Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-025-02694-2

DeepMachining: online prediction of machining errors of lathe machines

Xiang-Li Lu[1] · Hwai-Jung Hsu[1] · Che-Wei Chou[2] · H. T. Kung[3] · Sheng-Mao Cheng[4]

Received: 4 May 2025 / Accepted: 17 September 2025

Abstract
Tool condition monitoring (TCM), powered by sensor technology and artificial intelligence (AI), has been adopted in the machining industry but faces issues such as data quality and model generalization. A classical transfer learning approach, where a pre-trained model trained on a large labeled dataset is fine-tuned to the target task, can mitigate model generality challenges. However, collecting abnormal data that represents faulty machining states is prohibitively expensive, making it difficult to gather sufficient, high-quality training data. Moreover, the limited computational resources on CNC machines complicate AI deployment. To address those problems, we develop DeepMachining, a deep learning-based AI system for real-time error prediction in lathe machine operations. We built and evaluated DeepMachining using real manufacturing data in practice. Specifically, we first pretrain a deep learning model to learn the representation of machining states. Then, we fine-tune it for specific machining tasks. The validation results show that DeepMachining provides high prediction accuracy for diverse workpieces and cutting tools. To the best of our knowledge, this work is one of the first industrial demonstrations of pre-trained deep learning models for predicting lathe machining errors.

Keywords Deep learning · Pre-trained model · Fine-tuning · Online prediction · Computer numerical control (CNC) machine

Introduction

The structures of modern manufacturing devices are increasingly complex, while tolerance requirements for possible machining errors become more strict. High-quality machining with low errors is essential in the manufacturing of high-precision parts.

For lathe machines, popular in the manufacturing of precision parts, various machining errors such as geometric tooling, thermal-induced, and load-induced errors (Hamdan et al., 2012; Mekid & Ogedengbe, 2010), etc., can lead to inaccuracies above the tolerance level of manufactured workpieces, resulting in monetary losses to the manufacturers. Early detection of manufacturing quality degradation and process anomalies (Chien & Chen, 2020; Ramezani et al., 2023), and assessment of the wear of cutting tools in material removal processes (Benkedjouh et al., 2015) can help reduce such risks. In particular, implementing real-time monitoring and online machining quality prediction can enhance error detection's efficiency and efficacy.

In recent years, tool condition monitoring (TCM), enabled by sensor technology and artificial intelligence

Hwai-Jung Hsu and Che-Wei Chou have contributed equally to this work.

✉ Hwai-Jung Hsu
hjhsu@mail.fcu.edu.tw

✉ Che-Wei Chou
cwchou@fcu.edu.tw

[1] Department of Information Engineering and Computer Science, Feng Chia University, No. 100, Wenhua Rd., Taichung 40724, Taiwan, ROC

[2] Department of Industrial Engineering and Systems Management, Feng Chia University, No. 100, Wenhua Rd., Taichung 40724, Taiwan, ROC

[3] Department of Computer Science and Electrical Engineering, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA

[4] Victor Taichung, No. 1, Jingke Central 2nd Rd., Taichung 408, Taiwan, ROC

Published online: 13 October 2025

(AI), has been employed to address these needs (Gavahian & Mechefske, 2023). For example, TCM has been widely used for fault detection and diagnosis (FDD) (Ding et al., 2022; Fernandes et al., 2022; Lei et al., 2020; Ntemi et al., 2022), predictive maintenance (PdM) (Schwendemann et al., 2021; Serradilla et al., 2022; Soori et al., 2023; Zhang et al., 2019), prognostics and health management (PHM) (Kumar et al., 2023; Nasir & Sassani, 2021; Ramezani et al., 2023), etc. in the manufacturing industry.

Deep-learning-based AI driven by manufacturing data is a promising approach for error detection, given that these data-driven methods have been successful in fields like computer vision and natural language processing (Ding et al., 2022; Nasir & Sassani, 2021; Ntemi et al., 2022; Ramezani et al., 2023; Serin et al., 2020; Soori et al., 2023). However, applying deep learning techniques to manufacturing brings new challenges, such as data preprocessing and model generalization for factory environments. For example, real-world machining processes involve a variety of workpiece materials, cutting tools, process recipes, and equipment models. As a result, supervised deep-learning models trained on signals from sensors of specific CNC machines may not apply to other machines. In other words, AI-powered solutions may not generalize to diverse manufacturing environments (Lee & Chien, 2022).

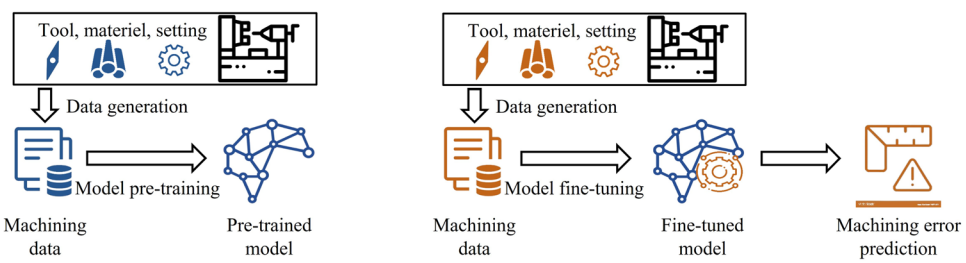
We may apply the classical transfer learning approach (Marei & Li, 2022; Marei et al., 2021; Sun et al., 2019) to address the model generality issue, where a pre-trained model trained on a large labeled dataset is fine-tuned to the target task. However, acquiring abnormal data corresponding to machining states that lead to the manufacture of erroneous workpieces is extremely costly in the machinery industry (Yang et al., 2024), especially concerning different materials, tools, and the variety of manufacturing settings. Furthermore, the acquisition of labeled data from specific manufacturing conditions may not guarantee the successful classification of unlabeled and imbalanced data, primarily due to the complexities of manufacturing environments and inherent discrepancies in data distribution (Chen et al., 2025; Pan & Yang, 2009; Ross et al., 2024). Thus, gathering sufficient high-quality data for pretraining models is challenging. Additionally, the limited computational resources of CNC machines necessitate addressing deployability concerns. Therefore, applying AI in complex manufacturing environments requires an adaptive learning approach with generality.

To address these challenges, this paper develops DeepMachining, a deep learning-based AI system, to predict machining errors utilizing the pre-trained model. As Fig. 1 shows, the pre-trained model was trained over the lifetime of the cutting tool until it was completely worn out. For model generalization, we perform model pretraining involving multiple spindle speeds. For fine-tuning, we propose a method similar to BitFit Ben Zaken et al. (2022), which adjusts the model's biases. This allows the pre-trained model to adapt to the target tasks using few-shot learning (typically two-shot). In other words, fine-tuning uses data collected from two instances of the target machining task. Merely 6.5% of the total parameters of the model are fine-tuned in less than 12.5% of epochs of the model pre-training. Thus, the proposed fine-tuning method not only suits existing machining processes but can also be completed with the limited computational power of the industrial computers in the CNC machines. Furthermore, to facilitate deployment on edge devices within CNC machines, we also investigate how the low sampling rate impacts the proposed approach in practice. To evaluate the proposed approach in predicting machining errors under various manufacturing settings, we use four machining tasks for validation.

The main contributions of this paper are:

- The proposed DeepMachining approach, and performed validation showing that, under the approach, we can pre-train a model that can be adapted to various downstream tasks.
- A few-shot model fine-tuning method (typically, two-shot) for adaptation to new manufacturing settings.
- The useful insight that the fine-tuning required in these manufacturing tasks is basically shifts of model's biases.
- An end-to-end factory demonstration of DeepMachining based in real-world factories.

The rest of this paper is organized as follows. Section "Related work" reviews the related literature on TCM



(a) Pre-trained Phase (b) Fine-tuning Phase

Fig. 1 The overall process of DeepMachining

and its applications. Section "Methodology" addresses the DeepMachine framework for online prediction of machining errors. Section "Experiments" details the experiments and analysis using real world machining tasks in factories. Section "Discussion" discusses the limitations and lessons learned in this study. Conclusions are drawn in Section "Conclusion".

Related work

Machining error, surface roughness, and tool wear are key quality control metrics in machining processes. Intelligent sensors, including accelerometers, data acquisition encoders, acoustic emission sensors, microphones, dynamometers, and image sensors, are utilized to monitor and diagnose machine health degradation and process anomalies (Jiang et al., 2021). Accelerometer sensors are sensitive and reliable in measuring workpiece dimensions with high precision (Duro et al., 2016; Lee et al., 2006). Therefore, we adapted accelerometers, a data acquisition (DAQ) encoder, and a microphone to collect manufacturing data for building DeepMachining to predict machining errors in this study.

Traditional machine learning (ML) approaches have been used to predict product quality during CNC machining tasks. Du et al. (2021) proposed a power spectral density based feature extraction method from spindle vibration and cutting force signals, which accurately predicted product roughness, profile, and roundness using tree-based regressor approaches in hard turning processes. Denkena et al. (2019) optimized workpiece quality and tool life in cylindrical turning processes by identifying the machined material based on machine learning algorithms. Papananias et al. (2020) proposed principal component analysis (PCA) based multilayer perceptron (MLP) networks to accurately predict the true position and circularity requirements of a workpiece in an experimental setting. Ura and Ghosh (2021) proposed the delay domain-based signal analysis approach to capture the dynamics of the underlying phenomenon in CNC machines. Furthermore, Ghosh et al. (2021) also addressed a sensor signal-based digital twin framework for intelligent machine tools. ML approaches could predict well on the collected datasets in manufacturing scenarios. However, ML approaches cannot be used as the kernel for pre-trained models to adapt to various downstream tasks via fine-tuning.

The advent of deep learning has reformed predictive approaches, enabling end-to-end prediction and diagnosis procedures to enhance CNC machining precision and reliability within smart tool condition monitoring systems (Lei et al., 2020; Zhao et al., 2019). Huang and Lee (2021) proposed one-dimensional convolutional neural network (1D-CNN) and sensor fusion approach accurately estimated tool wear and surface roughness for the CNC machining. Hesser and Markert (2019) demonstrated the feasibility of predicting CNC machine status and tool wear for maintenance plan using artificial neural networks. Proteau et al. (2023) proposed a variational autoencoder (VAE) regression model to predict the geometrical and dimensional tolerances of workpieces using sensor data in industrial settings. Zhu et al. (2020) established a long short-term memory (LSTM) model for one-dimensional time series and CNN for two-dimensional images. However, most machines operate normally in practice, and abnormal situations and fault events occur rarely. This kind of data imbalance tends to worsen data generalization capabilities, ultimately impeding the effectiveness and reliability of data-driven prediction methods (Chen et al., 2025; Yang et al., 2024; Yu et al., 2025).

Transformer-based networks have been applied to capture association relationships and dependency from vibration signals through the self-attention mechanism for improving performances of the developing models in recent years, (Bhandari et al., 2023; Li et al., 2024; Liu et al., 2020; Li et al., 2022; Wu et al., 2023). Wu et al. (2023) and Li et al. (2024) studied fault detection and classification in a rotary system with transformer-based models. Li et al. (2022) and Liu et al. (2020) applied for tool wear prediction in TCM topics. Compared to transformer-based approaches, in this study we utilize 1D-CNN networks with an attention mechanism to address the time series data of vibration signals, considering latency and computing power for prompt inference in practice.

Transfer learning (TL), which learns two types of networks to extract representations, solves cross-domain diagnosis problems with small and imbalanced data (Lei et al., 2020; Pan & Yang, 2009; Weiss et al., 2016; Zhang et al., 2022). Wang and Gao (2020) proposed a CNN-based transfer learning technique using vibration analysis for rolling bearing fault diagnosis. Specifically, adapting a pre-trained VGG19 network (Russakovsky et al., 2015), using non-manufacturing images from ImageNet (Deng et al., 2009) (i.e., model transfer) and transferring the adapted network structure to different fault severity levels and bearing types (i.e., feature transfer). Guo et al. (2019) proposed a deep convolutional transfer learning network to classify bearing health conditions with unlabeled data. Bahador et al. (2022) investigated a transfer learning approach for classifying tool wear based on tool vibration in hard turning processes. Ross et al. (2024) proposed a transfer learning model with Inception-V3 network (Szegedy et al., 2015) to detect tool flank wear under distinct cutting environments.

However, the research gap between practitioners and researchers remains in practice (Lee & Chien, 2022). Different processing parameters result in different data

distributions, which pose a significant challenge to ML and DL models. Collecting and labeling data with different combinations of materials, tools, process recipes, and machines in practice is difficult and expensive (Yu et al., 2025). Furthermore, even if labeled data is obtained from some manufacturing conditions, the resulting predictive models may fail to classify unlabeled and unbalanced data due to intricate manufacturing settings and data distribution discrepancies (Chen et al., 2025; Pan & Yang, 2009; Ross et al., 2024).

This study focuses on real-time monitoring and online machining quality prediction for CNC machining, specifically utilizing sensor data from manufacturing settings, such as cutting vibrations, current, and rotation, to predict machining errors for quality control. Therefore, how to design pre-trained models that are easily adaptable to empirical field applications with strong performance is an important topic (Brown et al., 2020; Devlin et al., 2018; Wolf et al., 2020). Furthermore, fine-tuning on task-specific supervised data enables seamless adaptation to various specific tasks in practical settings (Ben Zaken et al., 2022; Cai et al., 2020; He et al., 2021; Hu et al., 2021).

Methodology

Problem definition

The machining error is the difference between the dimension measured after the machining of a workpiece and the target dimension described in the specification. The proposed DeepMachining estimates machining errors under various processing conditions, e.g., different combinations of machining tools and configurations, on CNC lathes without actual measurement. Several factors can impact the machining error of a workpiece. These include the wear condition of cutting tools, the hardness and processing difficulty of the material, the environmental temperature (thermal expansion), and the wear of machine components on the equipment (i.e., the lathe). In order to perceive the factors, accelerometers are installed to collect the vibration signals that occur during the machining process; the machine status, such as the spindle speed and motor current, etc., during the machining process is also recorded. Besides, it's important to note that any specific section of a workpiece can be machined multiple times. In other words, multiple cutting processes may be performed at the same place on a workpiece to achieve the target size. The signals and data generated from multiple machining sessions should be gathered and processed.

(a) Internal structure of CNC lathe machine. (b) Accelerometer placement on the spindle. (c) Accelerometer placement on the turret.

Fig. 2 The experimental environment of the CNC lathe machine

Machine settings

This study conducted experiments on a horizontal CNC lathe machine, which features an internal spindle and three-axis linear guides, as shown in Fig. 2. Piezoelectric accelerometers are deployed at three distinct positions: First, behind the spindle, as depicted in Fig. 2b; Second, in front of the spindle, also shown in Fig. 2b; and Third, at the base of the tool turret, illustrated in Fig. 2c, to collect relevant vibration signals. The machine controller records and outputs the spindle speed and current of the drive motors for the spindle and the turret during the machining process, which serves as the machine status.

Input formulation

In order to predict the machining error $y \in \mathbb{R}^1$, two inputs $\mathcal{X} = \{X_1, X_2, \ldots, X_n\} \in \mathbb{R}^{N \times SR \times C_1}$, including the vibration signals and machine status during the machining process, and $\tilde{\mathcal{X}} = \{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n\} \in \mathbb{R}^{N \times (\frac{SR}{2}+1) \times C_2}$, the transformation of vibration signals in $\mathcal{X}$ from time domain to frequency domain using Fourier Transform (Brigham, 1988), are used. The duration of each input $X_i \in \mathcal{X}$ is one second around the location of the workpiece where the machining error y is measured. N is the number of cuts, C_1 and C_2 are the number of input channels, and SR indicates the sampling rate of the sensors used in input collection.

The core of DeepMachining

The core of DeepMachining is a two-stage model handling multiple cuttings across machining processes to estimate the machining error $\hat{y}$ of a workpiece, as illustrated in Fig. 3.

Stage I: Time/Frequency Domain Signals Encoding. For each cut of the machining procedures, DeepMachining accepts a pair of input signals, X_n and $\tilde{X}_n$, from both the time and frequency domains. These signals are then processed by two parallel Signal Encoders that share the same architecture but maintain independent weights to accommodate the distinct characteristics of each input domain. Across cuts, the same Signal Encoder (per domain) is reused, enabling consistent feature extraction with shared parameters and reduced model complexity. For each cut, the

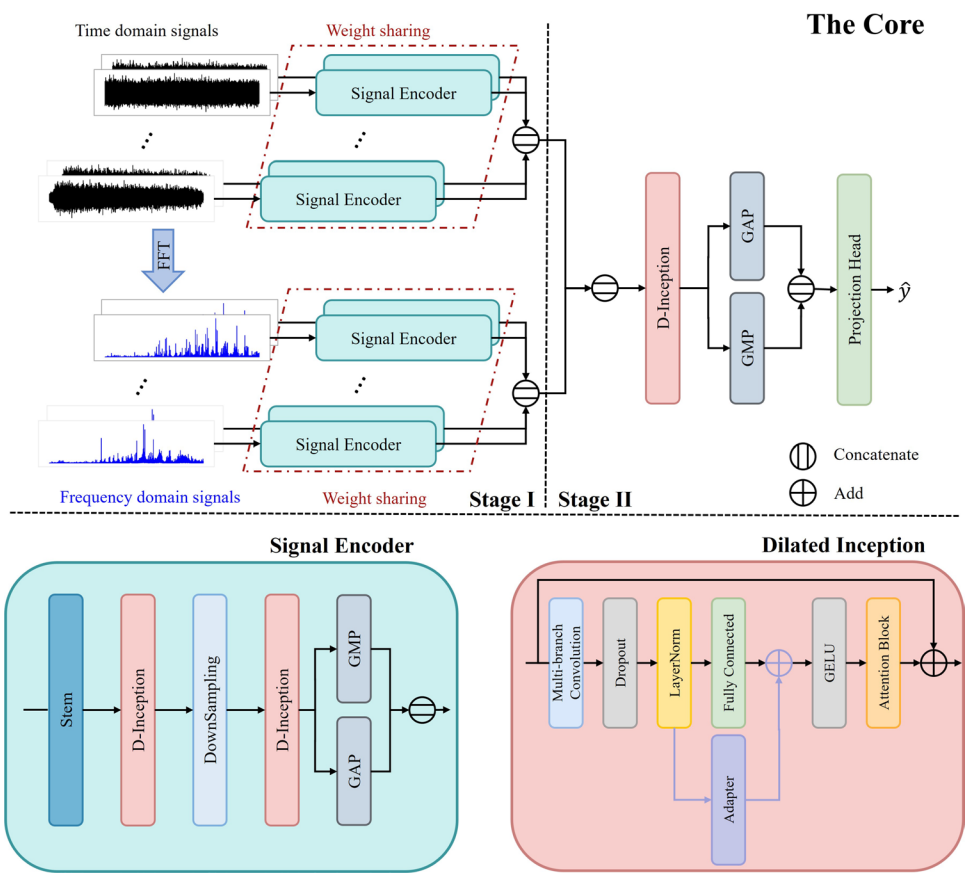


Fig. 3 Core architecture of DeepMachining, featuring parallel time-frequency signal encoders, D-Inception modules with multi-branch dilated convolutions and transformer-inspired components (Layer Normalization, GELU), followed by lightweight channel-temporal attention blocks

features extracted from the time and frequency branches are concatenated to form a per-cut embedding H_n.

Stage II: Aggregation and Estimation. The per-cut embeddings $\mathcal{H} = \{H_1, \ldots, H_n\}$ are stacked to form a sequence, which is processed by a Dilated Inception (D-Inception) block to model inter-cut relationships. The resulting aggregated representation is then fed into a Projection Head, a single-layer feed-forward network, to estimate the machining error $\hat{y}$.

Detailed descriptions of the Signal Encoder and D-Inception modules are given in the following subsections.

Stem

The Stem module serves as the first layer in each Signal Encoder. It is designed to reduce the temporal resolution and computational cost through a learnable downsampling operation. Specifically, it applies a 1D convolution with stride to shorten the input sequence while maintaining key structural information in the signal.

The Stem transformation is defined as:

$$\text{Stem}(X_n) = \delta(f^{11}(\text{LN}(\text{Dropout}(WX_n)))) \quad (1)$$

Here, $W \in \mathbb{R}^{C \times \frac{d}{4}}$ is a learnable matrix that maps the input into a higher-dimensional feature space, where $C \in \{C_1, C_2\}$ denotes the number of input channels and d represents the output channel dimension. f^{11} denotes a 1D convolution with kernel size 11, channel dimension d and stride 5. LN refers to Layer Normalization (Ba et al., 2016), and δ is the GELU activation function (Hendrycks & Gimpel, 2016).

Dilated inception (D-inception)

The D-Inception module is designed to extract multiscale temporal features with high computational efficiency. It extends the original Inception architecture (Szegedy et al., 2015) by incorporating dilated convolutions, attention mechanisms, and transformer-inspired enhancements; such as replacing Batch Normalization (Ioffe & Szegedy, 2015) with Layer Normalization (Ba et al., 2016); using GELU (Hendrycks & Gimpel, 2016) instead of ReLU Agarap (2018); and removing activation functions from bottleneck layers (Liu et al., 2022).

Multi-branch Convolution

The Multi-branch Convolution module inside D-Inception adopts a four-branch structure to capture features with varying receptive fields: (i) a 1D convolution with kernel

size 1 (CONV_1); (ii) a CONV_1 followed by a standard convolution with kernel size s (CONV_s); (iii) a CONV_1 followed by a dilated convolution with kernel size s and dilation rate 2 (DCONV_s); and (iv) a max-pooling layer of size s followed by CONV_1.

Transformer-inspired Components The outputs from these branches are concatenated along the channel axis and then sequentially processed by Dropout, Layer Normalization, and a GELU activation layer. This produces an intermediate representation $F \in \mathbb{R}^{L\times d}$, where L denotes the temporal length and d the feature dimension, which is subsequently fed into an Attention Block (discussed in the following paragraph) to selectively emphasize informative features.

Attention Block To emphasize informative features with limited parameter overhead, the Attention Block integrates a lightweight sequential attention mechanism inspired by Convolutional Block Atten tion Module (CBAM) (Woo et al., 2018), comprising channel attention followed by temporal attention. Given an input F, channel attention first computes an attention map $M_c(F)$, which is applied via element-wise multiplication to produce an intermediate tensor F'. This is followed by temporal attention, which generates $M_t(F')$ to refine F' into the final output F'':

$$F' = M_c(F) \otimes F, \quad F'' = M_t(F') \otimes F' \tag{2}$$

Channel Attention. The channel attention map $M_c(F) \in \mathbb{R}^{1\times d}$ is computed by applying global average pooling and global max pooling along the temporal dimension, concatenating the results, and passing them through a two-layer Multilayer Perceptron (MLP) with GELU and sigmoid activations:

$$M_c(F) = \sigma(W_1(\delta(W_0(\text{AvgPool}(F)||\text{MaxPool}(F))))) \tag{3}$$

Here, $W_0 \in \mathbb{R}^{\frac{d}{r}\times d}$ and $W_1 \in \mathbb{R}^{d\times\frac{d}{r}}$ are learnable projection matrices, δ denotes GELU, σ is sigmoid, and r is the reduction ratio. The attention weights are broadcast along the temporal dimension before multiplication.

Temporal Attention. Similarly, temporal attention derives $M_t(F') \in \mathbb{R}^{L\times 1}$ by performing average and max pooling across the channel axis, concatenating, and applying a 1D convolution:

$$M_t(F') = \sigma(f(\text{AvgPool}(F')||\text{MaxPool}(F'))) \tag{4}$$

where f denotes a 1D convolution. The resulting weights are broadcast along the channel axis to scale F'.

Finally, the output F'' is added to the original input feature via a residual connection (He et al., 2016), which promotes gradient stability.

Downsampling

The Downsampling module reduces the temporal resolution of feature sequences to improve computational efficiency and increase the receptive field, especially when stacked with D-Inception blocks.

Unlike the Stem module, which performs learnable downsampling through 1D convolutions, Downsampling employs non-learnable max pooling to reduce the sequence length without introducing additional temporal parameters. The pooled features are then transformed through a lightweight bottleneck projection using linear layers. The Downsampling is defined as:

$$\text{Downsampling}(F) = W_1(\text{LN}(W_0(\text{MaxPool}(F)))) \tag{5}$$

Here, MaxPool denotes temporal max pooling, and LN represents Layer Normalization (Ba et al., 2016). The learnable weights $W_0 \in \mathbb{R}^{\frac{d}{4}\times d}$ and $W_1 \in \mathbb{R}^{d\times\frac{d}{4}}$ constitute a bottleneck structure that first compresses the feature channels to reduce computational cost and then restores the original dimensionality for downstream processing. This structure facilitates efficient cross-channel interaction while maintaining a compact parameter footprint.

Fine-tuning method

To adapt the model to diverse machining tasks, we adopt a fine-tuning strategy inspired by TinyTL Cai et al. (2020). TinyTL introduces lightweight residual modules and updates only these modules, biases, and output head during fine-tuning.

In our framework, we employ *Adapter*, a two-layer bottleneck-style feed-forward network, designed to perform lightweight feature transformation during fine-tuning. Adapters are inserted into both the D-Inception and Downsampling blocks. During fine-tuning, only the Adapters, biases, and the projection head are updated, while all other pre-trained parameters remain frozen. This design enables rapid adaptation to new machining tasks with reduced memory and computational overhead, as only a small subset of parameters needs updating.

Compared to other parameter-efficient adaptation techniques, our approach exhibits several engineering advantages. For example, LoRA (Hu et al., 2021) constrains its low-rank updates to the query and key projections within transformer attention mechanisms, thereby inherently coupling it to transformer-based architectures. In contrast, our Adapter modules are integrated alongside the primary computation flows and are not restricted by architectural constraints, enabling straightforward incorporation into convolutional modules such as D-Inception

and Downsampling. Similarly, although AdapterFusion (Pfeiffer et al., 2021) exploits multiple adapters to facilitate transfer across tasks, it typically incurs additional memory overhead. By comparison, our design maintains a reduced parameter footprint and achieves low inference latency, which are essential for deployment on CNC edge computing platforms. The Adapter operation is formally defined as:

$$\text{Adapter}(F) = F + W_1(W_0(F)) \tag{6}$$

Here, $W_0 \in \mathbb{R}^{d \times \frac{d}{r}}$ and $W_1 \in \mathbb{R}^{\frac{d}{r} \times d}$ are learnable weight matrices, r is the reduction ratio, and $F \in \mathbb{R}^{L \times d}$ denotes the feature representation from either the D-Inception or Downsampling module.

Practical adaptation

In practice, workpiece dimensions are typically measured during CNC restarts, process resets, or when operators deem reconfiguration necessary. Based on these measurements, operators adjust cutting tools and machine parameters to ensure the accuracy of subsequent machining.

To accommodate such variations, including changes in machining conditions, workpiece geometry, and tool states, DeepMachining supports few-shot fine-tuning (typically two-shot) to adapt the pre-trained model on the fly.

DeepMachining maintains a compact architecture with approximately ***260,000*** parameters. Notably, only ***6.5%*** of the parameters need to be fine-tuned using just ***12.5%*** of the epochs required for pre-training, enabling efficient and lightweight adaptation across diverse machining configurations.

Deployment

Figure 4 illustrates the integrated hardware and software architecture used to deploy DeepMachining on a CNC machine. The system comprises an industrial PC (IPC), the machine controller, and sensors (accelerometers). The IPC communicates directly with both the sensors and the controller to handle data collection and adaptive control.

This deployment supports two primary workflows: *inference* and *fine-tuning*. In the inference workflow (shown on the left side of Fig. 4), during machine processing, the IPC continuously collects real-time data streams from the accelerometers and the controller. A signal segmentation module then utilizes the machining G-code, collected signals, and design drawings to extract relevant segmented signals. DeepMachining processes these segmented signals to estimate the machining error $\hat{y}$. Based on these estimates, the IPC automatically adjusts CNC parameters to perform tool calibration, allowing compensation adjustments without halting machining operations.

The fine-tuning workflow (right side of Fig. 4) illustrates how DeepMachining adapts to new machining contexts, such as changes in workpiece geometry, tooling, or process conditions. Similar to the inference pipeline, the IPC collects and segments the signals using G-code and design information. In this mode, however, manually or automatically labeled measurements (e.g., offline dimensional inspections) are used to update the model on-the-fly via few-shot fine-tuning. After fine-tuning, the updated model is validated, stored with version control, and deployed for subsequent inference, ensuring continuous improvement and consistent performance across diverse machining scenarios.

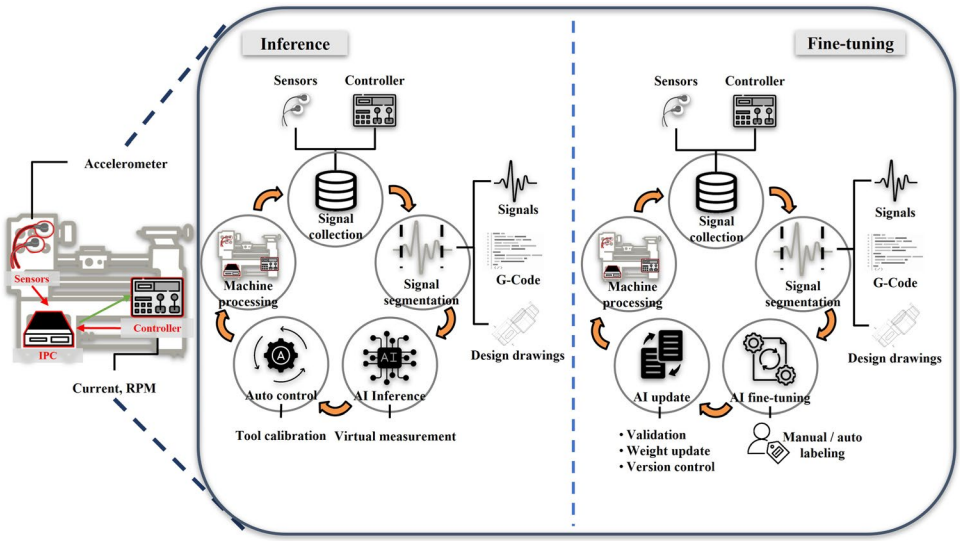


Fig. 4 Deployment of DeepMachining on a CNC system, illustrating real-time inference for tool calibration and on-site few-shot fine-tuning for model adaptation

Table 1 Machining configuration of datasets

Dataset	Spindle RPM	Feed Rate (mm/rev)	# of Configuration Changes
WC_AO-MS	1100 to 2700	[0.25, 0.1]	14
WC_TAN-MS	1600 to 2200	[0.25, 0.12]	2
WC_TC-AS	1000 to 2100	[0.12, 0.25]	3

Table 2 Train/test split of pre-trained datasets. The number of workpieces is shown in brackets

Dataset	Train	Test	Total
WC_AO-MS (Random)	277 [277]	70 [70]	347
WC_AO-MS (Sequential)	281 [277, 2, 2]	66 [11, 7, 48]	347

Table 3 Train/test split of adapted datasets. The number of workpieces is shown in brackets

Dataset	Train	Test	Total
WC_TAN-MS	4 [2, 2]	83 [37, 46]	87
WC_TC-AS	6 [2, 2, 2]	28 [5, 2, 19]	34

Experiments

See Tables 1, 2 and 3.

Settings

Datasets: The datasets were collected from three distinct outer diameter machining tasks, and were named on the basis of the material and coating of the cutting tool, as well as the material of the workpieces under machining. All of the cutting tools used in the experiments were made of Tungsten Carbide (WC). The coatings of the cutting tools include Aluminium Oxide (AO), Titanium Aluminium Nitride (TAN), and Titanium Carbonitride (TC). The materials of the workpieces included Medium-Carbon Steel (MS) and Alloy Steel (AS). On the other hand, except for the vibration and the machine controller signals, adjustments to the machining configurations (e.g. spindle speed, initial tool position) and context changes (e.g. machining dates) along with the machining processes were also recorded. Table 1 shows the summary of each dataset, and the details are described as follows:

- WC_AO-MS: 347 MS workpieces were machined using a tool made of WC and coated with AO. The workpieces were machined on seven different dates, with varying spindle speeds for each date. Besides, according to the judgment of on-site personnel, the cutting tool underwent eight position adjustments to offset its machining precision. Tool adjustments were required when machining precision declined beyond a threshold determined by the machining worker. Besides the first machining, the acts of machining on the other dates and the tool position adjustments are considered a machining configuration adjustment. This dataset was used for model pre-training.
  To evaluate the performance of the pre-trained model, the dataset was split into training and testing sets for assessment. The testing dataset was generated in two different ways. First, 80% of the data was randomly selected for training, and the remaining 20% for testing. Second, the first 80% of the dataset (sequenced by machining time) was used for training, and the remaining for testing. In the following sections, the first dataset is named WC_AO-MS (Random) and the second one as WC_AO-MS (Sequential).
- WC_TAN-MS: 87 MS workpieces were machined using a tool made of WC and coated with TAN. The workpieces were machined on two different dates, with varying spindle speeds for each date. Since we plan to fine-tune the pre-trained model to adapt the tool differences in WC_TAN-MS, each machining date in WC_TAN-MS is considered as machining configuration adjustment. To assess whether the pre-trained model can adapt to changes in cutting tools through fine-tuning, few-shot learning is applied for each machining date, i.e. machining configuration adjustments, as described in Section "Fine-tuning method". In other words, for each date, the first two workpieces are used for model fine-tuning, and the remaining ones are used for testing.
- WC_TC-AS: 34 AS workpieces were machined using a tool made of WC and coated with TC. All the workpieces were machined on the same date. However, there were three instances of machining configuration adjustments: (1) when the machine started in the morning, (2) when the machine resumed after the lunch break, and (3) when the cutting tool is adjusted for precision offset. The two workpieces processed after each machining configuration adjustment were used for model fine-tuning. Subsequent workpieces, processed until the next machining configuration adjustments or end of the machining, were used for testing. This allowed us to assess whether our fine-tuning approach could adapt to changes in both cutting tools and workpiece materials.**Evaluation Metrics:** The performance of our method is evaluated by Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Pearson Correlation (CORR). MAE and RMSE are both used to assess whether a model accurately estimates actual machining errors. RMSE is more sensitive to outliers compared to MAE, and MAE is considered more intuitive to the domain experts.

CORR is used to observe whether the model's estimated machining errors are correlated with the actual errors. In

certain machining processes that demand high precision, the variations of machining errors are small. A model can give machining error estimation within a narrow range of values to get small MAE and RMSE. However, in such circumstances, if the model is not truly capable of predicting the machining error, the CORR would be low. In other words, CORR assists us in distinguishing whether a model really learns the relationships between the signals during the machining processes and the machining errors. A low CORR accompanied by a high MAE or RMSE suggests that the model's estimation is biased.

Baselines: Three methods were chosen as the baseline methods for comparison.

- SVR: Support vector regression (SVR) is a kernel-based machine learning model for regression tasks (Cortes & Vapnik, 1995). SVR utilizes kernel functions to identify key data points influencing the regression hyperplane and efficiently map the input data into a high-dimensional feature space. In CNC machine applications, SVR has been applied to engineering optimization problems such as surface roughness improvement and cutting force reduction in milling (Yeganefar et al., 2019), as well as motor current control of machine tool drives (Schwenzer et al., 2020). In this study, referring to Sayyad et al. (2022), the statistical features of vibration, spindle speed, and motor current signals were processed as input to SVR with an RBF kernel for model training and machining error inference.
- 1D-CNN: The one-dimensional convolutional neural network (1D-CNN) is commonly employed for the analysis of time series data. In this study, we adopted the 1D-CNN method proposed by Huang and Lee (2021) as a representative baseline. In their approach, 1D-CNN was combined with a sensor fusion technique to accurately estimate tool wear and surface roughness in CNC machining. The vibration signals were utilized as inputs to the model for machining error estimation.
- 2D-CNN: Once a series of vibration or sound signals is transformed into spectrograms, a visual representation of the spectrum of frequencies as they vary with time, the resulting images can be analyzed using a two-dimensional convolutional neural network (2D-CNN). In this study, the approach proposed by Liao et al. (2021) was introduced as a representative 2D-CNN baseline. Liao et al. applied Short-Time Fourier Transform (STFT) (Brigham, 1988) to sound signals to obtain spectrograms, which were then used to predict specific machining configurations. They fine-tuned a VGG16-based model pre-trained on ImageNet (Deng et al., 2009) to accept spectrograms as input. We adopted their model framework but replaced the input signals with vibration signals processed into spectrograms using STFT, and fine-tuned the model for machining error estimation.

Table 4 DeepMachining architecture hyperparameter overview

Step	Module	Hyperparameters	Output Shape
1	Inputs	–	–
1.1	Inputs (time)	–	$N \times 10240 \times 8$
1.2	Inputs (freq)	–	$N \times 5121 \times 2$
2	Signal Encoder (time)	–	$N \times 192$
2.1	Stem	$s = 11, c = 96$	$N \times 2048 \times 96$
2.2	D-Inception	$s = 9, c = 96, r = 4$	$N \times 2048 \times 96$
2.3	Downsampling	$c = 96$	$N \times 1024 \times 96$
2.4	D-Inception	$s = 9, c = 96, r = 4$	$N \times 1024 \times 96$
2.5	GAP + GMP	–	$N \times 192$
3	Signal Encoder (freq)	Same as time branch	$N \times 192$
4	Concat (time + freq)	–	$N \times 384$
5	Aggregation	–	–
5.1	D-Inception	$s = 3, c = 96, r = 4$	$N \times 96$
5.2	GAP + GMP	–	$N \times 192$
6	Projection Head	–	1

Hyperparameters: The main hyperparameters of DeepMachining are summarized in Table 4. These include the kernel sizes s used in the multi-branch convolutions, the output channel dimensions c (i.e., feature channels), and the reduction ratio r in the bottleneck-style modules and attention mechanisms. Additionally, the overall model employs a dropout rate of 0.1 after convolutional concatenations.

Devices: We pre-trained the core of DeepMachining on a workstation equipped with an AMD Ryzen Threadripper 3990X processor (256 MB cache, 2.9 GHz), 256 GB RAM, and an NVIDIA Quadro RTX 8000 GPU (48 GB GDDR6 RAM) using TensorFlow. AdamW optimizer was employed for both pre-training and 2-shot tuning, with specific hyperparameters adjusted as follows: for pre-training, a learning rate of 0.001, a batch size of 512, and 512 epochs; for 2-shot tuning, a learning rate of 0.00001, a batch size of 32, and 64 epochs.

For 2-shot tuning, the core was executed on a host featuring an Intel Xeon Silver 4210 processor (13.75 MB cache, 2.2 GHz), 256 GB RAM, and an NVIDIA RTX 2080 Ti GPU (11 GB GDDR6 RAM) using TensorFlow. The 2-shot tuning required 2.5 min on the GPU and 35 min on the CPU, with respective inference times of 0.026 s and 0.036 s.

Additionally, we evaluated the practical runtime of DeepMachining on an industrial PC (IPC) equipped with an Intel Core i7-6700 processor (8 MB cache, 3.4 GHz) and 16 GB DDR4 RAM in a CNC lather machine. On this IPC, two-shot fine-tuning required approximately 25 min on the CPU, while the inference time per workpiece was around 0.1 s.

Evaluation and comparison

Initially, we validated the model's performance on the pre-trained dataset WC_AO-MS, as shown in Table 2. As demonstrated in Table 5, whether the testing set was generated randomly (WC_AO-MS (Random)) or sequentially (WC_AO-MS (Sequential)), our approach surpasses all baseline methods across various metrics. The bold font denotes the best performance achieved among the compared algorithms. SVR presents the weakest performance among all the methods, as highlighted by the highest MAE and RMSE, coupled with the lowest CORR. Notably, the CORR of SVR is close to 0.5 in WC_AO-MS (Random) but declines significantly to nearly 0 in WC_AO-MS (Sequential), indicating its limited robustness as machining progresses.

2D-CNN outperforms 1D-CNN in most metrics but exhibits a lower CORR for WC_AO-MS (Sequential). Moreover, compared to WC_AO-MS (Random), both 1D-CNN and 2D-CNN exhibit a substantial increase in estimation errors and a decrease in CORR on WC_AO-MS (Sequential).

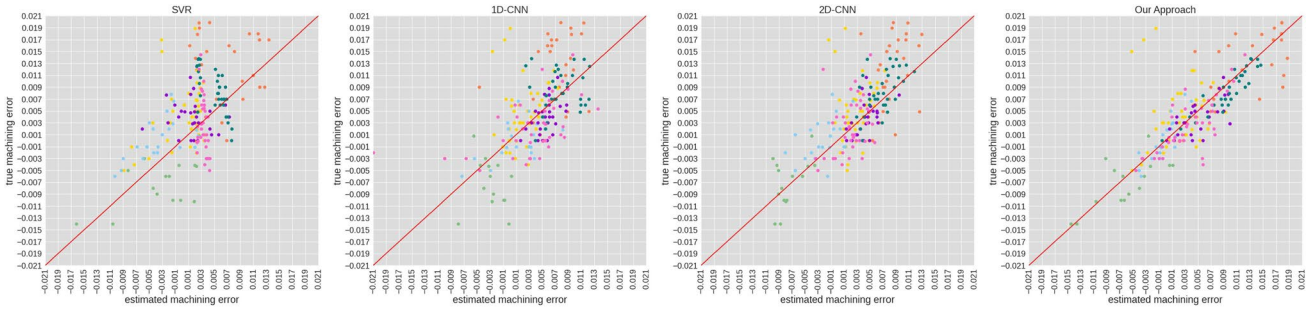
In reality, only the first few workpieces processed can be used for model fine-tuning. An approach that fails to perform well with sequential workpieces during production would not be practical for real-world applications. Compared to WC_AO-MS (Random), our approach exhibits only a slight uptick in estimation errors and a limited reduction in CORR in WC_AO-MS (Sequential). These results demonstrate that our approach can sustain robust predictive performance throughout continuous machining processes.

Table 5 Performance comparison on pre-trained dataset

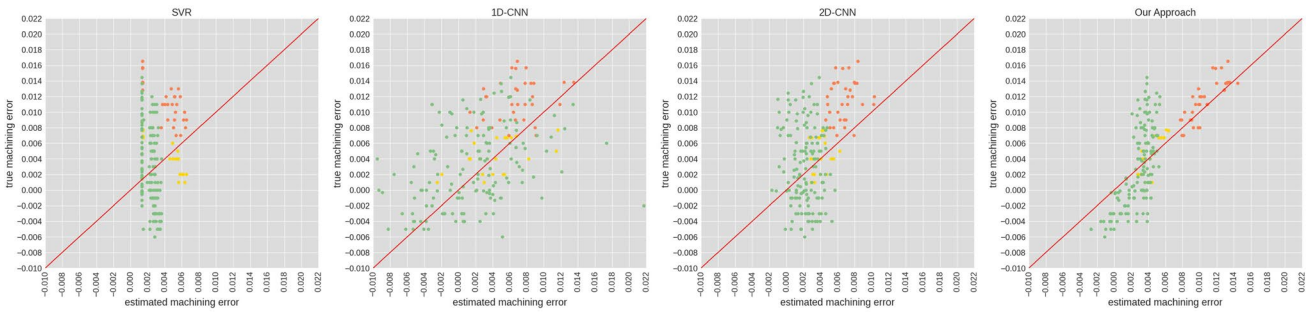
Dataset	Method	MAE	RMSE	CORR
WC_AO-MS (Random)	SVR	0.0049	0.0062	0.5052
	1D-CNN	0.0039	0.0053	0.5864
	2D-CNN	0.0036	0.0049	0.7353
	Our Approach	**0.0026**	**0.0040**	**0.8020**
WC_AO-MS (Sequential)	SVR	0.0050	0.0061	−0.0463
	1D-CNN	0.0045	0.0057	0.4722
	2D-CNN	0.0043	0.0052	0.4029
	Our Approach	**0.0028**	**0.0036**	**0.7754**

Figure 5 illustrates the relationships between the estimated machining errors produced by each method (x-axis) and the corresponding actual machining errors of the testing set (y-axis). Figure 5a shows the results for WC_AO-MS (Random), while Fig. 5b presents those for WC_AO-MS (Sequential). The axes scales are consistent across the plots for the same dataset. The red diagonal line indicates a perfect agreement between the estimated and actual values.

Figure 5b indicates that SVR only estimates machining errors within a limited range of 0.003 to 0.007 mm and shows virtually no correlation between the estimated and actual values on WC_AO-MS (Sequential). Both 1D-CNN



(a) WC_AO-MS (Random)



(b) WC_AO-MS (Sequential)

Fig. 5 Scatter plots of the actual machining errors versus the estimated errors by different methods on the pre-trained dataets. Different colors represent different machining configurations

Table 6 Performance comparison of adapted dataset

Dataset	Method	MAE	RMSE	CORR
WC_TAN-MS	SVR	0.0056	0.0064	0.0319
	1D-CNN	0.0027	0.0034	0.4058
	2D-CNN	0.0016	0.0022	0.7240
	Our Approach	**0.0013**	**0.0016**	**0.8838**
WC_TC-AS	SVR	0.2709	0.3247	0.0339
	1D-CNN	0.0041	0.0052	0.2518
	2D-CNN	0.0029	0.0037	0.6010
	Our Approach	**0.0024**	**0.0032**	**0.7599**

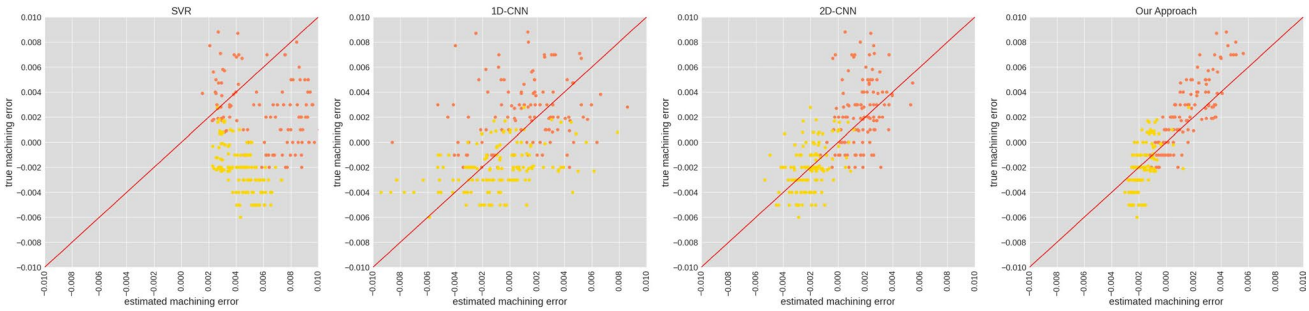
and 2D-CNN provide estimates with moderate correlation to the actual values in both datasets. However, as observed in Fig. 5a, the estimates produced by 1D-CNN exhibit higher variance, as evidenced by the wider spread of points around the diagonal line. In contrast, in Fig. 5b, the estimates made by 2D-CNN appear constrained by an invisible lower bound around -0.001 mm, despite the actual machining errors reaching as low as -0.006 mm in WC_AO-MS (Sequential). As illustrated in Fig. 5, compared to 1D-CNN and 2D-CNN, our approach exhibits smaller estimation errors and stronger correlation for both the WC_AO-MS (Random) and WC_AO-MS (Sequential) datasets. Additionally, the points are distributed more compactly along the diagonal line, indicating better estimation accuracy.

As summarized in Table 3, WC_TAN-MS is designed to evaluate the adaptability of models in machining scenarios involving identical workpiece materials but different cutting tools, while WC_TC-AN is used to assess adaptability when both workpiece and tool materials differ. As shown in Table 6, the correlation coefficient (CORR) of SVR is close to zero, revealing that SVR fails to adapt to changes in machining conditions (regardless of differences in workpieces or tools) for machining error prediction. On the other hand, 2D-CNN outperforms 1D-CNN on both datasets. Notably, our approach surpasses all other methods, achieving the best performance across all metrics.

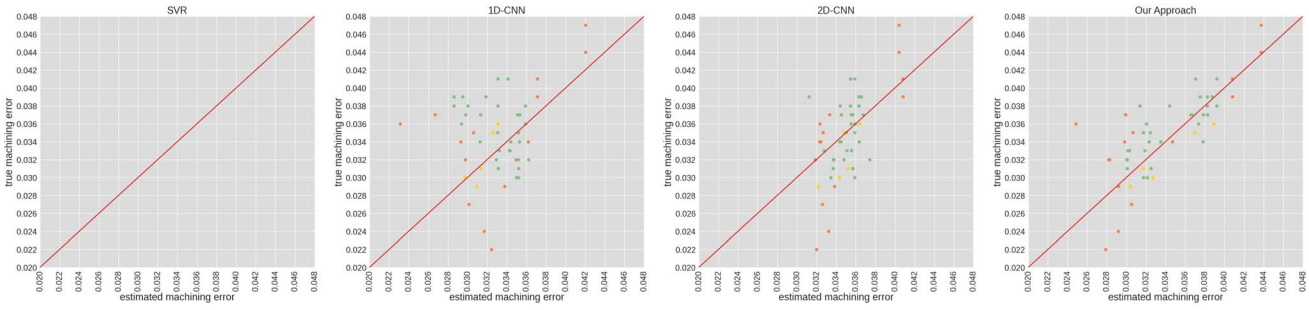
As shown in Fig. 6, the estimations made by SVR exhibit no correlation with the actual values and, in the case of WC_TC-AN, even fall outside the plotting range due to large errors. For the adapted datasets, both 1D-CNN and 2D-CNN perform better than SVR but still worse than our approach, and both methods appear constrained by a lower bound around 0.03 in their estimations on WC_TC-AN. These results demonstrate that our approach achieves strong adaptability and generalization across datasets involving variations in both workpiece and tool materials.

Ablation study

To investigate the contributions of key architectural components in the core of DeepMachining, we conducted an ablation study using the adapted datasets WC_TAN-MS and WC_TC-AS. Specifically, we examined the following variants:



(a) WC_TAN-MS



(b) WC_TC-AS

Fig. 6 Scatter plots of the actual machining errors versus the estimated errors by different methods on the adapted datasets. Different colors represent different machining configurations

Table 7 Ablation study of model architecture on adapted dataset

Dataset	Method	MAE	RMSE	CORR
WC_TAN-MS	w/o Transformer-inspired components	0.0075	0.0102	0.7689
	w/o adapter	0.0034	0.0052	0.3892
	w/o frequency inputs	0.0027	0.0044	0.3947
	w/o dilation convolution	0.0021	0.0028	0.5202
	Our Approach	**0.0013**	**0.0016**	**0.8838**
WC_TC-AS	w/o Transformer-inspired components	0.0033	0.0042	0.6367
	w/o adapter	0.0030	0.0038	0.7321
	w/o frequency inputs	0.0026	0.0035	0.7527
	w/o dilation convolution	0.0028	0.0038	0.7151
	Our Approach	**0.0024**	**0.0032**	**0.7599**

- w/o Transformer-inspired components: This variant removed Layer Normalization and Dropout, and replaced GELU activation with ReLU.
- w/o adapter: The Adapter modules used during fine-tuning were omitted, so that only the final projection head and biases were updated.
- w/o frequency inputs: This variant retained only the time-domain signal encoder by entirely removing the frequency branch.
- w/o dilation convolution: The dilated convolutions in the multi-branch structure were replaced by standard convolutions of the same kernel size.

Table 7 summarizes the performance of each ablated model. Across both datasets, removing the Transformer-inspired components led to the most substantial performance degradation, particularly on WC_TAN-MS where CORR decreased from 0.8838 to 0.7689 and both MAE and RMSE nearly doubled. This underscores the critical role of Layer Normalization, Dropout, and GELU in stabilizing feature extraction and enhancing correlation with actual machining errors.

Excluding the Adapter modules also notably weakened the model's ability to adapt through few-shot fine-tuning. For example, on WC_TAN-MS, CORR dropped from 0.8838 to 0.3892, highlighting the importance of lightweight feature transformation via adapters for rapid adaptation to new machining contexts.

Omitting the frequency-domain input resulted in further performance drops, especially in correlation on WC_TAN-MS, indicating that frequency features complement the time-domain signals and strengthen the model's robustness under varying cutting conditions.

Finally, replacing dilated convolutions with standard convolutions also led to consistent reductions in performance. Although the impact was less pronounced than omitting frequency inputs or adapters, the steady decrease in CORR across both datasets emphasizes the benefit of capturing broader temporal dependencies through dilations.

Table 8 Ablation study of signal sampling rate on adapted dataset

Dataset	Sampling Rate (Hz)	MAE	RMSE	CORR
WC_TAN-MS	102	0.0042	0.0051	0.3161
	204	0.0048	0.0057	0.3464
	512	0.0036	0.0048	0.4048
	1024	0.0020	0.0025	0.6500
	2048	0.0035	0.0049	0.3210
	5120	0.0030	0.0041	0.5283
	10240 (Original)	**0.0013**	**0.0016**	**0.8838**
WC_TC-AS	82	0.0028	0.0037	0.7449
	164	0.0029	0.0038	0.6939
	410	0.0028	0.0037	0.7086
	820	0.0029	0.0039	0.6830
	1638	0.0028	0.0039	0.7104
	4096	0.0030	0.0039	0.7087
	8192 (Original)	**0.0024**	**0.0032**	**0.7599**

Overall, these findings demonstrate that each architectural component, including the transformer-inspired components, adapters for fine-tuning, dual-domain inputs, and dilated convolutions, plays a vital role in ensuring DeepMachining's predictive accuracy and generalization.

To assess the effect of signal resolution on the proposed model's performance, we conducted an ablation study by systematically downsampling the input signals while keeping the adapted datasets unchanged. This analysis is essential for evaluating DeepMachining's deployment feasibility, particularly in edge environments with limited sensing or computational resources (Bernard et al., 2021).

Table 8 summarizes the results of this sampling rate ablation. For the WC_TAN-MS dataset, model performance declined markedly as the sampling rate decreased. Specifically, the CORR dropped from 0.8838 at 10,240 Hz to 0.3161 at 102 Hz, while both MAE and RMSE nearly quadrupled. This substantial degradation suggests that high-frequency components in the signal are critical for capturing diagnostic features specific to this turning configuration. A partial recovery at mid-level rates (e.g., 1,024 Hz yielding CORR $= 0.6500$) indicates that some low-frequency structures are retained, but aggressive downsampling discards fine-grained temporal information essential for accurate error estimation.

In contrast, the WC_TC-AS dataset exhibits substantially higher robustness to reductions in sampling rate. Even with a considerable drop from the original 8,192 Hz to 82 Hz, the model maintains a CORR of 0.7449. This performance is comparable to the baseline of 0.7599. While Table 8 indicates only marginal changes in MAE and RMSE across sampling rates (within ±0.0006), further examination of the scatter plots in Fig. 7 reveals a critical observation. At lower sampling rates, predicted values tend to converge toward a

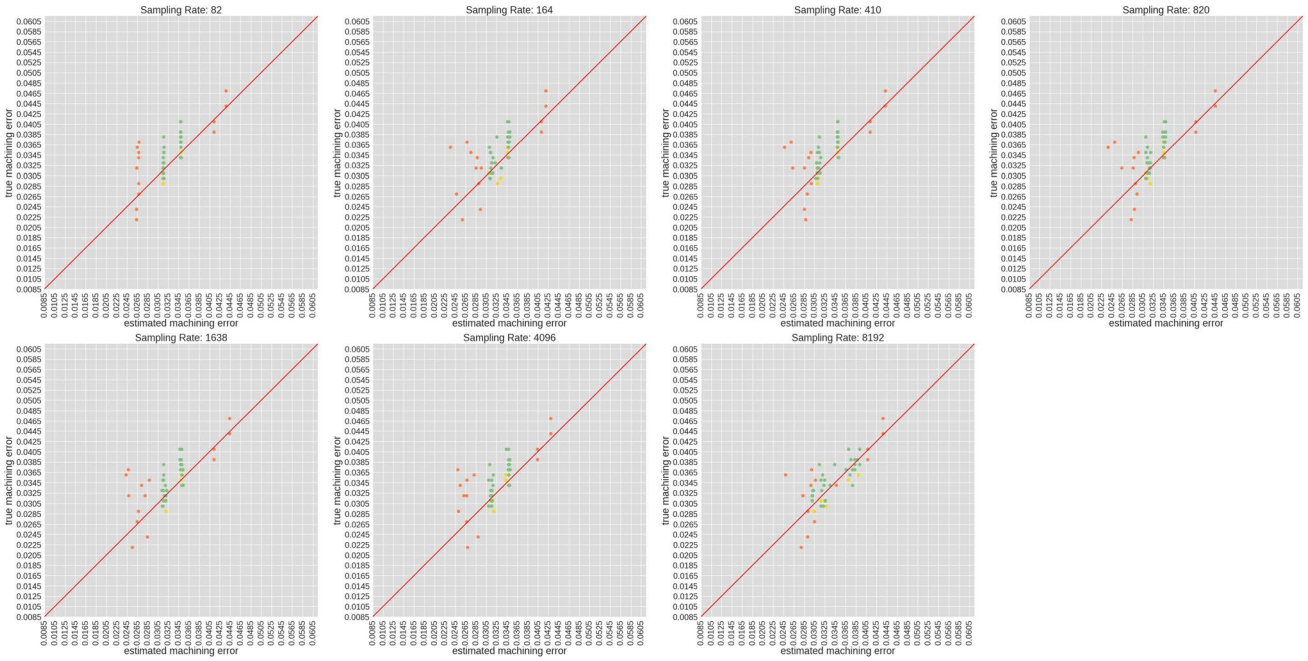


Fig. 7 Scatter plots of the actual machining errors versus the estimated errors at different sampling rates on the WC_TC-AS dataset. Different colors represent different machining configurations

limited range, reducing output variance. This phenomenon results in deceptively stable performance metrics, which obscure the model's inability to capture the full variability of true machining errors across different configurations (Ghosh et al., 2021).

This behavior can be attributed to two primary factors. First, the WC_TC-AS dataset inherently presents low variability in actual machining errors, which compresses the observable metric range and diminishes sensitivity to regression spread. Second, the axial force signals in this dataset predominantly contain low- to mid-frequency components, making them less prone to aliasing or distortion under aggressive downsampling. Consequently, even when the model yields oversmoothed predictions at low resolutions, standard error metrics such as MAE and RMSE remain deceptively low. Nonetheless, retaining high-frequency signal content enhances the model's capacity to capture fine-grained dynamics and produce physically meaningful predictions.

Collectively, these results underscore that sensitivity to sampling rate is inherently dependent on the machining task and signal spectral characteristics. As demonstrated by WC_TC-AS, conventional error metrics alone may fail to reveal degenerate prediction behaviors, such as variance collapse, especially under low signal variance conditions. This highlights the necessity of supplementing aggregate metrics with diagnostic visualizations to comprehensively assess model generalization and predictive fidelity (Ghosh et al., 2021). While high-frequency sampling remains essential for preserving discriminative features in configurations such as WC_TAN-MS, scenarios like WC_TC-AS allow for more lenient resolution constraints. DeepMachining maintains effective performance under reduced sampling conditions, provided that signal acquisition strategies are aligned with the frequency distribution and expected variability of the underlying process signals.

Discussion

Research hypothesis

One of the main challenges preventing the effective deployment of deep learning models in real-world machining environments is the diversity of machining conditions. Variations in tools, materials, and manufacturing settings often lead to distributional shifts that undermine the performance of deep learning models trained on large historical datasets, thereby resulting in degraded accuracy. To enable the practical application of deep learning in machining settings, models must be capable of adapting to changing machining conditions.

Model fine-tuning is a technique in deep learning designed to cope with shifts in data distributions. Since the training data typically only covers a limited subset of possible machining conditions, fine-tuning with newly collected data is necessary to adapt the model to novel scenarios. According to multiple studies on transfer learning

and distribution shift (Cohen-Wang et al., 2024; Wu et al., 2022; Kumar et al., 2022), model fine-tuning is effective only when the new data distribution does not fundamentally differ from the original training distribution, i.e., when the shift is due to bias rather than differences in the underlying nature of the data distribution.

Therefore, at the outset of this study, we made a strong assumption: the training data for the base model must include machining condition variations that cannot be corrected through model fine-tuning. In contrast, data associated with variations that can be addressed via fine-tuning can be reserved for future adaptation. Based on prior experiences, we posit that tool wear, the tool's life cycle, cannot be effectively compensated by fine-tuning alone. Consequently, the training data for the base model was constructed to cover the complete tool life cycle, from new to fully worn. Our experimental results finally validated this assumption.

Implementation insight and limitation

We conducted tests on DeepMachining across different types of products in various manufacturing factories. Several practical lessons were learned and are summarized as follows.

1. Sensor installation: Proper placement of sensors on CNC machines is crucial. Incorrect sensor placement may lead to ineffective signal acquisition, resulting in weak signal amplitudes or noise interference due to sensor cable tension.
2. Sensor sampling rate: Differences in sampling rates between the pre-training and fine-tuning stages can significantly impact model performance. To ensure the accurate functionality of DeepMachining, the sampling rate should remain identical during both stages.
3. Decimal precision: Low measurement precision—for example, measuring a workpiece with a required tolerance of $0.001\,\mathrm{mm}$ using an instrument with only 0.01 mm precision—cannot accurately reflect the differences in machining errors among workpieces. This lack of precision in the data can hinder the model's ability to generate accurate machining error estimations.

In this study, we evaluated the performance of DeepMachining exclusively on outer diameter machining tasks using a lathe machine, which limits its current application scope. However, CNC machining encompasses a broader range of processes, including internal turning, drilling, milling, and planing operations, across various machine types.

The proposed architecture of DeepMachining, comprising dual-domain signal encoders with D-Inception modules and lightweight adapters, is general in design and can be directly trained or fine-tuned on datasets collected from other machining processes. For instance, the same architecture can be applied to classify or regress quality metrics on milling or drilling signals, either by training from scratch or by leveraging pre-trained models on turning data to initialize representations and subsequently fine-tuning on milling or drilling data. This approach could also facilitate transfer learning for abnormal event detection in different CNC operations.

Comparison with transformer-based architectures

Transformer-based architectures are well-recognized for their ability to capture long-range dependencies via self-attention mechanisms. However, they typically incur substantially higher parameter counts and computational footprints compared to convolutional designs. Specifically, the self-attention operation scales quadratically with the input sequence length, resulting in significant memory and compute demands that pose challenges for real-time machining scenarios.

By contrast, the proposed DeepMachining framework leverages multi-branch convolutional modules (D-Inception) in conjunction with lightweight channel-temporal attention blocks. This design enables effective receptive field expansion and feature recalibration while maintaining a notably lower parameter and computational complexity than full self-attention mechanisms.

Quantitatively, the core of DeepMachining contains approximately **260,000** parameters, with only **6.5%** updated during fine-tuning. This is in sharp contrast to typical transformer-based approaches, which often comprise several million parameters and introduce considerable inference latency. Such parameter efficiency and reduced computational overhead make our approach particularly suitable for deployment on CNC edge devices subject to stringent resource constraints.

Conclusion

This paper presents DeepMachining, a deep learning-based framework for estimating machining errors in outer diameter processing using horizontal CNC lathe machines. DeepMachining consists of two stages: (1) pre-training a deep learning model using historical data, and (2) adapting the model to new machining tasks via few-shot learning, typically using two-shot adaptation.

The core model is initially trained on data collected from a single tool and workpiece material, covering the full tool life cycle from a new to a fully worn-out tool. When machining configurations change, the model can be adapted

by fine-tuning on just two workpieces from the new setting. This few-shot learning strategy allows DeepMachining to generalize effectively across different tools, workpieces, and machining conditions.

Experimental results show that DeepMachining consistently outperforms baseline methods in terms of estimation accuracy and generalizability. Notably, the core model is compact, containing approximately *260,000* parameters, and requires fine-tuning only *6.5%* of the parameters over *12.5%* of the training epochs used in pre-training, enabling deployment under limited computational resources. This design aligns with practical requirements in the CNC machining industry. Furthermore, the ablation study investigates the key components of DeepMachining and the impact of the sampling rate on the model, while considering its deployment on edge devices in CNC machines.

In addition, surface roughness is another important quality metric commonly used in CNC machining. Currently, DeepMachining has not yet been extended to address surface roughness estimation. Expanding DeepMachining to predict both machining error and surface roughness will be an important direction for future development. Another promising direction is to utilize DeepMachining for learning generalized CNC signal representations to predict the remaining useful life (RUL) of cutting tools. Ultimately, the long-term vision is to develop DeepMachining into an open-source intelligent manufacturing platform that can support a wide range of CNC applications and accelerate adoption in the industry.

Looking forward, the increasing complexity of manufacturing processes introduces richer interdependencies among machining data, machine logs, and product quality. Digital twin (DT) is an intelligent manufacturing solution for synchronizing machine tools within complex manufacturing environments (Ghosh et al., 2021). By integrating the proposed DeepMachining, DT can autonomously monitor and troubleshoot quality tasks, representing a promising research direction. Furthermore, large language models (LLMs) have demonstrated strong capabilities in understanding structured and unstructured data (Gautam et al., 2025). Leveraging LLMs for interpreting machine logs and process instructions holds promise for reducing machine downtime and enhancing product quality. Therefore, integrating LLMs into intelligent manufacturing represents a promising direction for future research.

Acknowledgements The authors would like to thank Victor Taichung Machinery Works Co. and their employees for the domain knowledge and the data shared in this project. We would also like to acknowledge the financial support of the Taiwan AI Academy.

Data availability The datasets supporting the findings of this study are partially available at https://github.com/ALiGoo/DeepMachining. The source code developed for the proposed method is also accessible at the same repository.

Declarations

Conflict of interest The authors declare no conflict of interest associated with the content of this study.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

Bahador, A., Du, C., Ng, H. P., Dzulqarnain, N. A., & Ho, C. L. (2022). Cost-effective classification of tool wear with transfer learning based on tool vibration for hard turning processes. *Measurement, 201*, 111701. https://doi.org/10.1016/j.measurement.2022.111701

Ben Zaken, E., Ravfogel, S., & Goldberg, Y. (2022). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 1–9). Dublin, Ireland: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.1.

Benkedjouh, T., Medjaher, K., Zerhouni, N., & Rechak, S. (2015). Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing, 26*(2), 213–223. https://doi.org/10.1007/s10845-013-0774-6

Bernard, G., Achiche, S., Girard, S., & Mayer, R. (2021). Condition monitoring of manufacturing processes under low sampling rate. *Journal of Manufacturing and Materials Processing, 5*(1), 26. https://doi.org/10.3390/jmmp5010026

Bhandari, B., Park, G., & Shafiabady, N. (2023). Implementation of transformer-based deep learning architecture for the development of surface roughness classifier using sound and cutting force signals. *Neural Computing and Applications, 35*(18), 13275–13292. https://doi.org/10.1007/s00521-023-08425-z

Brigham, E. O. (1988). *The fast fourier transform and its applications*. Englewood Cliffs, NJ, USA: Prentice-Hall.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*(NeurIPS 2020), 1877–1901.

Cai, H., Gan, C., Zhu, L., & Han, S. (2020). Tinytl: Reduce activations, not trainable parameters for efficient on-device learning. arXiv preprint arXiv:2007.11622.

Chen, W., Yang, K., Yu, Z., Nie, F., & Chen, C. L. P. (2025). Adaptive broad network with graph-fuzzy embedding for imbalanced noise data. *IEEE Transactions on Fuzzy Systems, 33*(6), 1949–1962. https://doi.org/10.1109/TFUZZ.2025.3543369

Chien, C.-F., & Chen, C.-C. (2020). Data-driven framework for tool health monitoring and maintenance strategy for smart manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 644–652. https://doi.org/10.1109/TSM.2020.3024284

Cohen-Wang, B., Vendrow, J., & Madry, A. (2024). Ask your distribution shift if pre-training is right for you. arXiv preprint arXiv:2403.00194.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 248–255). IEEE, Miami, FL, USA. https://doi.org/10.1109/CVPR.2009.5206848.

Denkena, B., Bergmann, B., & Witt, M. (2019). Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations. *Journal of Intelligent Manufacturing, 30*, 2449–2456. https://doi.org/10.1007/s10845-018-1404-0

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ding, Y., Zhuang, J., Ding, P., & Jia, M. (2022). Self-supervised pre-training via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety, 218*, 108126. https://doi.org/10.1016/j.ress.2021.108126

Du, C., Ho, C. L., & Kaminski, J. (2021). Prediction of product roughness, profile, and roundness using machine learning techniques for a hard turning process. *Advances in Manufacturing, 9*(2), 206–215. https://doi.org/10.1007/s40436-021-00345-2

Duro, J. A., Padget, J. A., Bowen, C. R., Kim, H. A., & Nassehi, A. (2016). Multi-sensor data fusion framework for cnc machining monitoring. *Mechanical Systems and Signal Processing, 66–67*, 505–520. https://doi.org/10.1016/j.ymssp.2015.04.019

Fernandes, M., Corchado, J. M., & Marreiros, G. (2022). Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: A systematic literature review. *Applied Intelligence, 52*(12), 14246–14280. https://doi.org/10.1007/s10489-022-03344-3

Gautam, A., Aryal, M. R., Deshpande, S., Padalkar, S., Nikolaenko, M., Tang, M., & Anand, S. (2025). Iiot-enabled digital twin for legacy and smart factory machines with llm integration. *Journal of Manufacturing Systems, 80*, 511–523.

Gavahian, A., & Mechefske, C. K. (2023). Motor current-based degradation modeling for tool wear hybrid prognostics in turning process. *Machines, 11*(8), 781. https://doi.org/10.3390/machines11080781

Ghosh, A. K., Ullah, A. M. M. S., Teti, R., & Kubo, A. (2021). Developing sensor signal-based digital twins for intelligent machine tools. *Journal of Industrial Information Integration, 24*, 100242. https://doi.org/10.1016/j.jii.2021.100242

Ghosh, A. K., Ullah, A. M. M. S., Teti, R., & Kubo, A. (2021). Developing sensor signal-based digital twins for intelligent machine tools. *Journal of Industrial Information Integration, 24*, 100242. https://doi.org/10.1016/j.jii.2021.100242

Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. (2019). Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics, 66*(9), 7316–7325. https://doi.org/10.1109/TIE.2018.2877090

Hamdan, A., Sarhan, A. A. D., & Hamdi, M. (2012). An optimization method of the machining parameters in high-speed machining of stainless steel using coated carbide tool for best surface finish. *The International Journal of Advanced Manufacturing Technology, 58*, 81–91. https://doi.org/10.1007/s00170-011-3392-5

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE, Las Vegas, NV, USA. https://doi.org/10.1109/CVPR.2016.90.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

Hesser, D. F., & Markert, B. (2019). Tool wear monitoring of a retrofitted cnc milling machine using artificial neural networks. *Manufacturing Letters, 19*, 1–4. https://doi.org/10.1016/j.mfglet.2018.11.001

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Huang, P.-M., & Lee, C.-H. (2021). Estimation of tool wear and surface roughness development using deep learning and sensors fusion. *Sensors, 21*(16), 5338. https://doi.org/10.3390/s21165338

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research* (Vol. 37, pp. 448–456). France: PMLR, Lille. https://proceedings.mlr.press/v37/ioffe15.html.

Jiang, Y., Yin, S., Dong, J., & Kaynak, O. (2021). A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sensors Journal, 21*(11), 12868–12881. https://doi.org/10.1109/JSEN.2020.3033153

Kumar, A., Raghunathan, A., Jones, R., Ma, T., & Liang, P. (2022). Fine-tuning can distort pretrained features and underperform out-of-distribution. arXiv preprint arXiv:2202.10054.

Kumar, P., Khalid, S., & Kim, H. S. (2023). Prognostics and health management of rotating machinery of industrial robot with deep learning applications–a review. *Mathematics, 11*(13), 3008. https://doi.org/10.3390/math11133008

Lee, C.-Y., & Chien, C.-F. (2022). Pitfalls and protocols of data science in manufacturing practice. *Journal of Intelligent Manufacturing, 33*(5), 1189–1207. https://doi.org/10.1007/s10845-020-01711-w

Lee, D. .-E., Hwang, I., Valente, C. M. O., Oliveira, J. F. G. D., & Dornfeld, D. A. (2006). Precision manufacturing process monitoring with acoustic emission. *International Journal of Machine Tools and Manufacture, 46*(2), 176–188. https://doi.org/10.1016/j.ijmachtools.2005.04.001

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing, 138*, 106587. https://doi.org/10.1016/j.ymssp.2019.106587

Liao, Y., Ragai, I., Huang, Z., & Kerner, S. (2021). Manufacturing process monitoring using time-frequency representation and transfer learning of deep neural networks. *Journal of Manufacturing Processes, 68*, 231–248. https://doi.org/10.1016/j.jmapro.2021.05.046

Li, W., Fu, H., Han, Z., Zhang, X., & Jin, H. (2022). Intelligent tool wear prediction based on informer encoder and stacked bidirectional gated recurrent unit. *Robotics and Computer-Integrated Manufacturing, 77*, 102368. https://doi.org/10.1016/j.rcim.2022.102368

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11976–11986). IEEE, New Orleans, LA, USA. https://doi.org/10.1109/CVPR52688.2022.01167.

Liu, H., Liu, Z., Jia, W., Lin, X., & Zhang, S. (2020). A novel transformer-based neural network model for tool wear estimation. *Measurement Science and Technology, 31*(6), 065106. https://doi.org/10.1088/1361-6501/ab7282

Li, Y., Zhou, Z., Sun, C., Chen, X., & Yan, R. (2024). Variational attention-based interpretable transformer network for rotary machine fault diagnosis. *IEEE Transactions on Neural Networks and Learning Systems, 35*(5), 6180–6193. https://doi.org/10.1109/TNNLS.2022.3202234

Marei, M., & Li, W. (2022). Cutting tool prognostics enabled by hybrid cnn-lstm with transfer learning. *The International Journal of Advanced Manufacturing Technology, 118*(3), 817–836. https://doi.org/10.1007/s00170-021-07784-y

Marei, M., Zaatari, S., & Li, W. (2021). Transfer learning enabled convolutional neural networks for estimating health state of cutting tools. *Robotics and Computer-Integrated Manufacturing, 71*, 102145. https://doi.org/10.1016/j.rcim.2021.102145

Mekid, S., & Ogedengbe, T. (2010). A review of machine tool accuracy enhancement through error compensation in serial and parallel kinematic machines. *International Journal of Precision Technology, 1*(3–4), 251–286. https://doi.org/10.1504/IJPTECH.2010.031657

Nasir, V., & Sassani, F. (2021). A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges. *The International Journal of Advanced Manufacturing Technology, 115*(9–10), 2683–2709. https://doi.org/10.1007/s00170-021-07374-0

Ntemi, M., Paraschos, S., Karakostas, A., Gialampoukidis, I., Vrochidis, S., & Kompatsiaris, I. (2022). Infrastructure monitoring and quality diagnosis in cnc machining: A review. *CIRP Journal of Manufacturing Science and Technology, 38*, 631–649. https://doi.org/10.1016/j.cirpj.2022.06.001

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Papananias, M., McLeay, T. E., Obajemu, O., Mahfouf, M., & Kadirkamanathan, V. (2020). Inspection by exception: A new machine learning-based approach for multistage manufacturing. *Applied Soft Computing, 97*, 106787. https://doi.org/10.1016/j.asoc.2020.106787

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2021). Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 487–503). Association for Computational Linguistics, Online . https://doi.org/10.18653/v1/2021.eacl-main.39.

Proteau, A., Tahan, A., Zemouri, R., & Thomas, M. (2023). Predicting the quality of a machined workpiece with a variational autoencoder approach. *Journal of Intelligent Manufacturing, 34*(2), 719–737. https://doi.org/10.1007/s10845-021-01822-y

Ramezani, S. B., Cummins, L., Killen, B., Carley, R., Amirlatifi, A., Rahimi, S., Seale, M., & Bian, L. (2023). Scalability, explainability and performance of data-driven algorithms in predicting the remaining useful life: A comprehensive review. *IEEE Access, 11*, 41741–41769. https://doi.org/10.1109/ACCESS.2023.3267960

Ross, N. S., Sheeba, P. T., Shibi, C. S., Gupta, M. K., Korkmaz, M. E., & Sharma, V. S. (2024). A novel approach of tool condition monitoring in sustainable machining of ni alloy with transfer learning models. *Journal of Intelligent Manufacturing, 35*(2), 757–775. https://doi.org/10.1007/s10845-023-02074-8

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sayyad, S., Kumar, S., Bongale, A., Kotecha, K., Selvachandran, G., & Suganthan, P. N. (2022). Tool wear prediction using long short-term memory variants and hybrid feature selection techniques. *The International Journal of Advanced Manufacturing Technology, 121*(9–10), 6611–6633. https://doi.org/10.1007/s00170-022-09784-y

Schwendemann, S., Amjad, Z., & Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry, 125*, 103380. https://doi.org/10.1016/j.compind.2020.103380

Schwenzer, M., Auerbach, T., Miura, K., Döbbeler, B., & Bergs, T. (2020). Support vector regression to correct motor current of machine tool drives. *Journal of Intelligent Manufacturing, 31*, 553–560. https://doi.org/10.1007/s10845-019-01464-1

Serin, G., Sener, B., Ozbayoglu, A. M., & Unver, H. O. (2020). Review of tool condition monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology, 109*, 953–974. https://doi.org/10.1007/s00170-020-05449-w

Serradilla, O., Zugasti, E., Rodriguez, J., & Zurutuza, U. (2022). Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. *Applied Intelligence, 52*(10), 10934–10964. https://doi.org/10.1007/s10489-021-03004-y

Soori, M., Arezoo, B., & Dastres, R. (2023). Machine learning and artificial intelligence in cnc machine tools, a review. *Sustainable Manufacturing and Service Economics, 2*, 100009. https://doi.org/10.1016/j.smse.2023.100009

Sun, C., Ma, M., Zhao, Z., Tian, S., Yan, R., & Chen, X. (2019). Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. *IEEE Transactions on Industrial Informatics, 15*(4), 2416–2425. https://doi.org/10.1109/TII.2018.2881543

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 1–9). IEEE, Boston, MA, USA. https://doi.org/10.1109/CVPR.2015.7298594.

Ura, S., & Ghosh, A. K. (2021). Time latency-centric signal processing: A perspective of smart manufacturing. *Sensors, 21*(21), 7336. https://doi.org/10.3390/s21217336

Wang, P., & Gao, R. X. (2020). Transfer learning for enhanced machine fault diagnosis in manufacturing. *CIRP Annals, 69*(1), 413–416. https://doi.org/10.1016/j.cirp.2020.04.074

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data, 3*(1), 1–40. https://doi.org/10.1186/s40537-016-0043-6

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., & Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45). Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – ECCV 2018. Lecture notes in computer science* (Vol. 11211, pp. 3–19). Cham: Springer. https://doi.org/10.1007/978-3-030-01234-2_1

Wu, H., Triebe, M. J., & Sutherland, J. W. (2023). A transformer-based approach for novel fault detection and fault classification/diagnosis in manufacturing: A rotary system application. *Journal of Manufacturing Systems, 67*, 439–452. https://doi.org/10.1016/j.jmsy.2023.02.018

Wu, J., Zou, D., Braverman, V., Gu, Q., & Kakade, S. (2022). The power and limitation of pretraining-finetuning for linear regression under covariate shift. *Advances in Neural Information Processing Systems, 35*, 33041–33053.

Yang, K., Chen, W., Shi, Y., Yu, Z., & Chen, C. L. P. (2024). Simplified kernel based cost-sensitive broad learning system for imbalanced fault diagnosis. *IEEE Transactions on Artificial Intelligence, 5*(12), 6629–6644. https://doi.org/10.1109/TAI.2024.3478191

Yeganefar, A., Niknam, S. A., & Asadi, R. (2019). The use of support vector machine, neural network, and regression analysis to predict and optimize surface roughness and cutting forces in milling. *The International Journal of Advanced Manufacturing Technology, 105*, 951–965. https://doi.org/10.1007/s00170-019-04227-7

Yu, Z., Huang, S., Yang, K., Lv, J., & Chen, C. L. P. (2025). Ensemble approaches for dynamic data stream classification under label

scarcity. *IEEE Transactions on Big Data*, 1–15. https://doi.org/10.1109/TBDATA.2025.3570072.

Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H., He, S., & Xu, E. (2022). Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA Transactions, 119*, 152–171. https://doi.org/10.1016/j.isatra.2021.10.025

Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal, 13*(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing, 115*, 213–237. https://doi.org/10.1016/j.ymssp.2018.05.050

Zhu, K., Li, G., & Zhang, Y. (2020). Big data oriented smart tool condition monitoring system. *IEEE Transactions on Industrial Informatics, 16*(6), 4007–4016. https://doi.org/10.1109/TII.2019.2957107