt*, *Fach*, *Hoch*, and *Flach*. The summary of dataset and the experimental parameters are provided in Table I.

TABLE I
SUMMARY OF DATASET AND THE EXPERIMENTAL PARAMETERS

Plastic brick use case	Studs on brick top	1 or 2 rows, (1, 2, 3, 4, 6, 8) in each row
	Total plastic bricks types	60
	Samples of each brick	77
	Tasks based on height	5
Neural Network Specifications	Number of layers	3
	Neurons in hidden layer	20
	Learning rate	0.001
	Activation function and Optimizer	ReLU and ADAM
MAS	Hyperparameter λ	1000

Average Overall Loss, $\mathscr{L}_{ol}$ is a performance measure that gives the average mean squared error of all the tasks after continually training a sequence of n tasks. Mathematically, it can be represented as in (5).

$$\mathscr{L}_{ol} = \frac{1}{n}\sum_{m=1}^{n} L_{n,m} \tag{5}$$

where, $L_{n,m}$ gives the mean squared loss of Task m after training of n tasks. Thus, $\mathscr{L}_{ol}$ represents the overall performance of the single DL model after coordinating a sequence of n consecutive tasks.

VI. RESULT AND DISCUSSION

This section provides experimental verification to show the impact of task orders, and the proposed solution is validated for reducing its impact. Initially, the performance of the MAS scheme (with single output (MAS-SH) head and multiple heads (MAS-MH)) is examined over the baseline finetuning scheme. Table II shows that the overall loss of the finetuning scheme is very high for each task order since it has no regularization on the performance of previous tasks. Thus, their overall performance ($\mathscr{L}_{ol}$) degrades over continual learning of new tasks. whereas MAS-SH accomplishes little better than finetuning, but MAS-MH performed the best since it maps the output function appropriately by learning the new output head corresponding to each task. This signifies the capability of the MAS scheme to learn new knowledge without forgetting past knowledge. Therefore, in further studies, MAS-MH CL scheme is considered by default unless specified.

TABLE II
AVERAGE OVERALL LOSS AFTER TRAINING OF 5 TASKS USING DIFFERENT TRAINING SCHEMES

Task Order	Finetuning	MAS-SH	MAS-MH
LeGeFaHoFl	3.598	3.590	0.044
GeFaFlHoLe	3.537	3.518	0.067
FlLeHoGeFa	15.945	15.726	0.038
HoFlLeFaGe	4.969	4.853	0.140
FaGeFlLeHo	3.755	3.429	0.014

Further, this section validates the problem from the experimental results, examines the vision of task similarity, and investigate performance improvement and deduction of task ordering impact from the proposed scheme.

A. Problem Validation

This section aims to illustrate the impact of task orders on MAS-based training of a single *PQI* model for the real-world use case of plastic bricks. Figure 4 gives the Average Overall Loss ($\mathscr{L}_{ol}$) after training five different task orders with a succession of five consecutive tasks. It reflects that after training all five tasks in different orders, the overall performance differs; for instance, Task order *FaGeFlLeHo* has minimum $\mathscr{L}_{ol}$ of 0.014 in comparison to Task order *HoFlLeFaGe* with maximum $\mathscr{L}_{ol}$ of 0.140. This validates the issue that the different task sequences converge the model parameters to different final values, affecting overall model performance extremely.

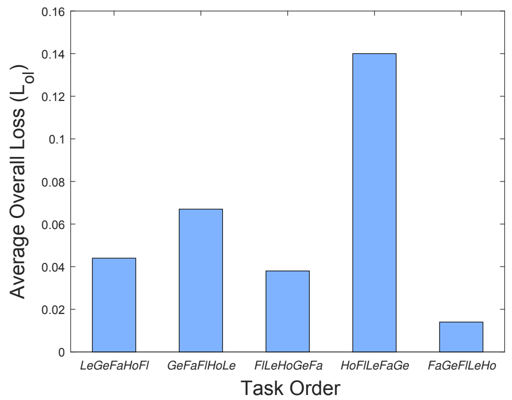

Fig. 4. Impact of task ordering on overall model performance (L_{ol})

B. Motivation to Similarity Analysis and Task Similarities

To further investigate the reason behind varying $\mathscr{L}_{ol}$ due to different task orders, analysis on the $\mathscr{L}_{ol}$ after training of each task was done. The results for this are formulated in

[1] https://github.com/tmdt-buw/continual-learning-mas-cloning-injection-molding.

Table III using MAS-SH and MAS-MH schemes. In the Task order, *LeGeFaHoFl*, the $\mathscr{L}_{ol}$ increases abruptly from 0.202 to 14.018 after training Task 3 using the MAS-SH scheme. The identical behaviour can also be seen in other task orders, highlighted (in red) in Table III. This unanticipated increase in $\mathscr{L}_{ol}$ is more prevailing in MAS-SH; however, MAS-MH suppresses these behaviours but cannot completely remove them. Further analysis shows that this unforeseen increase results only at the arrival of Task *Fa* in any task order.

TABLE III
AVERAGE OVERALL LOSS AFTER TRAINING OF EACH TASK IN A GIVEN TASK SEQUENCES

Scheme	Task order	Task 1	2	3	4	5
MAS-SH	*LeGeFaHoFl*	0.001	0.202	14.018	4.069	3.590
	GeFaFlHoLe	0.000	11.656	5.823	4.0136	3.518
	FlLeHoGeFa	0.001	0.003	0.035	0.327	15.726
	HoFlLeFaGe	0.002	0.041	0.022	12.817	4.854
	FaGeFlLeHo	0.026	11.424	6.003	4.299	3.430
MAS-MH	*LeGeFaHoFl*	0.001	0.006	0.068	0.052	0.044
	GeFaFlHoLe	0.000	0.175	0.119	0.092	0.067
	FlLeHoGeFa	0.001	0.003	0.007	0.0054	0.038
	HoFlLeFaGe	0.002	0.011	0.010	0.164	0.140
	FaGeFlLeHo	0.024	0.017	0.014	0.012	0.014

This suggests moving towards the similarity analysis of the tasks to investigate this behaviour further. The similarity analysis using Euclidean distance and Cosine similarity is done on each task's mean training data vector, and the similarity values corresponding to all possible task pairs are shown in Table IV. Lower Euclidean distance and higher Cosine value (near one) imply similar tasks and vice versa. Clearly, Task *Le* and *Fl* have a minimum Euclidean distance of 21.92 and a maximum Cosine similarity of 0.9988, suggesting that Task *Le* and *Fl* are highly similar. In contrast, Task *Fa* has the highest Euclidean distance from *Ge* of 4249.08 and a minimum Cosine similarity of 0.1213, suggesting they are highly dissimilar. This can further be seen from Task order *HoFlLeFaGe* in Table III that tasks with higher similarities (highlighted in green) do not increase $\mathscr{L}_{ol}$ extensively over sequential training.

Additionally, it can be seen that Task *Fa* has a minimum similarity value of 0.4576, 0.4478, 0.4356, and 0.1213 with *Ho*, *Le*, *Fl*, and *Ge*, respectively, and therefore shows a sudden surge in $\mathscr{L}_{ol}$ at its arrival in a sequence (highlighted in red in Table III). This suggests it is a highly dissimilar task and influences the overall performance of a task sequence.

C. Improved Performance

The proposed scheme signifies similarity analysis to be performed among tasks before sequentially training various tasks to a single DL model to reduce the impact of the task order. The similarity threshold (γ) considered for the given use case is 0.9, and tasks having a higher similarity value than γ with any other task are trained sequentially; otherwise, a separate DL model is prepared for them. Based on the proposed scheme, the $\mathscr{L}_{ol}$ for the five task orders is given in Fig. 5, wherein for given sequences Task *Fa* is trained separately.

TABLE IV
SIMILARITY ANALYSIS AMONG TASKS BASED ON EUCLIDEAN AND COSINE SIMILARITY MEASURE

Task P vs Q	Euclidean Distance	Cosine Similarity
Le vs *Fl*	21.920	0.9988
Le vs *Ho*	25.871	0.9984
Fl vs *Ho*	47.717	0.9949
Fl vs *Ge*	170.731	0.9323
Le vs *Ge*	175.073	0.9292
Ho vs *Ge*	182.842	0.9236
Ho vs *Fa*	4083.833	0.4576
Le vs *Fa*	4089.806	0.4478
Fl vs *Fa*	4096.376	0.4356
Ge vs *Fa*	4249.084	0.1213

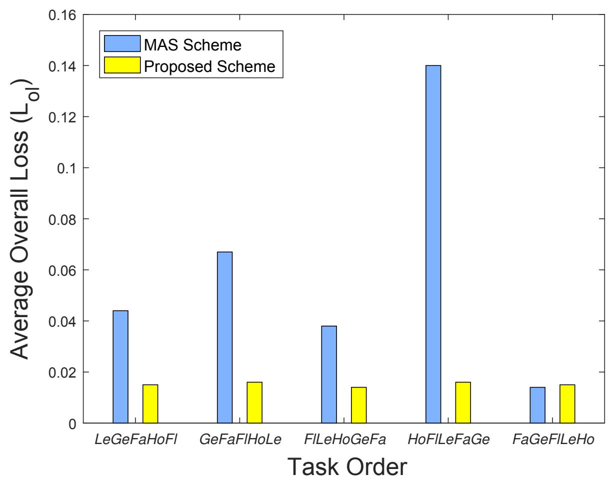

Fig. 5. Reduced impact of task ordering and performance improvement using proposed scheme

It is evident from Fig. 5 that from the proposed training mechanism, the $\mathscr{L}_{ol}$ almost remains constant for all the task orders. In contrast, the impact of task ordering is high in the MAS scheme. The overall impact of the tasks ordering is reduced by 70% using proposed scheme. Therefore, the proposed scheme ensures better quality inspection of all products by sharing a single DL model for similar tasks and a separate DL model for dissimilar tasks using light-weight similarity module.

VII. CONCLUSION

Smart manufacturing is embracing flexibility in manufacturing processes and efficient quality inspection using DL solutions on edge devices. CL caters the solution for training a DL model sequentially over multiple tasks. However, overall performance suffers greatly due to the task ordering. To this extent, MAS-MH suppresses this impact of the task order to a certain extent, but both MAS-SH and MAS-MH have its influence. The proposed lightweight similarity-based mechanism reveals that certain tasks in the continual training sequences are highly dissimilar to others. Therefore, while maintaining the tradeoff between model sustainability and performance, continual tasks are trained with similarities above a threshold (γ). Additionally, the similarity analysis module reduced the impact of task order on the overall performance of the DL model while ensuring the effective *PQI* of changing manufacturing processes. In the future, the applicability of the proposed scheme can be validated in other application scenarios, and the overhead in finding similarities between new and each previous tasks can be reduced.

REFERENCES

[1] V. Rutgers and B. Sniderman, “Around the physical-digital-physical loop,” tech. rep.

[2] G. Nain, K. K. Pattanaik, and G. K. Sharma, “Towards edge computing in intelligent manufacturing: Past, present and future,” *Journal of Manufacturing Systems*, vol. 62, pp. 588–611, 2022.

[3] H. Tercan, P. Deibert, and T. Meisen, “Continual learning of neural networks for quality prediction in production using memory aware synapses and weight transfer,” *Journal of Intelligent Manufacturing*, vol. 33, no. 1, pp. 283–292, 2022.

[4] Y. Liu, L. Guo, H. Gao, Z. You, Y. Ye, and B. Zhang, “Machine vision based condition monitoring and fault diagnosis of machine tools using information from machined surface texture: A review,” *Mechanical Systems and Signal Processing*, vol. 164, p. 108068, 2022.

[5] H. Gauttam, K. K. Pattanaik, S. Bhadauria, and G. Nain, “Bayesian Network based Reliability Analysis in Edge Computing enabled Machine Vision System,” in *2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, pp. 1–6, 2022.

[6] T. Zhang, B. Ding, X. Zhao, G. Liu, and Z. Pang, “LearningADD: Machine learning based acoustic defect detection in factory automation,” *Journal of Manufacturing Systems*, vol. 60, pp. 48–58, 2021.

[7] M. D. T. McDonnell, D. Arnaldo, E. Pelletier, J. A. Grant-Jacob, M. Praeger, D. Karnakis, R. W. Eason, and B. Mills, “Machine learning for multi-dimensional optimisation and predictive visualisation of laser machining,” *Journal of Intelligent Manufacturing*, vol. 32, no. 5, pp. 1471–1483, 2021.

[8] X. Li, X. Jia, Q. Yang, and J. Lee, “Quality analysis in metal additive manufacturing with deep learning,” *Journal of Intelligent Manufacturing*, vol. 31, no. 8, pp. 2003–2017, 2020.

[9] X. Zheng, S. Zheng, Y. Kong, and J. Chen, “Recent advances in surface defect inspection of industrial products using deep learning techniques,” *The International Journal of Advanced Manufacturing Technology*, vol. 113, no. 1, pp. 35–58, 2021.

[10] R. Ozdemir and M. Koc, “A Quality Control Application on a Smart Factory Prototype Using Deep Learning Methods,” in *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 1, pp. 46–49, 2019.

[11] S. Bharti and A. Mcgibney, “Privacy-Aware Resource Sharing in Cross-Device Federated Model Training for Collaborative Predictive Maintenance,” *IEEE Access*, vol. 9, pp. 120367–120379, 2021.

[12] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, “Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges,” *Information Fusion*, vol. 58, pp. 52–68, 2020.

[13] M. Dehghan, Z. Zhang, M. Siam, J. Jin, L. Petrich, and M. Jagersand, “Online Object and Task Learning via Human Robot Interaction,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2132–2138, 2019.

[14] B. Maschler, H. Vietz, N. Jazdi, and M. Weyrich, “Continual Learning of Fault Prediction for Turbofan Engines using Deep Learning with Elastic Weight Consolidation,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 959–966, 2020.

[15] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A Continual Learning Survey: Defying Forgetting in Classification Tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.

[16] J. Spaulding, “How smart manufacturers bring Industry 4.0 principles to quality,” 2018.

[17] “The Importance of Quality Control in Manufacturing,” 2021.

[18] H. Gauttam, K. K. Pattanaik, S. Bhadauria, G. Nain, and P. B. Prakash, “An efficient DNN splitting scheme for edge-AI enabled smart manufacturing,” *Journal of Industrial Information Integration*, p. 100481, 2023.

[19] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory Aware Synapses: Learning What (not) to Forget,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11207 LNCS, pp. 144–161, 2018.

[20] M. Harmouch, “17 types of similarity and dissimilarity measures used in data science. — by Mahmoud Harmouch — Towards Data Science,” tech. rep., 2021.

[21] S. Ebnesajjad, “10 - Injection Molding,” in *Fluoroplastics (Second Edition)* (S. Ebnesajjad, ed.), pp. 236–281, Oxford: William Andrew Publishing, second edi ed., 2015.