New data-driven predictive modelling methods for data scarcity scenarios in smart manufacturing

Gengxiang Chen

THESE DE DOCTORAT

NNT : 2023UPAST129

Nouvelle démarche de développement de modèles prédictifs de pilotage par les données en Smart-Manufacturing dans les cas de pénurie de données par l'utilisation de techniques de machine-Learning et l'IA.

New data-driven predictive modelling methods for data scarcity scenarios in smart manufacturing

Thèse de doctorat de l'université Paris-Saclay et de l'université Nanjing University of Aeronautics and Astronautics

École doctorale n 579 Sciences mécaniques et énergétiques, matériaux et géosciences SMEMAG
Spécialité de doctorat : génie mécanique
Graduate School : Sciences de l'ingénierie et des systèmes

Thèse préparée à l'ENS Paris-Saclay en cotutelle internationalle entre l'Université Paris-Saclay, et Nanjing University of Aeronautics and Astronautics au Laboratoire Universitaire de Recherche en Production Automatisée (LURPA) sous la co-direction et co-encadrement de Charyar Mehdi-Souzani, Maître de conférences USPN-LURPA , de Yingguang Li, Full professor NUAA et d'Olivier Bruneau, Professeur des universités UPS-LURPA,

Thèse soutenue à Paris-Saclay, le 25 Septembre 2023, par

Gengxiang CHEN

Composition du Jury

Membres du jury avec voix délibérative

M. James Gao Full professor, University of Greenwich	Président
M. Jean-Yves Dantan Professeur des universités, Arts et Métiers ParisTech	Rapporteur
M. Kai Tang Full professor, Hong Kong University of Science and Technology	Rapporteur
M. Nabil Anwer Professeur des universités, Université Paris-Saclay	Examinateur
M. Changqing Liu Full professor, Nanjing University of Aeronautics and Astronautics	Examinateur

Acknowledgements

This thesis is finished under the framework of joint international supervision of Laboratoire Universitaire de Recherche en Production Automatisée (Université Paris-Saclay, ENS Paris-Saclay) and Nanjing University of Aeronautics and Astronautics (NUAA). I am deeply grateful to numerous people who have played pivotal roles in guiding and supporting me throughout this remarkable journey.

First, I would like to express my heartfelt gratitude to Professor Yingguang Li, my supervisor in NUAA. To be honest, you are not only my research supervisor but also a mentor guiding me in life and teaching me to grow up.

Next, I want to thank Professor Charyar Mehdi-Souzani, my supervisor in ENS. You have spent a lot of time helping me find the right direction for this project, helping me to build the framework step by step, and whenever I've encountered technical challenges or roadblocks, you will bring me back to the right path. You always encourage me to broaden my international research perspective, and give me so many opportunities. Really appreciate.

I also want to express my thanks to Professor Olivier Bruneau, the head of our lab LURPA. Over the past three years, you have consistently made time to assist me, whether it was related to research, administrative matters, or personal life, and for that, I am truly appreciative.

I would like to convey my appreciation to the esteemed members of my jury: Prof. James Gao, Prof. Jean-Yves Dantan, Prof. Kai Tang, Prof. Nabil Anwer, and Prof. Changqing Liu, for their attendance at my presentations and their valuable comments and suggestions during my defense.

I am lucky to have the opportunity to work in two labs in different countries. My lab in NUAA is Ideahouse, with the spirits of *Ambitious*, *Proactive*, and *Passionate*, which have also been my own personal goals and aspirations. Thanks for Prof. Xu Liu for the consistent support and encouragement. Thanks to Zhiwei, Tianchi, Yinghao, Yunfei, Wenjun, Lu, Qinglu, Bo, Lin, Zhiliang, Xiaoyang and other ambitious colleagues in Ideahouse. Your ambition and dedication have enriched my time here.

Special thanks to all my wonderful colleagues in LURPA. We have a great lab culture, serious and relaxing. We are very serious about the coffee break time, and babyfoot time, but very relaxing about working and meeting time. I really like it. Many thanks to Yifan, Clément, Marc-Antoine, Yan, Romain, Kevin, Michèle, Khalil, Charles, Louis, Carlos, Syrine, Léo, Vi, Xinping, Baptiste and other students. Thanks to the support team in LURPA, Philip, Marc and Floriane. I appreciate my kind colleagues for helping me finish experimenting, improving my representation, teaching me to play babyfoot, and also teaching me to prarle francais. I really enjoy my time here in LURPA.

Finally, I want to thank my friends and family, my parents, my wife Jin Wang and my newborn daughter Yining Chen. I cannot finish this long journey without their support.

Contents

List of Figures

List of Tables

Chapter 1

Data modelling for smart manufacturing: Context and challenges

Je pense, donc, je suis.

– René Descartes

1.1 Introduction

In pursuit of customisation and smartness in the manufacturing industry, continuous in-depth integration of digital and intelligent manufacturing technologies led to the concept of smart manufacturing [1, 2]. Meanwhile, the development of modern information technologies also initiated and enriched various paradigms of smart manufacturing, such as digital twin, Industry 4.0, cloud manufacturing and IoT-enabled manufacturing [2, 3]. As shown in Fig. 1.1, smart manufacturing can generate enormous benefits spanning across the entire manufacturing life cycle, including intelligent maintenance, procedure optimisation, and process modelling.[4, 5]

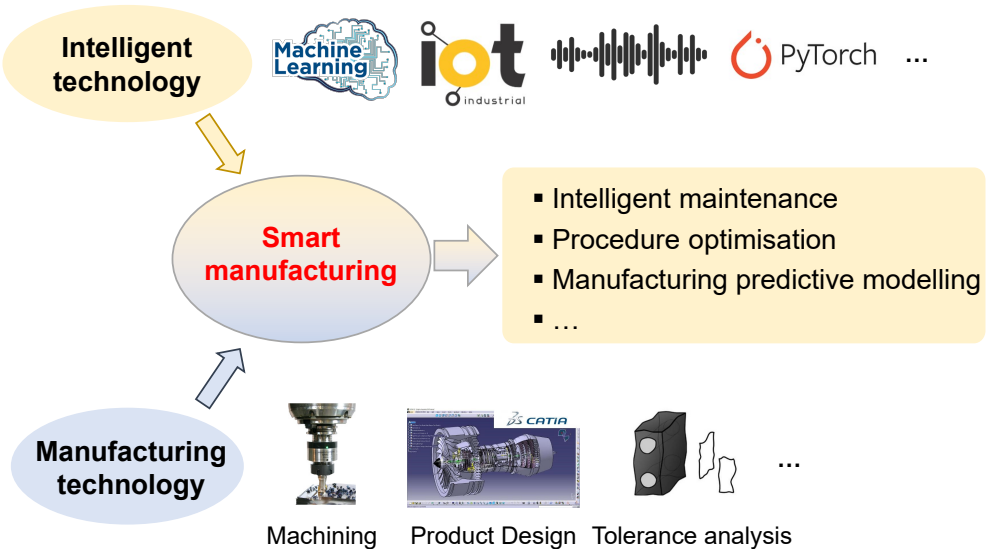


Figure 1.1: Smart manufacturing concept and applications [1].

As a fundamental topic of smart manufacturing, **manufacturing predictive modelling (MPM)** aims to construct high-fidelity predictive representations of the concerned properties of products, processes or manufacturing systems, such as the health conditions of machine tools, the surface roughness of products, or the milling dynamic stability states, as shown in Fig. 1.2 [6, 7]. A well-established predictive model is necessary for subsequent process optimisation and decision-making [8]. Take for example composite manufacturing, the large size, complex shape and high accuracy requirements of aerospace composite parts impose increasing demands on deformation control during curing process [9]. Therefore, constructing the predictive model from the curing temperature to the final deformation field can provide essential support for further curing process optimising [10, 11].

Traditional **mechanism-based modelling** methods aimed to construct the basic functional form to describe the nonlinear manufacturing process based on a series of physical and chemical laws or mechanisms [12]. Different mechanism models have been developed and widely applied in simulating the manufacturing systems and processes, such as the dynamic milling model [13] and the thermochemical curing model of composites [14]. Since manufacturing processes are always accompanied by a large number of non-linear multi-physics dynamics, mechanism-based modelling contains inevitable

Figure 1.2: Manufacturing predictive modelling

assumptions and simplifications, and sometimes ill-posed solutions due to the limited capacity of describing complex manufacturing processes [3]. Therefore, it is of great difficulty or even impossible to establish accurate mechanism models [15].

With the substantial developments of machine learning techniques, **data-driven modelling**, which can learn the relationships between complex influence factors and concerned properties from collected data, has gained considerable attention in both academic and engineering fields [4, 16]. MPM problems can be categorised as supervised learning tasks, where model training requires labelled data, meaning that each data sample contains the input, also known as the feature, and an associated output label [17]. In contrast to mechanism-based models, data-driven models can establish precise nonlinear relationships between inputs and outputs with powerful expressive capacity, without requiring assumptions of mechanisms or formulas. Therefore, different data-driven modelling methods have been developed and applied in various MPM problems, such as composite deformation analysis [18], milling stability analysis [19] and tolerancing for additive manufacturing [20].

However, the high performance and generalisability of data-driven modelling methods heavily rely on significant amount of labelled data [21]. Although advanced sensors and high-speed communication techniques enable companies to collect massive data of manufacturing process, these data can truly enable and promote smart manufacturing only when labelled with meaningful information [22]. Nevertheless, it is usually expensive and time-consuming to label manufacturing data both computationally and experimentally. Taking composite manufacturing as an example, a complete curing state simulation of a typical aerospace composite part (e.g. the wing skin of the Boeing 787) requires several hours or days [10], and the actual curing experiments are even more expensive for accumulating labelled data [23]. To train a data-driven cutting tool dynamics prediction model, it would take weeks to carry out a large amount of impact testing experiments manually for label collection [24, 25]. Besides, in the era of customised manufacturing, the variable configuration and flexible requirements make it impossible to collect sufficient labelled data for all potential manufacturing configurations [6, 26]. Therefore, establishing data-driven MPM models with limited labelled data is an inevitable challenge for the development of smart manufacturing technologies [27, 28].

Therefore, this research focused on data-driven manufacturing predictive modelling

under data scarcity. Generally speaking, the information carried in the data set directly determines the upper bound of the performance of the data-driven models, especially in circumstances of data scarcity. Previous researchers focused on how to train a data-driven model based on given dataset. This presupposition deprives us of the possibility of actively exploiting the data-generating process. Therefore, by reassessing the complete process from data generation to data modelling, this project proposed to solve the data scarcity challenge from the following perspectives:

- If only a small number of labelled data can be collected, how to collect a better dataset that can benefit model training?
- In cases where labelled data is insufficient, how to leverage other related data or mechanisms to enhance the performance of the data-driven model?

1.2 Global overview and literature review

1.2.1 The concept of smart manufacturing

The fourth industrial revolution was formalised in the mid-2000s and today has become the heart of the development of the more than ever globalised world industry [29]. This revolution started with the advent of information, and its dissemination through communication at the heart of the so-called industry 4.0, finally promoting the generation and development of smart manufacturing.

Scholars and institutions have given several interpretations of smart manufacturing concerning different application scenarios and perspectives. Tao et al. [30] described smart manufacturing aims to *convert data acquired across the product lifecycle into manufacturing intelligence in order to yield positive impacts on all aspects of manufacturing*, where the key element is data and the enabler is manufacturing intelligence, namely the fusion of machine learning and manufacturing. Similarly, Jianrong et al. [1] described the connotation of smart manufacturing as *the integration of both manufacturing and intelligent technologies to solve manufacturing problems.*

Lu et al. [31] defined smart manufacturing as *fully-integrated, collaborative and responsive operations that respond in real time to meet changing demands and conditions in the factory, in the supply network, and in customer needs via data-driven understanding, reasoning, planning, and execution of all aspects of manufacturing processes, facilitated by the pervasive use of advanced sensing, modelling, simulation, and analytic technologies.* This definition points out the various application scenarios, data-driven solutions and also potential sources of data. Wallace and Riddick [32] described smart manufacturing as *a data-intensive application of information technology at the shop floor level and above to enable intelligent, efficient, and responsive operations.*

Recent research brought more connotations to smart manufacturing toward sustainable, resilient and human-centric [33]. Andrew et al. [34] pointed out that resiliency and sustainability in smart manufacturing are worthy of business consideration, and then

introduced several trends of data-driven manufacturing resiliency. Research in human-centric smart manufacturing focused more on the integration of human-in-the-loop with technologies, to address challenges of human-machine relationships based on human-generated data and product-sensed data [35].

Despite the slight differences in the above definitions, it is indisputable that data is the irreplaceable fuel and advanced data modelling methods are the enablers for smart manufacturing. Smart manufacturing can meet the demands in various applications, including factory level [36, 37], supply chain level [38, 39] and manufacturing process level[40, 16]. This research aimed to develop advanced data-driven techniques for manufacturing predictive modelling.

1.2.2 Data-driven technologies in smart manufacturing

With the increasing complexity of manufacturing systems, traditional mechanism-based modelling cannot satisfy the above-mentioned requirements for smart manufacturing. Data-driven manufacturing utilises data analytics and machine learning techniques to exploit the data from manufacturing to refine the manufacturing process, improve the flexibility and smart level of manufacturing [30]. Different data-driven smart manufacturing frameworks have been developed by previous researchers to provide support and guidance for the application of data-driven techniques [4]. This section will review the two development stages of data-driven smart manufacturing frameworks, and then analyse their characteristics and limitations.

In the early stage of developing a data-driven smart manufacturing framework, many researchers focused on the data perspective regarding the life cycle of data from the manufacturing system and the data processing techniques. Tao et al. [30] introduced a manufacturing data life cycle from data collection, data storage, and data processing to final data application, as well as the key techniques of each step. Based on this, they developed a data-driven smart manufacturing framework consisting of the manufacturing module, data-driven module, real-time monitoring module, and problem processing module. Majeeda et al. [41] proposed a similar big-data-driven smart additive manufacturing framework by defining the whole data flow in the manufacturing cycle.

These frameworks described the behaviour flows of data in the manufacturing system, thus deepening the understanding of the significance of data. Furthermore, as shown in Fig. 1.3, Wang et al. [42] proposed a framework of big-data-driven smart manufacturing by combining 'correlation', 'prediction' and 'regulation', which meant analysing the correlation from the perspective of data, predicting with machine learning methods, and optimising process based on the prediction results. Kong et al. [22] developed a data construction framework that integrated data collecting with subsequent data organisation and data representation.

These data-perspective frameworks provided valuable guidance for various manufacturing applications [43]. However, despite these achievements, various shortcomings of these pure data-driven modelling methods, in particular deep learning algorithms, have drawn more attention to the additional problems in manufacturing predictive mod-

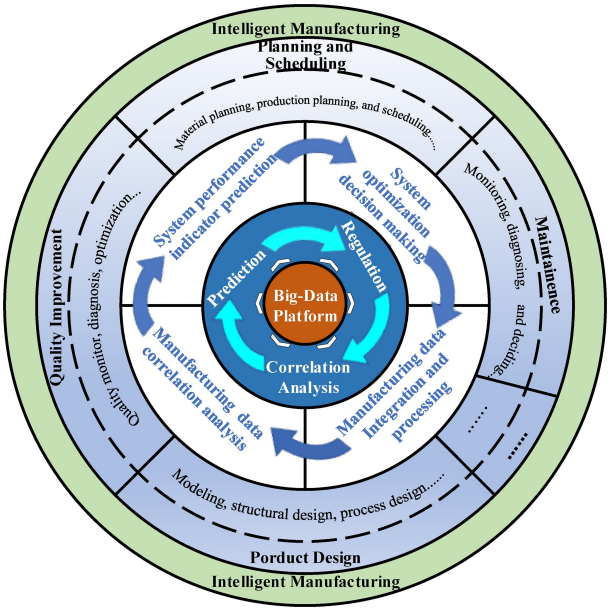


Figure 1.3: The framework of big data-driven intelligent manufacturing. [42].

elling. First, the requirements for the huge amount of labelled data bring non-negligible labelling costs. Furthermore, there is still a lack of understanding about the interpretability and reliability of data-driven modelling [44]. Recent researchers also started to think that it may not be wise to ignore existing mechanism knowledge in pursuing purely data-driven modelling. Based on this inspiration, physics-informed data-driven modelling was proposed in machine learning and different engineering scenarios, including manufacturing predictive modelling [45]. The term 'physics' here refers to a general concept that includes not only physical and chemical laws, but also mechanistic formulas and well-studied prior knowledge. For manufacturing problems, the empirical knowledge about process parameters and machining equipment can also be defined as 'physics' for improving pure data-driven models.

Recent MPM research started to leverage physics knowledge to improve the interpretability of data-driven models and reduce data requirements [46]. As shown in Fig. 1.4, Mozaffar et al. [47] introduced Mechanistic Artificial Intelligence for manufacturing process, defined as the methods that combined the raw mathematical power of AI methods with mechanism-driven principles and engineering insights. Similarly, Wang et al. [46] reviewed several data-physics combination strategies and defined a Hybrid Physics-based and Data-driven framework for smart manufacturing. The three defined modelling schemes were physics-informed machine learning, machine learning-assisted simulation, and explainable artificial intelligence. Greis et al. [48] developed a physics-guided self-aware approach that used numerical experiments to enhance the performance of the data-driven model of experimental data. Similarly, Saha et al. [49] proposed a Hierarchical Deep Learning Neural Netowrk that combined data and physics on different levels, as shown in Fig. 1.5.

Although various information was defined as 'physics' in the above research, the detailed form of 'physics' and its influence on modelling remains unclear. For example, the

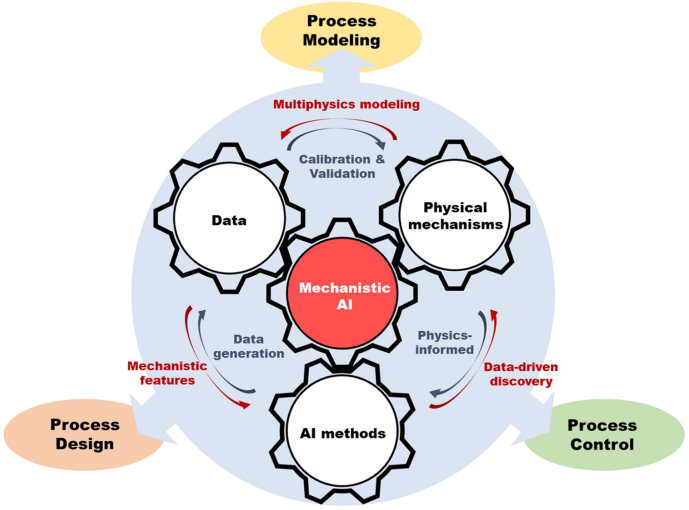


Figure 1.4: Mechanistic Artificial Intelligence (Mechanistic-AI) for advanced manufacturing processes. [47]

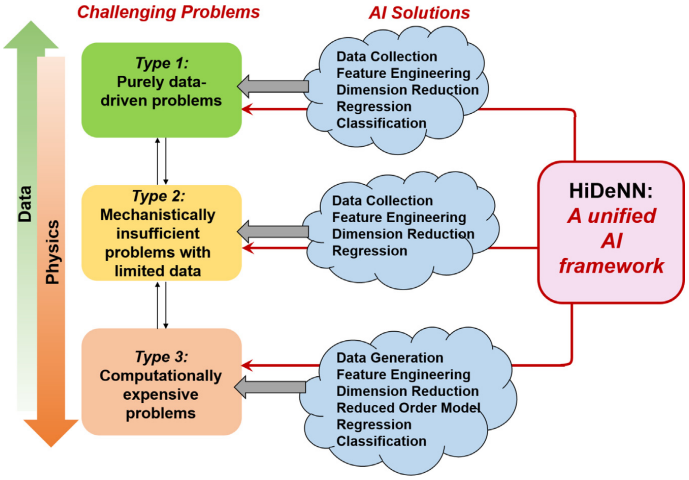


Figure 1.5: Hierarchical Deep Learning Neural Netowrk (HiDeNN) for engineering. [49]

mechanism or physics formulas can be interpreted and generalised. But these properties will degrade when we use these models to generate simulation data. Some simulation software tools have been developed based on the physics and mechanism of manufacturing process, such as Vericut for cutting simulation [50] and Comsol for composite curing simulation [26]. Although the simulation models are mechanism-based, the simulated data can also be used to support the building of data-driven MPM models. The milling stability model follows the dynamic equilibrium equation strictly [13]. But after simulating the discrete milling stability data from the model, the simulation data cannot provide the milling state prediction for the cutting parameters out of the distribution [51]. That means that simulation data cannot hold the interpretability and generalisability of the original physics model. Therefore, simulation data and physics formulas have different characteristics and deserve different treatment, although they are both defined as 'physics' information in previous research. A well-designed framework should carefully distinguish between the generalisable physics information and non-generalisable

simulation data, as well as other available data from the manufacturing system.

The previous research in data-driven manufacturing frameworks had deeply investigated the life cycle of data and revealed that the widely recognised data scarcity problem restricted further applications of data-driven manufacturing. The previous research showed that the integration of physics knowledge could be a potential solution for data-driven manufacturing under data scarcity [49, 47]. For example, the cutting force model can be used to design the specific neural network [52], the milling dynamics model can be used to compensate for the insufficient chatter experimental data [19]. However, there is a lack of a systematic framework that clearly analyses the characteristics of different available information and the practicable techniques that can make full use of various types of data. This is one of the gaps addressed by this research.

1.2.3 Manufacturing process data and modelling techniques

As a fundamental topic of smart manufacturing, data-driven MPM aims to build the predictive model of the concerned properties of manufacturing based on the input data of interest [6]. Therefore, data-driven MPM can be categorised as **supervised learning**, which is defined as the machine learning task of learning a function that maps an input to an output based on the given input-output pairs [53, 54]. The input and output are defined as features and labels respectively, and their combination is named labelled data [54]. The effectiveness of data-driven MPM depends on both the data and the modelling approach. Therefore, this section will first introduce manufacturing process data collection and analyse the characteristics of different data. After that, a series of data-driven modelling methods and their applications are provided in detail.

1.2.3.1 Manufacturing process data collection

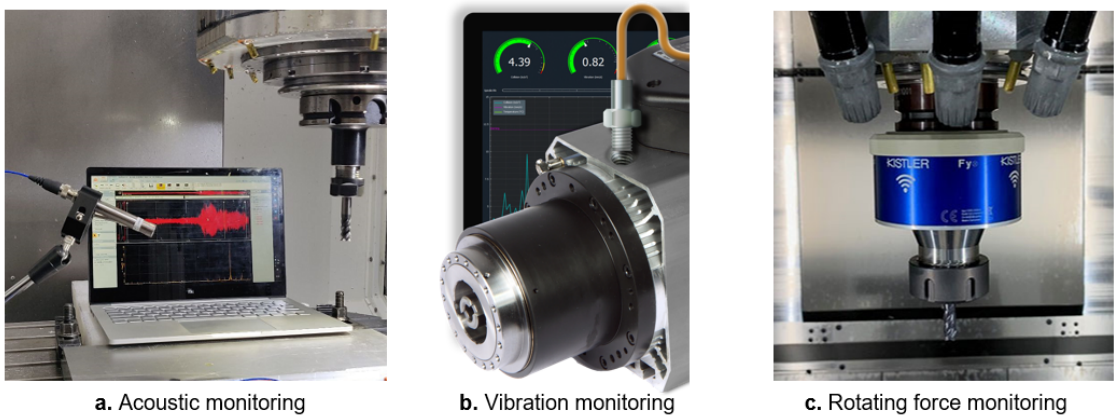
Data collection is the foundation of data-driven MPM. The manufacturing data may come from online monitoring, offline measurement, simulation or historical accumulation. The cost and quality of data collection will influence the selection of modelling as well as the potential application scenarios [54]. This section will introduce the related work about manufacturing data from different sources. Besides, the characteristics of data are also analysed to reveal their influence on the subsequent modelling.

(i) Data collection from online monitoring

The development of sensory techniques enables real-time collection of various physical signals from manufacturing system, namely online monitoring [55]. Since the setup and embedding of sensors do not interfere with the manufacturing process, online monitoring can record the evolution of the concerned attributes completely during the manufacturing process [56, 57].

For milling or turning scenarios, the cutting force during material removal can lead to variable current and power of the spindle and feed axes, thus the machining process

can be monitored by the current and power sensors installed on machine tools [15]. At the same time, since the cutting force will cause the vibration of the cutting tool, workpiece, and then the vibration of the air, the machining process can also be monitored by installing an acceleration sensor on the part, or setting up a microphone sensor to monitor the acoustic signal [56]. Fig. 1.6a shows an acoustic monitoring example for milling. Fig. 1.6b and Fig. 1.6c are advanced milling process monitoring sensors, namely 3D piezoelectric technique from Precision Drive Systems [58] and rotating dynamo-meters from Kistler [59], respectively.



a. Acoustic monitoring **b.** Vibration monitoring **c.** Rotating force monitoring

Figure 1.6: Data collection by online monitoring.

For the curing of composites workpiece, the curing degree, viscosity, flowing and other properties of the resin will change along with the variable external heat and pressure [60]. At the same time, the thermal expansion and chemical shrinkage of the material will also lead to the inevitable curing stress [61]. Monitoring these properties with thermocouples [62], Fiber Bragg Grating (FBG) or other advanced sensors is necessary for the quality control and process optimisation of CFRP curing [63].

The increasing number of sensors and the sampling frequency of the equipment make it possible to build large-scale monitoring data for almost all manufacturing processes, including structural part milling[64] and metal additive manufacturing[65]. Although monitoring data can comprehensively describe the real changing process of physical properties in the manufacturing process, the low value-density and the big volume bring significant challenges for data processing and data modelling [66]. Meanwhile, the high volume of monitoring data does not mean sufficient information for manufacturing predictive modelling. These data can only enable and promote smart manufacturing when labelled with meaningful information [40]. Normally, the monitoring data play the role of input feature of deep learning models to predict the attributes of interest.

In summary, the monitoring data is usually big volume, and easily accessible because of advanced sensors. But the value density is relatively low and thus requires further data mining and data processing to obtain useful labelled data. Besides, the online monitoring data is used as the input feature for predictive modelling, which means these data have to be labelled with other concerned information.

(ii) Data collection from offline measurement

Offline measurement refers to the targeted manual data collection during the period out of manufacturing behaviour, such as the idle state of manufacturing equipment [67], and when carrying out quality assessment of manufactured products [68]. Generally, specialised equipment will be utilised to collect the data that characterise the key properties of manufacturing systems or products, and these data can potentially contribute to assessing or optimising the manufacturing process.

The key properties of products include geometrical information [69], mechanical properties [70] and other indicators that can be used to evaluate the performance of the manufacturing process. For milling operations, the geometrical data collection consists of the interim geometrical measurement of the workpiece and the final quality assessment after machining [71]. The concerned indicators include the dimensional deviations measured by Coordinate Measurement Machine, and the surface texture from the roughness tester [72]. For additive manufacturing process, the X-ray computed tomography (XCT) is widely used for its ability to reconstruct the internal structures of the workpiece [73]. Besides the geometrical information, the mechanical properties, material characterisation and micro-morphology have also received a lot of attention in additive manufacturing [74].

The key properties of a manufacturing system include the equipment parameters, healthy state or other internal indicators that can reflect or influence the manufacturing process [55]. Take an industrial robot as an example, the kinematics parameters, dynamics parameters, and the load capacity of the end effector are all influential parameters for robot welding [75] or robot polishing [76]. For a machine tool, the pose-dependent dynamics [77], tool wears [78] and the repeatability [79], are properties considered by engineers and researchers.

The above-mentioned key properties of the manufacturing system are usually the target or the labels in predictive modelling. The effectiveness of data-driven models heavily relies on sufficient labelled data. However, it is normally expensive and time-consuming to collect sufficient above key property labels because of the expensive measurement equipment, complex measurement procedures and even strict requirements of operations [80]. For the tool wear prediction problem, the input feature is the monitoring signal, and the output label is the tool wear value. The combination of signal data and the corresponding tool wear constitute a labelled sample. As shown in Fig. 1.7, to measure the tool wear of milling, the machine tool has to be stopped first, then the cutting tool is removed from the spindle and fixed in front of a special microscope [80]. Engineers have to find both the original boundary and the broken boundary of the blades, and then calculate the final wear value. It will take more than 10 minutes to collect one tool wear label.

Fig. 1.8 shows an example of collecting the pose-dependent tool tip dynamics of a five-axes machine center [24]. Tool tip dynamics, including natural frequency w, damping ratio ξ and stiffness K, are very important for chatter suppression in machining complex parts like the aero-engine impeller. In the machine's workspace defined by three linear axes (X, Y, Z) and two rotating axes (A, C), the tool tip dynamics (w, ξ, K) usu-

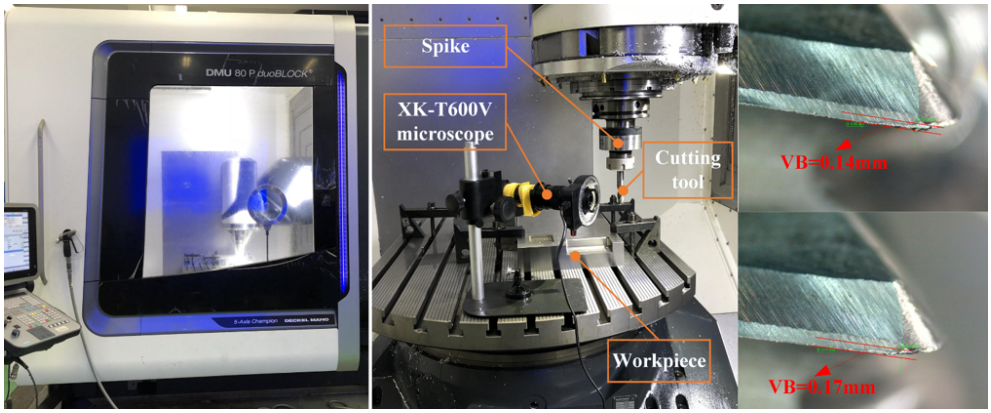


Figure 1.7: Tool wear measurement using microscope. [80]

ally varies at different positions, thus the data-driven model f that maps (X, Y, Z, A, C) to (w, ξ, K) is required for predicting the pose-dependent dynamics. The input feature of this case refers to the coordinates of the machine tool, and the output labels are the corresponding tool tip dynamics. The data pair of the coordinate and the tool tip dynamics constitute one labelled sample. Since the workspace of the machine tool is huge, for each tool-holder assembly, the machine has to be turned off for a couple of weeks to carry out a large amount of impact testing experiments manually for label collection in different coordinates[81]. Moreover, a CNC machine usually has more than one hundred frequently used tool-holder assembles. Therefore, it is impossible to collect enough tool tip dynamics data in real manufacturing practice [24].

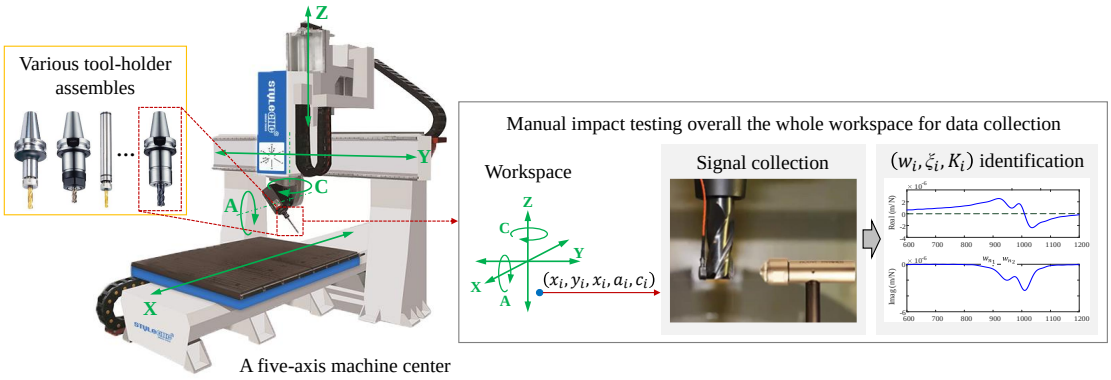


Figure 1.8: Machine center pose-dependent dynamics measurement. [24]

To sum up, the key properties of manufacturing processes can be defined as the to-be-predicted labels in data-driven MPM problems, such as the above-mentioned tool wear value. These property data are normally collected from ineffective offline manual measurements with specialised instruments, which means that it is difficult to collect sufficient labelled data for real applications.

(iii) Data collected from simulation

With the continual deep understanding of manufacturing processes, various simula-

tion software tools have been developed based on mechanism knowledge, which further benefits manufacturing process analysis and optimisation. Multi-body dynamics (MBD) models can analyse the kinematics and dynamics of manufacturing systems, including industrial robots and machine tools [82]. Fig. 1.9 shows an example of machine center pose-dependent dynamics simulations. The complexity of machine tool brings significant computational efforts for the simulation data generation. Finite element analysis softwares are widely used in simulating metal cutting [83], additive manufacturing[84] and composite curing[85]. Simulation data has also plays important roles in data-driven MPM. For example, surrogate model based optimisation requires simulation data to train a high-fidelity replica of the simulation process for the iterative process optimisation [86]. Compared with experimental data, simulation data have its own characteristics in accuracy and low collection cost, thus deserves different processing in data-driven MPM.

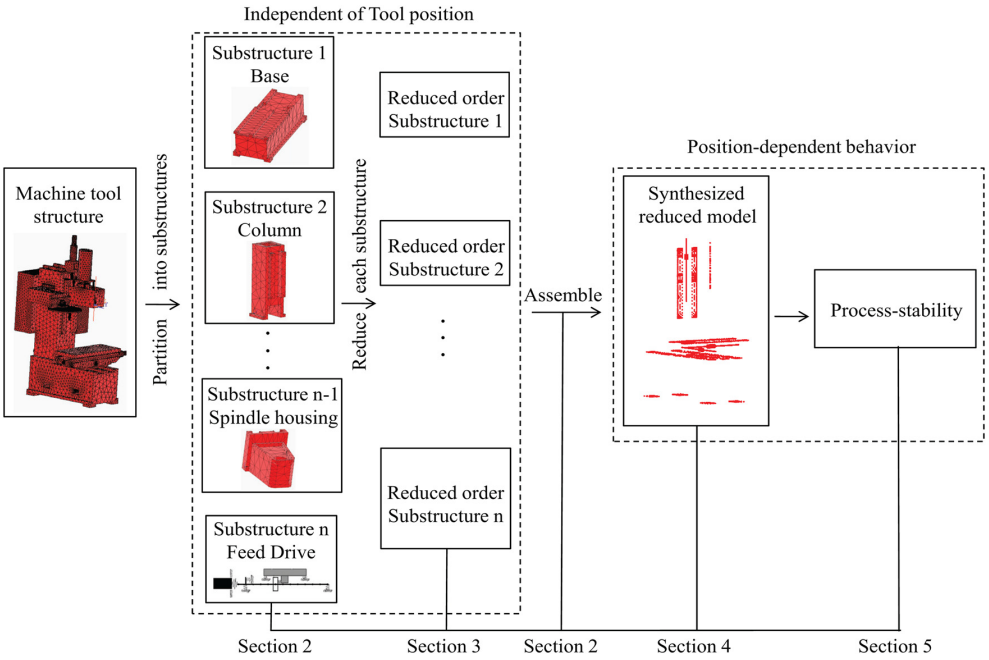


Figure 1.9: Machine center pose-dependent dynamics simulations. [82]

Manufacturing process simulation offers various complementary advantages compared with online monitoring and offline measurement. Firstly, online monitoring places high demands on the locations of sensors and the working environment. During the milling process, the cutting tool rotates and interacts with the workpiece, thus bringing great difficulties to the setup of accelerometers. At the same time, the cutting environment with metal chips and coolant cannot meet the high requirements of many sensors [87]. On the contrary, advanced finite element models can comprehensively simulate tool vibration [88], and chip formation mechanism [83]. Secondly, the setup of specific sensors may bring negative impacts on the performance of the product. During composite curing, the Fiber Bragg Gratings (FBG) sensors used to measure the temperature and degree of cure will be part of the workpiece after curing, thus bringing inevitable influences to the mechanical properties of the workpiece [89]. FE software tools allow a complete simulation of the internal properties and thus can be the complementary

information for monitoring data.

However, manufacturing simulation techniques still have some noteworthy limitations. Because of the complexity of the manufacturing process, simulation systems generally focus on the primary mechanism of the process, thus leading to inevitable assumptions, simplifications and approximation [82]. For example, during the simulation of machine tool dynamics, the FE models have to make assumptions about the joints by experience because it is difficult to obtain the interfaces and joint parameters between components [81]. For the cutting deformation simulation of aircraft structural parts, the initial residual stress inside the stock workpiece is unpredictable, thus the distribution of stress is generally designed based on experience, which brings inevitable inaccuracy in predicting the final deformation [90].

Another limitation of manufacturing simulation is the trade-off between computational cost and model accuracy. Theoretically, numerical simulation can output high-fidelity solutions, which require expensive computing resources and time. For example, the FE model of a machine tool has a very large order, typically 1,000,00 degrees of freedom, which brings time-prohibitive simulation . Therefore, it is impossible to build the pose-dependent dynamics analysis model using full simulation. A practical solution is reduced order FM modelling which seeks the balance between fidelity and computational cost by reducing the degree of freedom of the model [82]. Another example is the composite part curing shown in Fig. 1.10. The computation time and cost will significantly increase with the higher fidelity, and it will take several hours or days to conduct the complete 3D thermo-chemical analysis of the wing skin of the Boeing 787 [85]. The balance between fidelity and cost reflects the fact that in data-driven MPM problems, the generation of data should depend on the accuracy and quantity of data required for the purpose of modelling.

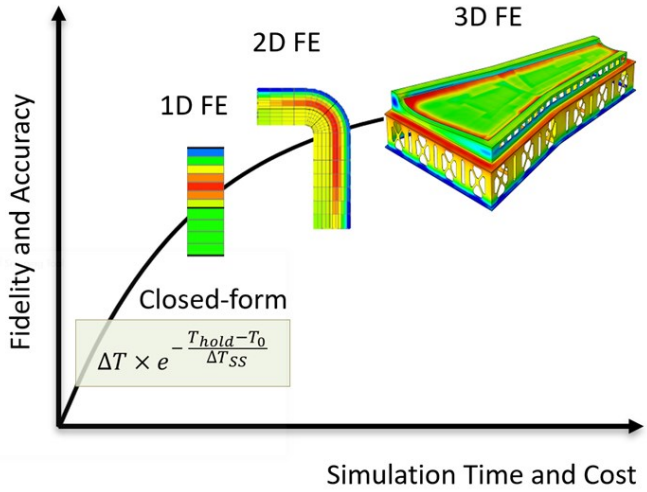


Figure 1.10: Trade-off between simulation time and accuracy for thermo-chemical analysis of composite parts. [85]

In summary, the simulation data can provide a complete representation of the manufacturing process without restrictions on experimental conditions, but there exist inevitable assumptions, simplifications and approximations about the manufacturing pro-

cess. Besides, the trade-off between computation accuracy and cost means that simulation data can be generated with different fidelity based on different purposes of the application scenarios. High-fidelity data and low-fidelity data can play different roles in data-driven manufacturing predictive modelling.

1.2.3.2 Data driven modelling methods

Data-driven modelling technologies have been applied in manufacturing predictive modelling in the 20th century, even before the concept of smart manufacturing [91]. The development of modern information technologies has breathed a new lift into data-driven modelling. Early data-driven studies used statistical approaches to analyse the effects between parameters. For example, the Response Surface Method was proposed to investigate the relationship between several explanatory variables and one or more response variables [92]. This method was used early in chemistry and biology, and later also applied in tool wear prediction, surface-roughness modelling [93], and other manufacturing scenarios [91]. Another related approach is the surrogate model, also called metamodel or emulator [94], which refers to constructing an approximation model of collected data from experiments or simulations instead of the engineering process [95]. For example, Gustave et al. [96] built a surrogate model for additive manufacturing to predict the melt pool depth under the given laser power, scan speed, and laser beam size combination. Despite the different names, the above two approaches essentially belong to data-driven modelling approaches.

This section will first present two of the representative supervised machine learning methods that are widely used in data-driven MPM, namely **neural network** and **Gaussian process regression**, and then introduce **deep learning** based modelling methods. The characteristics and limitations of these methods are also discussed.

(i) Neural network based manufacturing predictive modelling

Neural Network (NN) is widely used in data-driven modelling, especially for regression problems with low dimensional inputs [97]. A standard neural network consists of the input layer, multiple hidden layers and the output layer. These layers are connected by trainable weight parameters, and the non-linear activation function of the neurons enables the neural network to fit non-linear functions with high accuracy. After obtaining labelled data, the features are fed into the network to perform forward matrix computation for obtaining the predicted label. The loss function can be calculated by comparing the predicted labels with the actual labels, and the network parameters can then be continuously updated by minimising the loss function using gradient descent [98].

For the fully connected neural network, the number of trainable parameters will increase exponentially as the input dimension increases. More parameters will enlarge the hypothesis spaces of the model and bring significant difficulties to model training. Therefore, the fully connected neural network is more suitable for tasks with low input dimensions. For example, As shown in Fig. 1.11, Postel et al. [99] proposed a neural

network to predict the relationship between the input cutting parameters (spindle speed and cutting depth) and the unknown cutting coefficients, including the linear tangential and radial cutting coefficients. In this case, the label is not the direct output of the network layer, but the stability results predicted using the network output. Since the stability prediction is non-differentiable, this neural network cannot be trained using the gradient descent method. The parameters of the network were updated using the genetic algorithm to reduce the loss between the predicted stability results and the experimental results. Since this neural network has only two-dimensional input and one hidden layer, the model was trained with only 40 labelled cuts (21 stable and 19 unstable). Ramezankhani et al. [26] trained a neural network to predict the temperature thermal lag for composite part manufacturing. This problem is a regression task, and the input feature consists of five-dimension parameters of the curing process. In the experiments, 44,000 labelled data were generated using RAVEN software to train the neural network. In summary, the amount of required labelled data increases significantly with the parameter complexity of the modelling task.

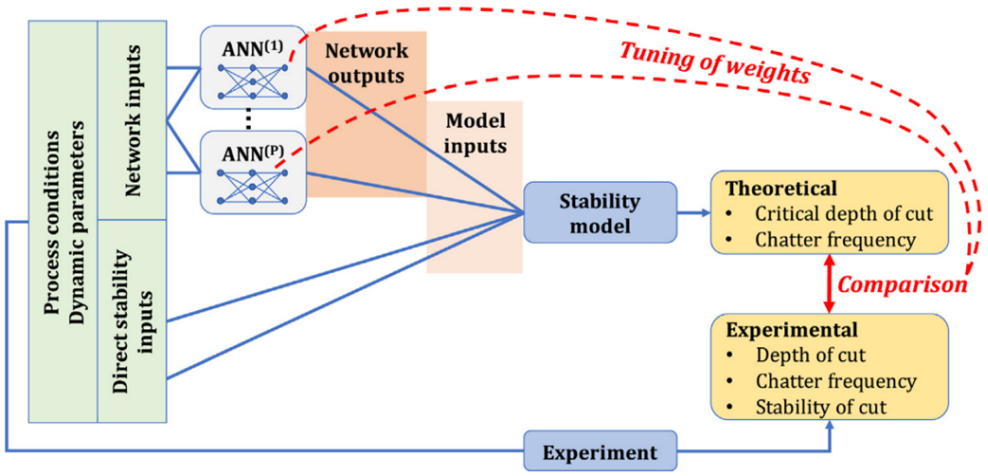
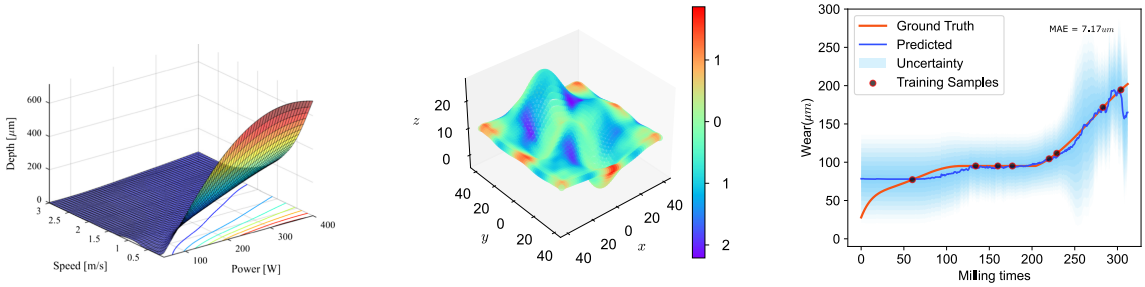


Figure 1.11: Illustration of neural network for milling stability prediction. [99]

(ii) Gaussian process regression

Gaussian process (GP) regression is a non-parametric probabilistic regression model that is widely used for low-dimensional few-sample regression problems in manufacturing processes [17]. For a given labelled dataset, GP can learn the joint distribution and inference the conditional distribution of the samples to be predicted. As a probabilistic model, GP can provide both the predicted means and the uncertainties, which can help to quantify the reliability of predicted results [100]. The predicted means of GP is consistent with the kernel least squares method under L-2 regularisation, namely kernel ridge regression. Compared with the randomness of the neural networks, GP can always provide stable prediction results. Note that, the computation complexity of GP is $\mathcal{O}(N^3)$(N is the number of training samples), so a large dataset will significantly increase the computing cost of GP. Fig. 1.12 show the application of GP in melt pool depth prediction [96], surface reconstruction [101] and cutting tool wear prediction.



(a) GP for melt pool depth. [96] (b) GP for surface reconstruction. [101] (c) GP for tool wear prediction.

Figure 1.12: Gaussian process regression in manufacturing applications.

(iii) Deep learning based manufacturing predictive modelling

Many modelling tasks in the manufacturing process involve high-dimensional data inputs, namely a very large number of input dimensions. For example, the monitoring signals with high sampling rates mean a high-dimensional input vector, the time series [102]. The high-resolution simulation data of a workpiece is normally represented by a mesh grid with thousands of nodes. As shown in Fig. 1.13(a), traditional machine learning models such as GP and NN are technically not applicable to process high-dimensional data because of their insufficient representation ability. Therefore, engineers have to generate and select different high-level features from the original high-dimensional data to enhance the performance of traditional machine learning models. For tool wear prediction, different signal features can be generated and selected from the original monitoring signals, e.g., time domain features such as mean square, variance, peak coefficient, empirical mode decomposition, and frequency domain features including Fourier analysis and wavelet analysis [103]. Conceivably, the complex feature selections of traditional machine learning tasks cannot match the requirements of modern data-driven manufacturing predictive modelling.

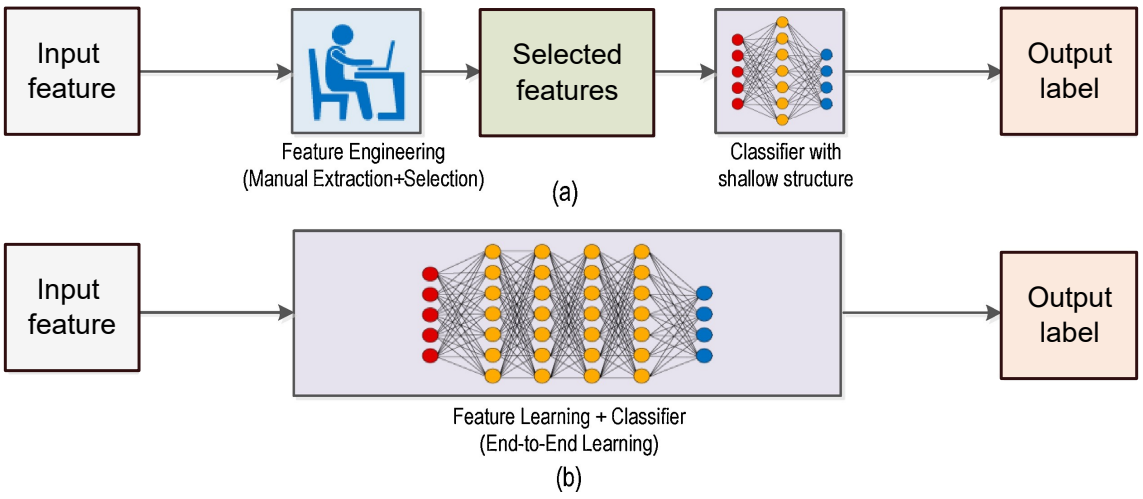


Figure 1.13: Comparison between two techniques: a) traditional machine learning, b) deep learning. [40]

Deep learning, also a subcategory of machine learning, enables automatic feature learning through more hidden layers and more expressive structures, thus can handle

complex high-dimensional data [40]. As shown in Fig. 1.13(b), the multiple hidden layers of the deep neural network continuously pass the primary features of each layer to the more abstracted representation of the later layer, and the fully connected layers at the end will play the role of classification or regression using the high-level features. The advanced abstract representability enables deep learning to explore the correlation between input and out from the massive amounts of data [40]. Various deep learning frameworks have been proposed for different data structures and task requirements, such as Deep Convolutional Network [104], Long and Short Term Memory network [105], deep autoencoders etc [106]. These methods have led to new directions for manufacturing predictive modelling.

Fig .1.14 show the AlexNet model for machining state detection proposed by Rahimi et al. [107]. The task is defined as predicting the stability of the milling process based on the monitoring data of the microphone. Therefore, the monitoring data is defined as the input feature, and the cutting state classification is the corresponding label. The combination of the input signal and the output cutting state constitute one labelled data. AlexNet is a famous convolutional neural network framework for image recognition problems.The continuous monitoring signal was segmented into smaller windows and the time-frequency spectrum of each window with the size of 400×52 was defined as the input of the model. The output is the probability of the five-dimension predefined machine stages, namely air cutting, chatter, the entrance of cutting, the exit of cutting, and stable state. This model can achieve a superior prediction accuracy of 98.90% compared with the traditional detection method (59.58%). Meanwhile, the authors designed 464 cutting tests to collect 53,884 milling labelled data. The experimental results do show the effectiveness of the deep learning method, but it is prohibitively expensive to collect such an amount of chatter data in real manufacturing scenarios.

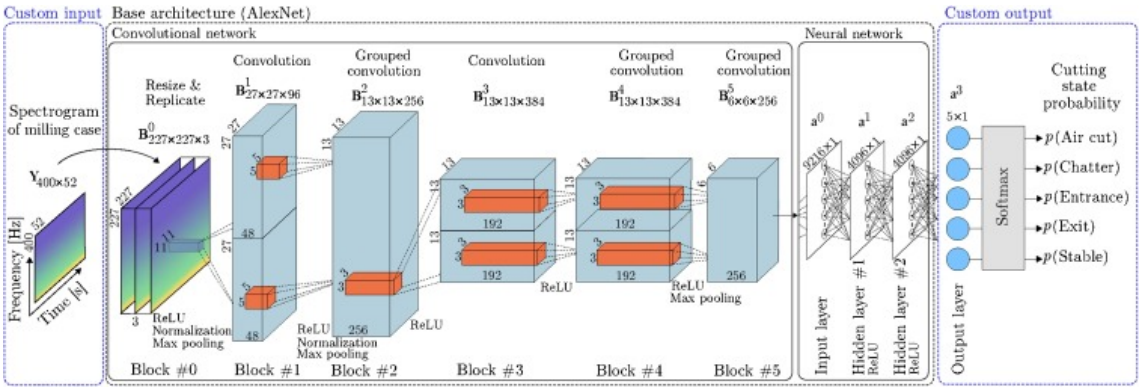


Figure 1.14: AlexNet deep learning model for machining state detection. [107]

As shown in Fig .1.15, Humfeld et al. [18] from Boeing proposed an LSTM network to predict the time-temperature histories of composite tools. The model aimed to investigate the relationship between the curing parameters and the corresponding temperature series in the workpiece for further process optimisation. The air temperature profile is the input feature, and the temperature histories of the part are the output label to be predicted. Because of the high-dimensional input and output temperature profiles, the LSTM model requires sufficient labelled data to achieve a high prediction accuracy. In this case, 100,000 simulations were conducted by a commercial FE software, RAVEN.

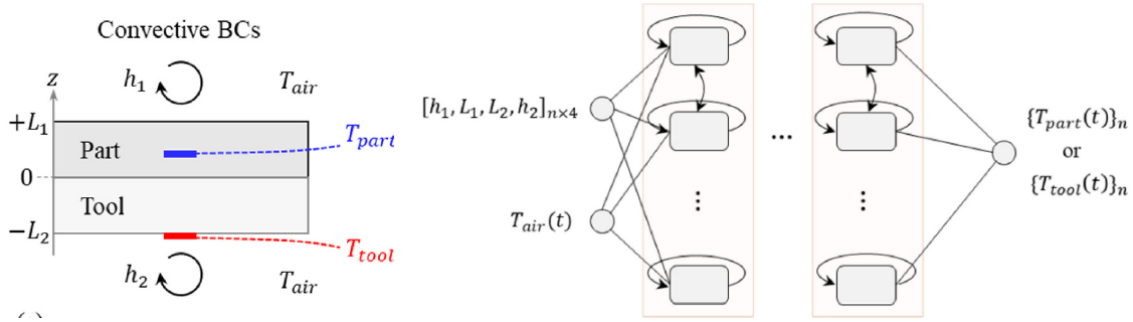


Figure 1.15: LSTM Neural Network to predict time-temperature histories of composite workpiece and tool. [18]

As reviewed above, the strong representability of deep learning facilitates the modelling and mining of complex manufacturing processes. However, the performance of deep learning relies on a large number of labelled data, which becomes an inevitable challenge for manufacturing applications.

1.2.4 Advanced modelling techniques for data-scarcity scenario

Although various data-driven modelling methods have been developed for manufacturing predictive modelling, the the high cost and difficulties in data labelling restrict its further development and applications. Establishing data-driven models with limited labelled data is an inevitable challenge for the development of smart manufacturing. This Section will review several advanced modelling techniques for data-scarcity scenarios, **including active data generation, transfer learning, and data-physics combination.**

1.2.4.1 Active data generation

Although many studies have shown that the distribution of samples has a significant impact on the performance of data-driven models [108, 109], most manufacturing predictive modelling studies had not concerned with the generation of the labelled data set. This Section will introduce existing and potential techniques for the active generation of the labelled data, which means actively determining which samples to label. The aim of active labelling is to ensure the information requirements of data-driven modelling with as few samples as possible. As shown in Fig. 1.16, active labelling should be considered from two perspectives. **Initial active labelling** means to determine the distribution of data before performing experimental labelling or simulation labelling. **Iterative active labelling** refers to querying new labelled data after training the model with previous labelled data.

(i) Initial active labelling: data sampling

Initial active labelling aims to find a subset from the potential unlabelled data pool

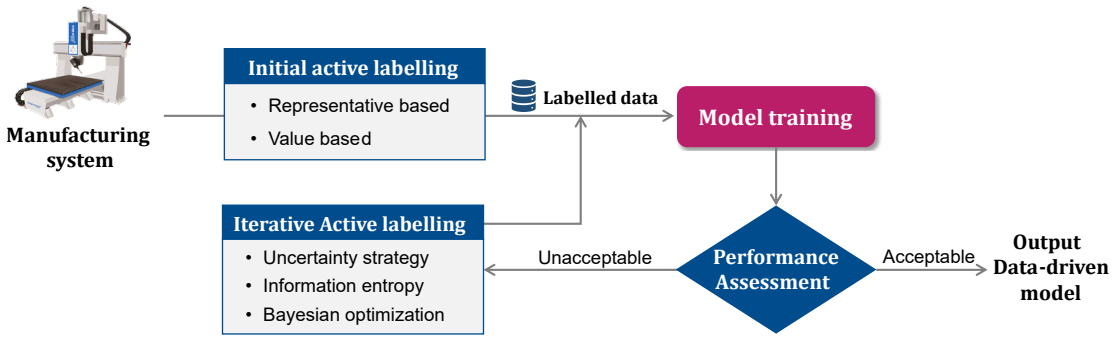
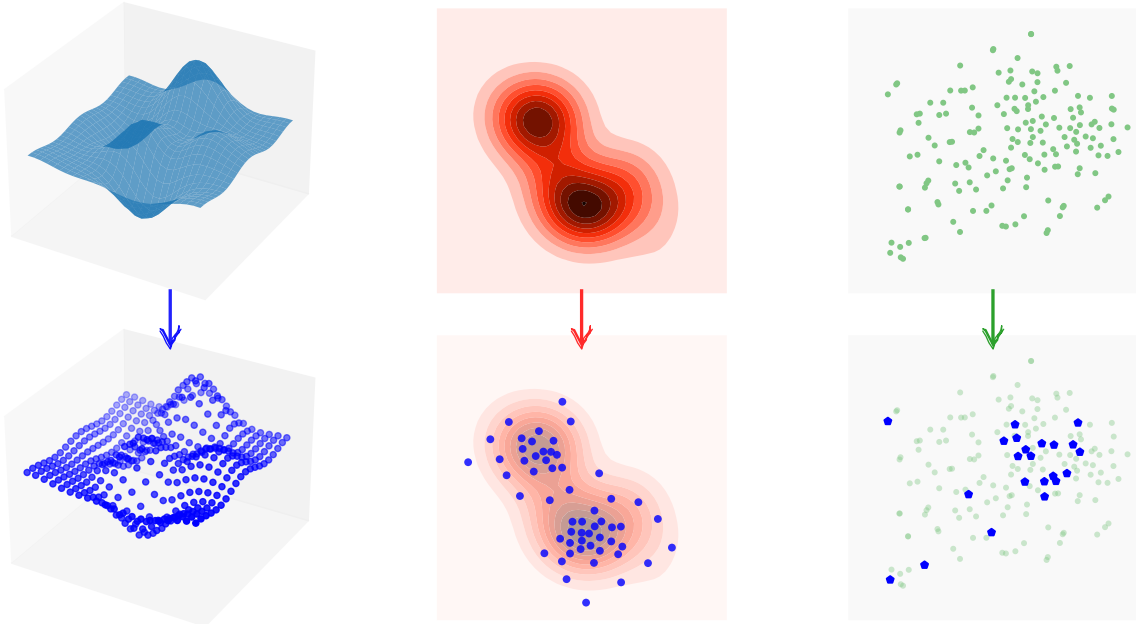


Figure 1.16: Two stage active labelling framework.

for labelling. Therefore initial active labelling becomes equivalent to selecting the initial unlabelled dataset. This problem here can be treated as a data sampling problem, which is a statistical analysis technique widely used in many fields [68]. Before the concept of smart manufacturing, the initial active labelling problems have already been investigated in Design of Experiments [110] and meta-modelling field [111]. This Section will review the potential solutions of initial active labelling techniques from multiple research fields, including sampling from feature space, distribution and discrete data, as shown in Fig. 1.17.



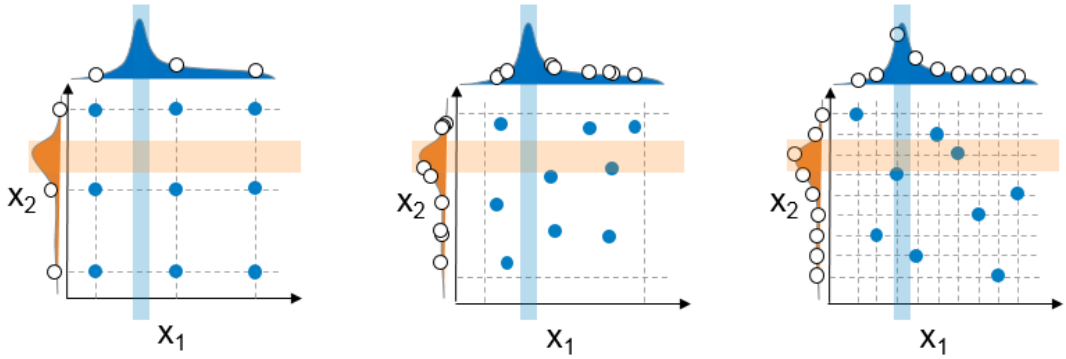
(a) Sampling from feature space. (b) Sampling from distribution. (c) Sampling from discrete data.

Figure 1.17: Three sampling scenarios.

The most intuitive solution of initial active learning is the **representative sampling**, which samples a coreset that represents the distribution of the total data [112]. Coreset selection is attracting immense attention in accelerating computation, reducing labelling effort, mining informative patterns and many other aspects [108]. From the perspective of combinatorial optimisation, selecting an optimal training coreset from a potential data pool can be attributed to a submodular function maximisation problem that appears

in various applications, including sensor placement, and multi-document summarising, feature selection and so on [113]. Representativeness is the primary objective for the coreset selection problem, where the selected coreset is expected to represent the Probabilistic Density Function of the potential total dataset.

The representativeness-based coreset selection strategy depends on how the potential data pool is defined. If the entire potential dataset is given by defining the feature space (Fig. 1.17a), uniform sampling, Hammersley random sampling, or curvature-based sampling are practical representativeness-based sampling methods [114]. For multidimensional features, Latin hypercube sampling and Sobol sampling can ensure representativeness while balancing the uniform and random properties. Note that, most design of experiments and meta-modelling problems can be defined as sampling from the given feature space. Fig. 1.18 show the illustration of uniform sampling, random sampling and Latin hypercube sampling, respectively. The uniform sampling and Latin hypercube sampling can maintain the representativeness of each dimension, which also restrict their application in high-dimensional sampling problems.



(a) Uniform sampling. (b) Random sampling. (c) Latin hypercube sampling.

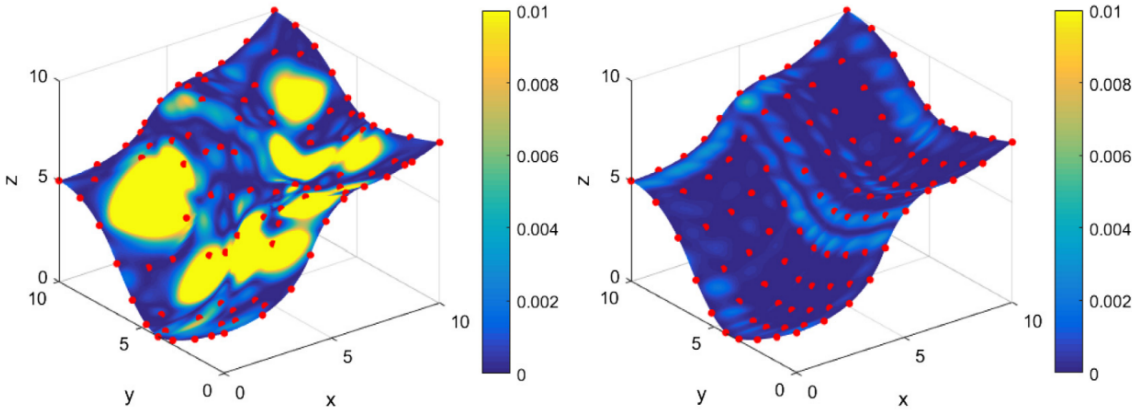
Figure 1.18: Sampling from given feature space.

If the dataset is defined by the underlying distribution (Fig. 1.17b), probabilistic sampling methods, including Markov Chain Monte Carlo or Gibbs sampling, can provide reasonable samples to approximate the distribution [17]. In practice, most potential datasets exist in the form of a finite amount of unlabelled data without predefined distribution information (Fig. 1.17c). Therefore, clustering becomes the most simple but reasonable strategy to select the representative initial unlabelled dataset [112, 115]. Take k-means clustering as an example, suppose the potential dataset N is partitioned into n_T observation groups as $\{N^1, \ldots, N^{n_t}\}$. The objective function is then defined as minimising the within-cluster sum of distances:

$$\min_{T} \sum_{i=1}^{n_t} \sum_{\mathbf{x} \in N^i} \|\mathbf{x} - \mu_i\|^2 \tag{1.1}$$

where $T = \{\mu_1, \ldots, \mu_{n_t}\}$ is the optimised representative unlabeled dataset, and μ_i is the sample that represent the group N^i. Therefore, the dataset T can be actively labelled to construct the initial labelled dataset $\mathcal{D}_T$.

Apart from representativeness-based sampling, adaptive sampling places more points in regions of interest by learning the information from the meta-model [116]. Curvature-based sampling for surface reconstruction is a classical adaptive sampling example, where a higher density of samples is assigned in the region with higher curvature [117]. Fig. 1.19 shows the error map of the reconstructed surface under Latin hypercube sampling and curvature-based sampling, where adaptive sampling can achieve a high reconstruction accuracy. Although the idea of adaptive sampling is reasonable and effective, there is a lack of a general method to quantify the region of interest, and most existing research relies on task-specific indicators [116], such as the curvature of surfaces.



(a) Latin hypercube sampling. (b) Curvature based sampling

Figure 1.19: Different sampling methods for surface reconstruction.

(ii) Iterative active labelling: active learning

After training a data-driven model with the initial coreset, iterative active labelling can improve the performance of the model by querying new promising samples, that is what defines the active learning technique [118]. The criteria definition for active queries and the number of queries are the primary focus of active learning. The general representation of the iterative active labelling is

$$\max_{x \in N \setminus T} \Delta_{\mathcal{A}}(x \mid T), \tag{1.2}$$

where $\Delta_{\mathcal{A}}(x \mid T)$ means the marginal gain of the sample x for the given subset T and data-driven algorithm $\mathcal{A}$.

The query criteria can be the representativeness when generating the initial coreset, that is, to exploit the data structure of unlabelled data to find the target samples and improve the distribution of the coreset [119]. Another query criterion is informativeness, which measures the ability of an instance in adding new information or reduce the uncertainty of a statistical model [120]. For example, Query-by-Committee chooses samples that result in maximum disagreement amongst an ensemble of basis data-driven models [121]. Uncertainty sampling selects the instances with the maximum uncertainty, thus the new dataset will train a more reliable model by reducing uncertainties [119].

A similar concept in design of experiment is Sequential Experimental Design. As shown in Fig. 1.20, it consists of space-filling sampling or adaptive sampling, which are similar to representativeness-based active learning and informativeness-based active learning. The difference is that Sequential Experimental Design generally focuses on sampling from feature space (as shown in Fig. 1.17a), while active learning in the machine learning field targeted at sampling from data pool (as shown in Fig. 1.17c).

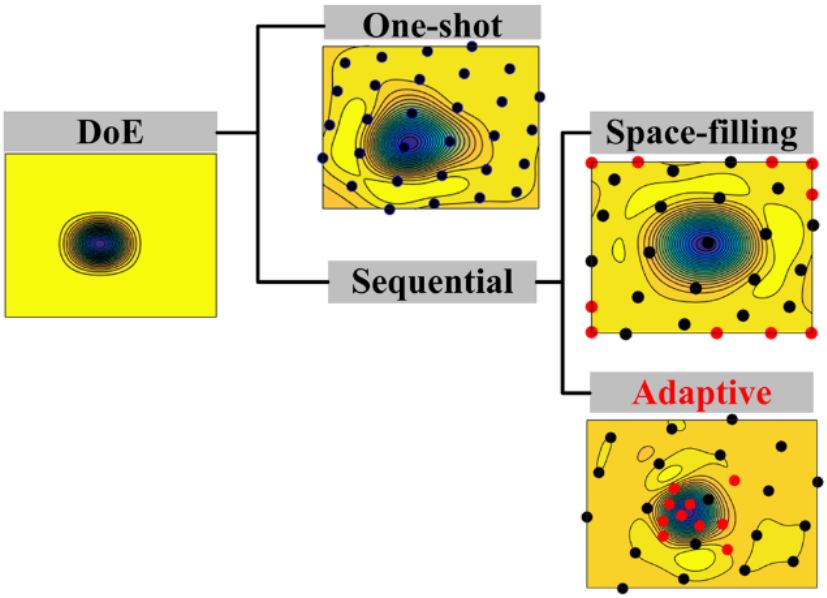


Figure 1.20: Sampling categories in Design Of Experiments (DoE). [116]

The above-mentioned active learning techniques have been widely used in manufacturing predictive modelling problems. Leco et al. [122] proposed an active learning algorithm for robotic machining error modelling that made online inspection decision based on the prediction confidence of the current model . Hughes et al. [123] presented a risk-based active learning solution for structural health classification, in which the promising class-label information was queried by the expected value for each incipient data point. Similarly, Arellano et al. [124] built a Bayesian Convolutional Neural Network for online tool condition classification, that could determine whether the incoming data should be labelled. Those applications demonstrate that active learning strategies could achieve satisfactory model performance with smaller training dataset.

1.2.4.2 Transfer learning

To reduce labelling consumption, transfer learning, which can improve the performance of learning by leveraging rich labelled data from related domains [125, 126], has drawn much attention recently in multiple machine learning fields including natural language processing, and image recognition [127, 128]. The basic illustration of transfer learning is shown in Fig. 1.21. The objective is to improve the learning of the target task using the knowledge from similar tasks, namely the source tasks, so that the target task only needs a few labelled data. From a probabilistic point of view, transfer learning methods

refer to adapting distribution discrepancy between different tasks so as to extract the common knowledge across domains [125].

This Section will introduce two transfer learning techniques widely used in manufacturing predictive modelling, namely **covariate shift adaptation** and **parameters transfer** .

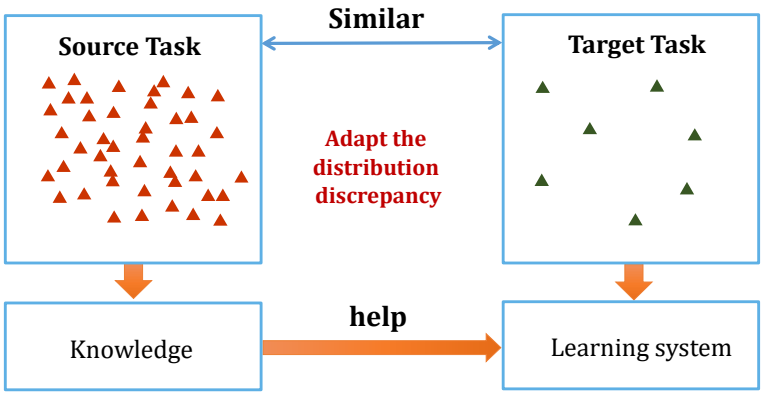


Figure 1.21: The illustration of transfer learning concept.

(i) Covariate shift adaptation

Suppose $\mathcal{D}_s = \{\mathbf{X}_s, \mathbf{y}_s\}$ as the auxiliary data from the source domain and $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{y}_t\}$ as the direct labelled data from the target domain, where the $\mathbf{X}_s$ and $\mathbf{X}_t$ are input features of two domains, $\mathbf{y}_s$ and $\mathbf{y}_t$ are output labels. Since the two datasets are generated from different domains, they cannot be processed consistently with a standard machine learning algorithm. The purpose of transfer learning is to extract common and transferable information through a customised learning procedure. According to different distribution assumptions, transfer learning problems can be categorised into covariance shift, conditional shift and prior shift [129]. Covariance shift classification is the most well-investigated transfer learning configuration [126], in which the marginal distribution of features across domains are assumed to be different while conditional distribution remains consistent, namely $p(\mathbf{y}_s \mid \mathbf{X}_s) = p(\mathbf{y}_t \mid \mathbf{X}_t)$ and $p(\mathbf{X}_s) \neq p(\mathbf{X}_t)$. The failure diagnosis and cutting condition classification problems are classical covariance shift scenarios in manufacturing [130, 131]. The primary challenge of covariance shift lies in adapting the distribution difference between $p(\mathbf{X}_s)$ and $p(\mathbf{X}_t)$. If there is underlying common knowledge in the two datasets, a knowledge transformation operator φ can be learned by minimising the distribution between the embedded representation of two datasets as:

$$\min_{\varphi} \text{dis}\left[p\left(\varphi\left(\mathbf{X}_s\right)\right), p\left(\varphi\left(\mathbf{X}_t\right)\right)\right] \tag{1.3}$$

where dis[] is the distribution distance involving Maximum Mean Discrepancy (MMD), Kullback-Leiblec divergence, and Wasserstein distance [132]. The specific form of operation transformation φ is also diverse, which can be subspace, neural network, and low-rank decomposition [126].

Xu et al.[133] proposed a digital-twin-assisted fault diagnosis method using deep transfer learning to realise fault diagnosis both in the virtual space and the physical space, the procedure is shown in Fig .1.22. The process monitoring data in the two environments have different distributions, while the state classification results share the same label space, namely four classes, 'Good', 'Warning', 'Watch' and 'Fault'. As defined above, this is a classical covariance shift scenario which can be solved by minimising the distribution difference between the features of two domains. Therefore, Xu et al.[133] added an adaptation layer between the feature extraction and classification layer and then minimising the Maximum Mean Discrepancy distance to train the feature extraction layers.

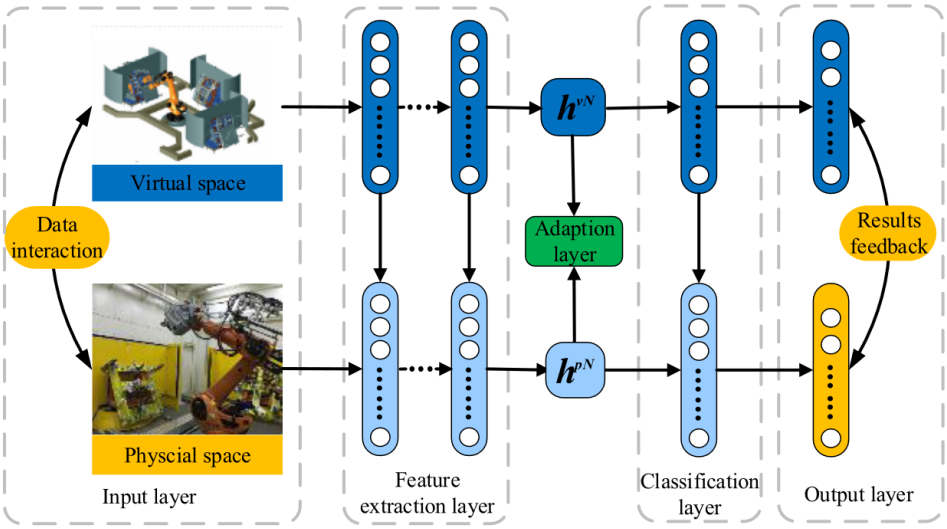


Figure 1.22: Digital twin-assisted fault diagnosis using deep transfer learning. [134]

The idea of learning invariant features using deep neural networks has been well explored in the machine learning field. Tzeng et. al [135] first introduced Maximum Mean Discrepancy to deep networks to learn a representation that is both semantically meaningful and domain invariant. Long et al. [136] extended deep convolutional neural networks to domain adaptation problems and explored the transferability of different layers in the networks. Wu et al. [137] proposed the Geometric Knowledge Embedding method to learn underlying geometric structures to minimise the Maximum Mean Discrepancy in a graph convolutional network. These deep learning methods have also been generalised to heterogeneous domain adaptation [127, 138], multi-source domain adaptation [139] and other settings.

As shown in Fig. 1.23, a similar idea was also applied in the tool wear state prediction task, where the monitoring data from different cutting conditions were defined as source data and target data respectively. Theoretically, both the marginal distribution $p(\mathbf{X})$ and conditional distribution $p(\mathbf{y} \mid \mathbf{X})$ across two cutting environments should be adapted because different cutting conditions will lead to different signals and tool wear values. Their research adapted the covariate shift using an adversarial framework, where the feature extractor had to learn the common feature from two domains so as to maximise the domain discrimination loss while minimising the classification loss.

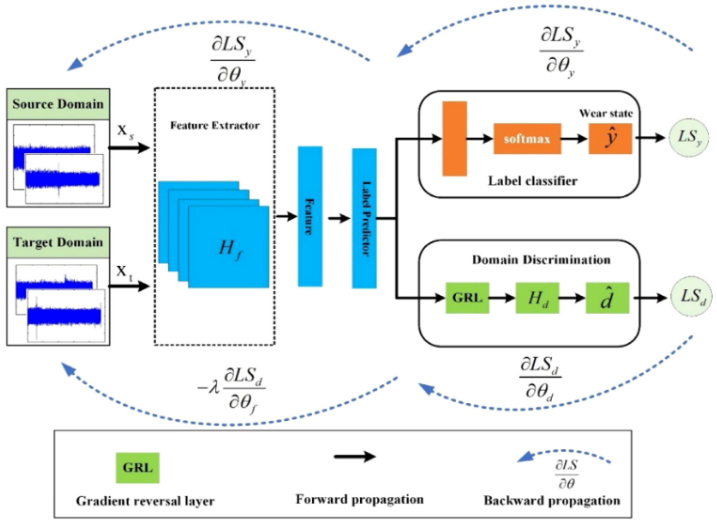


Figure 1.23: Adversarial domain adaptation transfer learning model for tool wear state prediction. [133]

Covariate shift classification problems have been well investigated in both the machine learning field and the manufacturing field. However, there are many regression tasks involving transferable scenarios in manufacturing engineering, which have the characteristic of conditional shift $p(\mathbf{y}_s \mid \mathbf{X}_s) \neq p(\mathbf{y}_t \mid \mathbf{X}_t)$. Chapter 3 will focus on the distribution adaptation for regression tasks under conditional shift.

(ii) Parameters transfer

Parameters transfer methods refer to the reuse and fine-tuning of parameters that are pre-trained in the source domain based on the assumption that similar tasks may share similar network parameters [140]. Yosinski et al. [141] first revealed that the features extracted by layers of the network may be generic or task-specific, and parameters pre-trained by the source dataset usually perform better than random initialisation. After that, this technique had been widely adopted in both industrial and academic fields [106]. Many large-scale networks (such as Resnet [1] and Alenet [2]) with pre-trained parameters were reused as backbone models for other similar problems to reduce data requirements and accelerate model training.

As shown in Fig. 1.24, Ramezankhani et al. [26] proposed a parameters transfer scheme for composites curing, where the input was the five-dimension parameters of the curing process, and the outputs were thermal lag and exotherm, that have been proven to be key metrics for the quality of composites part. The source data consisted of 44,000 simulated labelled data under one-hold curing cycles, and the target data included only 500 labelled data under two-hold curing cycles. Firstly, a full-connected neural network with six hidden layers and ten neurons per layer was established with source data. Then the transfer-ability of each layer was investigated comprehensively by freezing the pre-trained parameters. The authors found that increasing the number of frozen layers could

[1] https://pytorch.org/vision/main/models/resnet.html
[2] https://pytorch.org/hub/pytorch_vision_alexnet/

restrict the flexibility of the model, which led to high prediction errors. The experimental results shown that freezing one layer could achieve the best performance of 1.91 K of MSE.

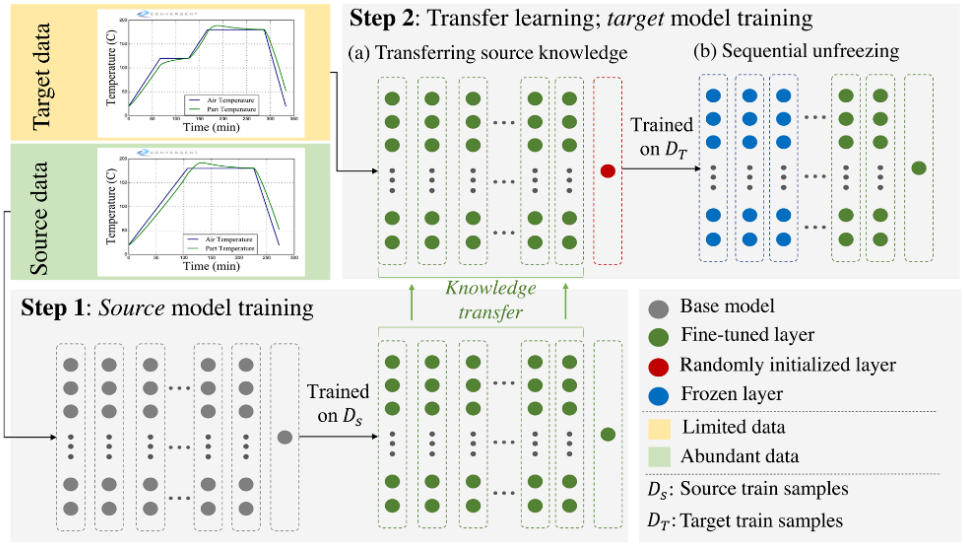


Figure 1.24: The cross-process transfer learning method in composites curing. [26]

A similar fine-tuning solution was also developed by Postel et al. [19] for milling stability prediction. The basic model was a deep neural network with a softmax function at last to output classification results. The input of the model consisted of cutting parameters and tool dynamics, and the output of the model was a binary value to represent stability or chatter. As shown in Fig. 1.25, 14850 simulation samples were defined as the source data to train the source model. After that, part of the model parameters was fine-tuned on a small experimental dataset with 10 100 samples. The author investigated the influence of the training set size on the final performance of the target model. The experimental results shown that the transfer learning model could significantly reduce the data requirements compared with the deep neural network.

Although the parameters transfer method is convenient to implement, Li et al. [104] demonstrated that fine-tuning the pre-trained network via minimising the empirical Mean Square Error based on limited target data might suffer from the risk of overfitting. This drawback was also summarised by Postel et al. [19] in terms of the shape of the experimental milling stability lobe. According to the milling stability theory, the stability boundary should follow strict lobe characteristics, which is a basic mechanism constraint. However, insufficient samples of transfer learning will lead to significant randomness in the stability boundary and loss of physical interpretability.

1.2.4.3 Data physics combination

Manufacturing industry has a much longer history than data-driven modelling, and pioneering engineers and researchers shave broad and in-depth understanding of the

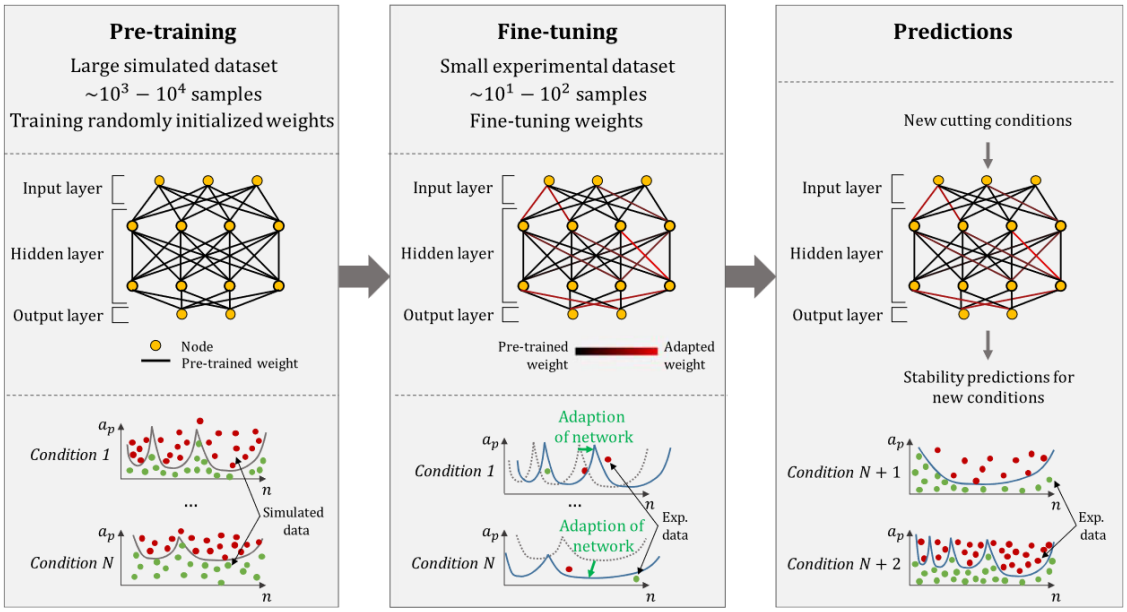


Figure 1.25: The cross-process transfer learning method in composites curing. [19]

physics and accumulated prior knowledge of various manufacturing processes [46]. Although the knowledge may not be accurate enough to describe the complex real manufacturing process, it is certainly unwise to abandon these priors in pursuit of data-driven models. Despite various related concepts being developed, such as physics-hybrid learning [46] and physics-guided learning [85], the basic strategies of data-physics combination can be categorised into **designing the loss function** and **designing the model structure**.

(i) Design physics-informed loss function

Data-driven predictive models, such as neural networks, aim to learn a parametric hypothesis that satisfies the observation of the training dataset. The inconsistency between the predicted results and the true labels is defined as the loss function to provide direction for the gradient-descent optimisation of the model [142]. The inconsistency between the predicted results and the priors could also be included in the loss function as a punishment to enforce the physics priors [143].

The physics priors here include strong priors and weak priors. Strong priors refer to the strict mathematical formulas that can describe global and local phenomena or properties in the manufacturing process, such as dynamics force models in milling operations [13], and heat transfer equations in composites curing [10]. Besides strong priors, there always exist weak priors defined as empirical knowledge in non-formula form. Since each input or output of MPM problems has an explicit interpretation, the most simple but intuitive weak prior is the feasible region of physics parameters, such as the power limit of manufacturing equipment and tool wear value limit [64]. Empirical knowledge can also provide evolution constraints of parameters or the coupling constraints between multiple parameters. For example, the remaining life of a cutting tool

always decreases along with machining times, and the degree of cure of prepreg always increases and finally approaches 100% during heating [144]. Weak priors can provide soft penalty constraints or regularisation terms to the final loss function, while strong priors can sometimes even replace the traditional loss function, especially for partial differential equations, such as Physics Informed Neural Networks [145] (Fig. 1.26) and Hamiltonian Neural Networks [146]. In practice, the loss function can be the weighted combination of predicted loss and physics loss so as to leverage both the data and physics priors.

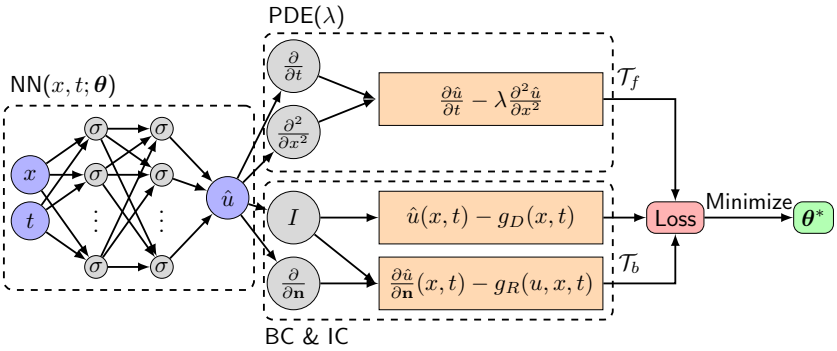


Figure 1.26: Physics informed neural networks [145]

(ii) Design model structure

Another kind of strategy focuses on designing specialised model structures that implicitly intergrating the physics priors of tasks. For example, Deep Lagrangian Networks [147] directly incorporate each part of the robot dynamics equations to construct explainable data-driven networks (as shown in Fig. 1.27). The global structure from strong priors normally requires craftsmanship and elaborate implementations, thus not very practicable. For MPM problems, the widely-existed weak priors can be integrated into feature extraction modules more easily, such as the physics-guided input module based on tool cutting force model[52], or the specialised activation function for transient thermal analysis [85].

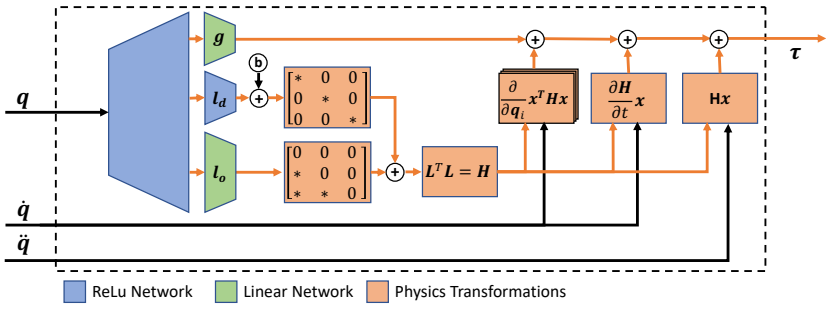


Figure 1.27: Deep Lagrangian networks [147]

Fig. 1.28 shows a neural network for composites curing temperature prediction, in which the network structure was designed with several physical priors [148]. According

to the analytical solution of heat transfer equation, the temperature has cosine function relationship with the coordinate x has an exponential relationship with t. Therefore, the sine activation function was added for x and an exponential activation function was designed for t. The two convective heat transfer coefficients h_1 and h_2 were input to the network after the multiply of x term and t term, rather than in the sample input positions. With these well-designed details, the experiments shown that the network could maintain strong generalisability compared with traditional neural network.

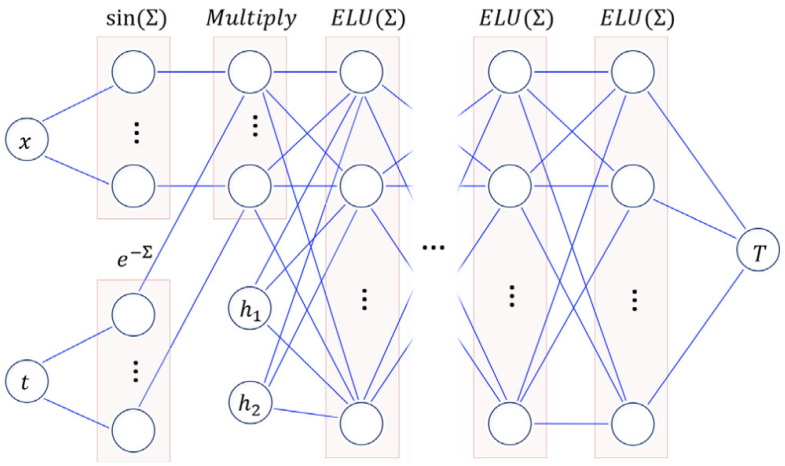


Figure 1.28: Neural network with physics-informed features for composites manufacturing. [148]

Fig. 1.29 is a mechanism-based structured deep neural network for cutting force forecasting proposed by Cheng et al. [15]. The mechanism here can also be categorised into physics knowledge. During the milling process, the cutting force generation consisted of the static part and the dynamic parts. The mechanism of the static part was integrated into the input monitoring signal, and then the processed features were passed to the dynamic part, which was a neural network. These sub-models could approximate different functions during cutting force generation, and the connected model could naturally inherit the physical interpretability and generalisation ability of the mechanism.

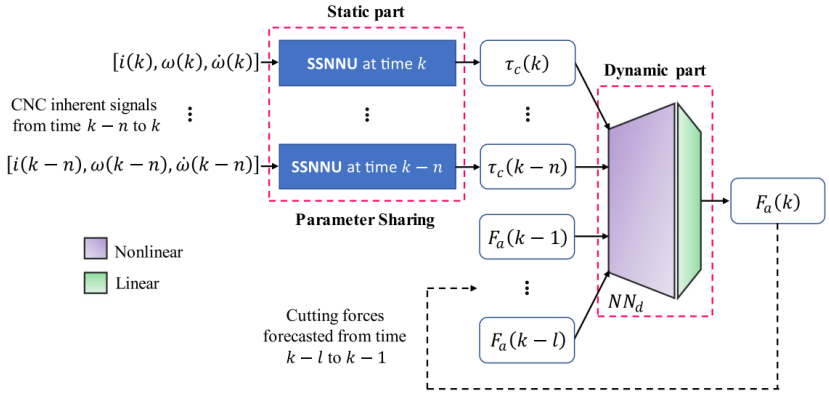


Figure 1.29: Mechanism-based structured deep neural network (MS-DNN) for cutting force forecasting. [15]

Although different data physics combination methods have been proposed and applied in manufacturing predictive modelling, they are only applicable to low-dimensional modelling problems. Physics-informed loss functions heavily rely on the automatic differentiation of output parameters on the neural input, which only exists in strict ordinary differential equations or partial differential equations, while existing model design research focuses on the low-dimensional input-output relationship. There is still a lack of investigation of how to apply prior knowledge for the high-dimensional mapping manufacturing predictive modelling problems. Many manufacturing predictive modelling problems have high-dimensional inputs and outputs in the form of fields or functions, such as the prediction of deformation fields in composite manufacturing processes [149], or stress field of workpieces during milling processes [90]. It remains a challenge to predict the high-dimensional property field of workpieces for existing data-driven modelling methods, which will be the research topic of this research as described in Chapter 4.

In summary, the challenge for modelling part property fields lies in how to represent high-dimensional complex geometries and extract low-dimensional features without increasing the complexity of the data-driven model.

1.3 Research gap, aim and objectives

In previous Sections, the basic concepts and definitions of data-driven smart manufacturing were introduced. Furthermore, the existing manufacturing process data collection and modelling methods were also presented. This Section will identify the research gap and state the main aim and objectives of this research.

1.3.1 Research gaps

Due to the expensive efforts of labelled data collection, establishing data-driven models with limited labelled data is an inevitable trend and also a challenge for the development of smart manufacturing. As reviewed above, existing MPM research has deeply investigated the different manufacturing process data and modelling techniques. In summary, existing MPM research suffers from the following research gaps, which also bring inspiration for the thesis.

(i): Passive data generation and collection: Previous research focused on how to train data-driven models based on given insufficient datasets, which means that the datasets are defined as determined and passive modelling conditions. This presupposition of passive data generation deprives us of the possibility of exploiting the data generation process actively. Despite the limitation of data size, different samples have different values for model training, and the distribution of the training data will also influence the performance of the machine learning model. Therefore, a research question is how to actively generate a more informative but smaller dataset to preserve the characteristics of the task while reducing the required amount of training data. However, ex-

isting representativeness-based sampling methods cannot capture the core samples that can reflect the characteristics of the models. There is a requirement for a new method that can actively guide the generation of labelled data by leveraging various types of information from manufacturing systems.

(ii): Insufficient modelling information: Generally, the information carried by a dataset directly determines the upper limit of performance of a data-driven model, especially when data is scarce. Therefore, it is impossible to build an accurate model with insufficient modelling information. Since the labelled data is scarce, it is necessary to compensate for the insufficient information from other available information sources, such as using transfer learning and data-physics combination. As reviewed in Section 1.2.4.2, several transfer learning methods have been developed for covariate shift problems, while the conditional shift problems widely existing in MPM problems are not well-investigated. For the existing data-physics combination research, the physics-informed loss function heavily relies on the strict form of mechanism equation and is only applicable for low-dimensional prediction. For high-dimensional prediction problems, such as the workpiece property field prediction, the complex parameterisation of data-driven models and the massive demand for labelled data remain as challenges.

1.3.2 Aim and objective

Based on the above-mentioned research gaps, this research aimed at setting up a systematic framework and developing machine learning methodologies for manufacturing predictive modelling under data scarcity scenarios. The main research objectives are described below:

- **Objective 1:** : Investigating the influence of the distribution of labelled data on the performance of data-driven modelling and developing data generation strategies considering the distribution and task-dependent values.
- **Objective 2:** : Analysing transfer learning problems in manufacturing under the conditional distribution shift. Developing transfer learning techniques for reusing knowledge from similar tasks to reduce data requirements and enhance the performance of process modelling.
- **Objective 3:** : Integrating physics knowledge to data-driven models to reduce the data requirements and increase generalisability and interpretability.
- **Objective 4:** : Applying the developed technologies for the high-dimensional property field prediction case and evaluating the modelling performance in data-scarcity scenarios.

1.4 The proposed framework for data-driven manufacturing predictive modelling under data scarcity

This project proposed a systematic guidance framework from data generation to data modelling to deal with the data scarcity problems in MPM problems. As shown in Fig. 1.30, the framework starts with the definition of three types of available information sources in manufacturing systems and then focuses on how to build data-driven models through multiple interactive operations of these three types of information.

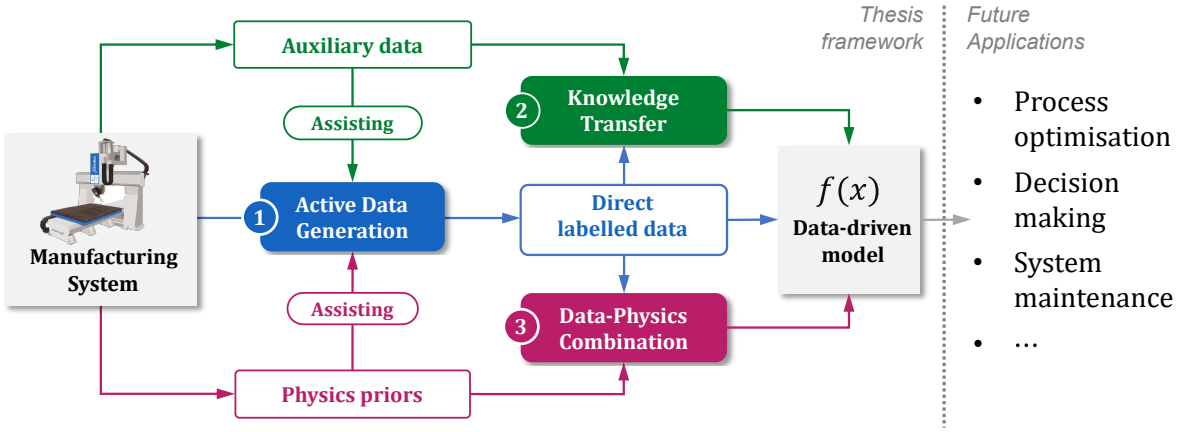


Figure 1.30: The proposed framework for data-driven smart manufacturing under data scarcity.

From the information perspective, the data-driven MPM can potentially extract information from not only task-related direct labelled data, but also the auxiliary data with potential value for process modelling, as well as the physics prior knowledge. These three information sources have different but complementary characteristics for model training. The direct labelled data here refers to the labelled data that can be directly used for training the target MPM tasks. It is the most valuable but always insufficient because of the expensive labelling processes, as reviewed in Section 1.2.3. Auxiliary data refers to the data that, apart from the direct labelled data, still benefits the training of the MPM tasks, for example, the data from similar tasks. The widely exsiting auxiliary data is less valuable but may compensate for the insufficient direct labelled data (As reviewed in Section 1.2.4.2). The physics priors, including strict formulas or weak priors, are more general and do not require special collections. As reviewed in Section 1.2.4.3, integrating these priors into data-driven models can potentially reduce the requirement for labelled data and enhance the generalisability of models.

Based on the defined three information sources, the framework also allows how to leverage the different information sources to achieve small-data modelling, including active generation of direct labelled data, knowledge transfer from the auxiliary data and data-physics combination, which correspond to the research objectives listed above.

- **Active data generation**: This part aims to generate a small but informative dataset that preserves the valuable information of the MPM task so as to reduce the required amount of training data. An aggregation-value-based sampling method

was developed based on Game theory. The special value function defined from auxiliary data and physics priors can be used to guide the sampling of the direct labelled data. Experiments demonstrated that the size of labelled data could be reduced by more than 50 % under the fixed accuracy requirements.

- **Knowledge transfer**: Transferring the auxiliary information of manufacturing systems can compensate for the insufficient target labelled data. A transfer learning method based on structured distribution adaptation was proposed for the widely existing conditional shift MPM problems. Experiments on different MPM problems shown that the proposed method could enhance the performance of the task under data scarcity by leveraging the available auxiliary data.
- **Data-physics combination**: A physics-guided low-dimensional neural operator was proposed for solving the high-dimensional workpiece property field prediction problems. Detailed experiments and analysis revealed that embedding the physics priors of the MPM problem into the data-driven model enabled more effective information extraction under the small amount of labelled data.

1.5 Thesis structure

Chapter 1 introduced the background of smart manufacturing, reviewed the related literature in data-driven MPM and summarised the aim and objectives of this thesis. Chapter 2 describes a proposed aggregation-value-based sampling method based on game theory to address the challenge of passive data generation. Experiments on several manufacturing cases demonstrated that the proposed method can sample a smaller but informative dataset to reduce the labelled data requirements. Chapter 3 presents a structured distribution adaptation method for transferring knowledge from the auxiliary data. Chapter 4 describes a proposed physics-informed neural operator by embedding the prior knowledge of MPM problems into the feature learning component of the network. Chapter 5 describes the validation of the proposed framework by a complex case study about composite part deformation prediction. The three developed techniques, namely active data generation, knowledge transfer and data-physics combination, were all applied in this case to show the effectiveness in data scarcity scenarios. Finally, in Chapter 6, the main contribution of this research is summarised, and future research work is recommended.

Chapter 2

Aggregation-value-based sampling for the generation of the direct labelled data

Vouloir c'est pouvoir!

– Tout le monde

Data-driven modelling has shown promising potential in many industrial applications, while the expensive and time-consuming labelling of experimental and simulation data restricts its further development. Since the distribution of training data influences the performance of data-driven models, designing a small but informative dataset that preserves the characteristics of the task and significantly reduce the required amount of training data [109, 108, 150]. As shown in Fig. 1.30, this Chapter will investigate **objective** 1 of the proposed framework, data generation, i.e., the direct labelled data generation with the assistance of auxiliary data and physics priors. An aggregation-value-based sampling is proposed based on the Game theory for sampling the most promising labelled data.

2.1 Introduction and challenge analysis

Generating a smaller but informative subset can potentially reduce the data labelling efforts of MPM problems. As shown in Fig 2.1, the generation of the direct labelled data can be defined as sampling a specialised subset from the sample space of the modelling task. The sample space can be defined discretely as a potential dataset where one sample refers to one data point. The aimless expensive labelling from experiments or simulations can then be replaced by directly labelling the informative subset.

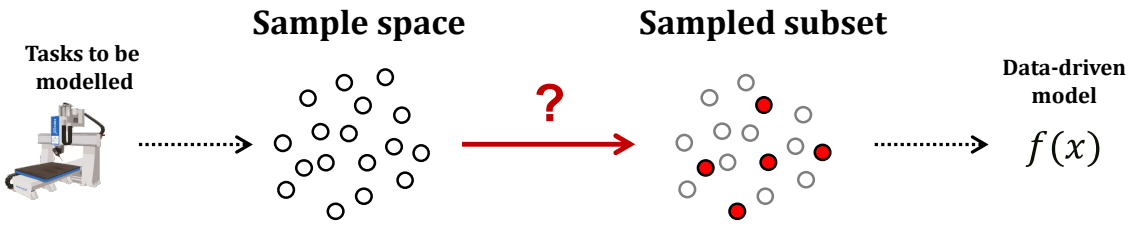


Figure 2.1: Problem definition of data sampling

Representativeness-based sampling is the most widely used sampling method, where the selected samples are expected to represent the characteristics that should be preserved, such as the probabilistic distribution or the representative data patterns. [150]. Probabilistic sampling methods aim at finding a reasonable subset to approximate the probabilistic distribution of the sample space [17]. If the sample space is given by a discretised potential dataset, clustering sampling can select some clusters as the representative subset [151]. Similarly, low-rank-based methods can select the fewest samples to preserve the representative patterns or basis for high-dimensional samples [109, 152]. An important underlying presupposition of representativeness-based sampling methods is that the representative samples should provide more valuable information for the data-driven model [108]. Although reasonable, the presupposition is insufficient because representativeness is only the indirect characterisation of the value of samples. Thus, some core samples that can reflect the characteristic of the models might not be captured. This problem is further exacerbated in highly imbalanced real-world datasets where the representative sample set, with high probability, may miss the dominant samples [151].

To directly quantify the contribution of each sample during model training, researchers

recently proposed another interesting indicator, the value of samples. The value function $v(x)$ is then defined as the function that can quantitatively measure the value of the sample x with respect to a given learning algorithm and a performance metric [153]. The first attempt at data valuation was leave-one-out (LOO) and the subsequent influence function method [154], in which a specific value was determined for each sample according to the performance difference when the sample was removed from the sample space. A fundamental limitation is that these methods can only represent the marginal gain for one specific sample set N$\backslash\{i\}$. From the perspective of game theory, training a dataset can be treated as a coalitional game, in which all data samples are players working for a common goal. Based on the cost-sharing theory, Ghorbani et al. [153] introduced Shapley value into data valuation and sample selection problems. The Shapley value of each sample was represented as the average marginal gains of all potential subsets. Then highest-ranking samples were selected as the satisfied sample set based on the standard greedy algorithm in submodular function maximisation [155, 156]. A series of improved versions and accelerating algorithms were further proposed to boost the development of Shapley value in the machine learning field [153].

Although Shapley value can provide a "favourable and fair" valuation of data, the highest value sample set sometimes reduced the data diversity, especially when the size of the sample set was small, which would lead to high generalisation error [157]. Further analysis in multiple datasets revealed the high-value samples clustering phenomenon in the feature space. The samples that are close to each other in the sample space, namely neighbouring samples, might carry similar or redundant feature information, which cannot bring a proportional contribution to the model training. Therefore, close or similar samples can only provide very few additional contributions for machine learning tasks, regardless of regression, classification or structural learning tasks [158, 159].

This means that the sum of the values of samples in the selected subset cannot represent the actual value of the subset. **Therefore, defining the actual value of a sample set considering redundant information becomes the critical challenge for sample selection problems.** This research will break the limitations of the existing representative-based and value-based sampling methods, and propose a new sampling method by identifying the redundant information and quantifying the value of the subset.

2.2 General idea of the new introduced aggregation-value-based sampling (AV4Sam)

As shown in Fig. 2.2a, existing value-based sampling methods determine the optimal sample set by maximising the sum of sample values, which leads to information redundancy and reduces the diversity of the samples [153]. To overcome this problem, this Chapter describes a proposed Aggregation-Value-based Sampling method (AV4Sam). As shown in Fig. 2.2b, to represent the values of neighbouring samples, the **Value Aggregation Function (VAF)** is constructed by aggregating the values of its neighbouring samples. Therefore, the close samples would share significant overlaps in their VAFs, which

can explicitly represent the redundant information carried by these samples. Since the close high-value samples cannot increase the 'area' of VAFs, the union of individual VAFs becomes a reasonable representation of the actual value of a sample set. Based on this, aggregation value, defined as the expectation of the united VAFs, can be the intuitive target to assess the sampling results. Maximising the aggregation value can effectively find the most contributing samples while mitigating redundant information.

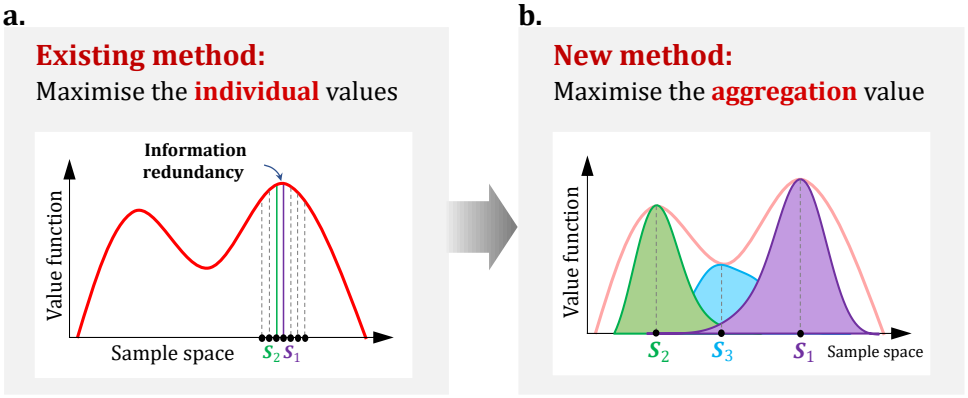


Figure 2.2: The general idea of aggregation-value-based sampling.

The implementation procedure of the proposed AV4Sam is shown in Fig. 2.3, where a value function for the sample space is established first, followed by maximising the aggregation value for sampling. The purpose of aggregation-value-based sampling is to reduce the labelling efforts for industrial applications by designing an informative but smaller sample set, thus it is less meaningful if the establishment of value function requires too much labelled data. Although the proposed method is derived from Shapley value function, the basic idea of the aggregation value can be generalised to other forms of value function as long as it is positively correlated with the real contribution of samples. Therefore, the proposed method is generalised to more practical scenarios in the case studies by introducing four value function schemes A, B, C and D which are explained below.

Scheme A: evaluate value function from direct labelled data. Sufficient direct labelled data could provide a more accurate value function but increase the labelling burden. Therefore, the case study on this scheme aims to demonstrate that the proposed method could find a better sample set rather than focusing on comparing the labelling efforts.

Scheme B: reuse value function from similar tasks. Just as transfer learning and meta-learning can utilise data from similar or relevant tasks to assist the target task, the value function from similar tasks, such as different manufacturing systems or cutting conditions, could also provide a reference for the target task.

Scheme C: reuse value function from low fidelity data. High-fidelity manufacturing process simulation is expensive and time-consuming, while simplified low-fidelity models are far more efficient. Although not accurate enough, the low-fidelity data can still provide an effective value function to select better samples for the following high-fidelity simulation.

Scheme D: define value function from prior knowledge. With a broad and in-depth understanding of the prior knowledge of various manufacturing processes, researchers and engineers can define specific value functions according to the sample requirements. From this point of view, AV4Sam can be extended to various engineering-based sampling scenarios, such as curvature-based sampling for surface measurement[160], adaptive sampling for aerodynamic modelling[116].

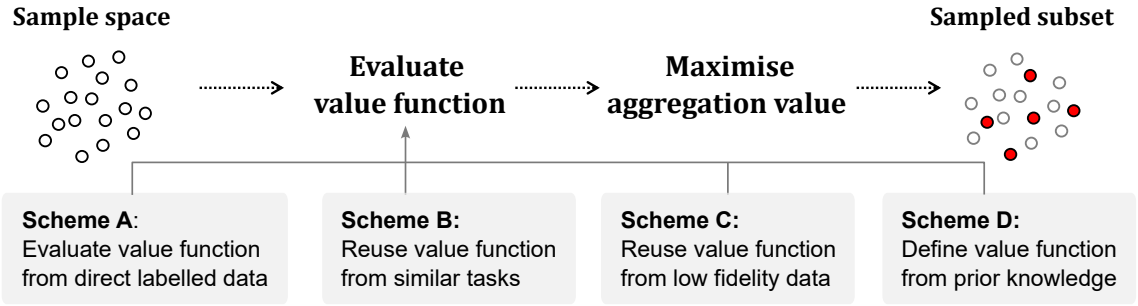


Figure 2.3: The implementation procedure of the proposed aggregation-value-based sampling method.

The following Sections will introduce the details of the proposed method. The four schemes will be validated in the case study Section.

2.3 Valuation of data using the Game theory

Data-driven modelling method is able to learn the mapping from the input features to output labels based on the given labelled data. From the probabilistic perspective, training a data-driven model can be regarded as learning the joint or conditional probability distribution on the given observable dataset [17]. However, according to the Game theory, training a model can also be defined as a coalition game between finite players [153], with each data sample defined as a participating player and the training accuracy of the model defined as the estimated value of the coalition game.

This section will first give a basic definition of the sampling problem for data-driven modelling from the perspective of the coalition game and then construct the value function for data samples based on the Shapley theory.

2.3.1 Sub-modularity in model training

Suppose there are n data points in the dataset N for a given data-driven modelling task. The sampling problem means to obtain a subset S from the entire dataset N, namely $S \subseteq N$. Therefore, N is defined as the sample space, and each data point in N refers to one sample. From the perspective of the Game theory, the dataset N can be treated as n players for a cooperative game, where the sampling problem refers to finding the best players combination to achieve a better performance in the game. This section will first introduce two important concepts from cooperative game theory, marginal gain and sub-modularity [161].

Denote $\varphi : 2^{|N|} \rightarrow \mathbb{R}$ as the indicator function that describes the prediction accuracy of the model trained on any subset of the sample space. The function φ is defined on the power set of N, i.e. the set of all subsets of N, denoted as $2^{|N|}$. The domain for the function values of φ is the real space $\mathbb{R}$.

For a subset $S \subseteq N$ and a given machine learning model $\mathcal{A}$. $\varphi(S, \mathcal{A})$ denotes the model prediction accuracy achieved by training the model $\mathcal{A}$ on the subset S. For the classification task, $\varphi(S, \mathcal{A})$ can be the prediction accuracy (%) of the model. However, for the regression task, $\varphi(S, \mathcal{A})$ can represent the negative value of the Mean Square Error (MSE) or Mean Absolute Error (MAE) of the model because a smaller MSE value means a higher accuracy of the model. In this case, the optimal subset sampling can be defined as the following optimisation problem:

$$\max_{S \subseteq N} \varphi(S, \mathcal{A}) \tag{2.1}$$

For the simplicity of expression, the indicator function $\varphi(S, \mathcal{A})$ will be abbreviated as $\varphi(S)$ because the learning model $\mathcal{A}$ is consistent in the context. For example, $\varphi(N)$ refers to the prediction accuracy of the model training on the total dataset N.

Intuitively, the 'contribution' of a single sample during the training of a data-driven model can be represented as the difference in the model performance when the sample was removed from the current dataset. Therefore, to quantify the influence of a single sample on model training, the marginal gain, also known as discrete derivative, can be defined as follow [162].

Definition 1 (Marginal gain) *For a set function $\varphi : 2^{|N|} \rightarrow \mathbb{R}$, a subset $S \subseteq N$, and an element $e \in N$. The marginal gain of φ at N respect to e is defined as:*

$$\Delta_{\varphi}(e \mid S) := \varphi(S \cup \{e\}) - \varphi(S) \tag{2.2}$$

Marginal gain can represent the increase in performance when sample e is added to the set S, or the decrease in performance when sample e is removed from the set $S \cup \{e\}$. Therefore, it does represent the 'contribution' of the sample to some extent. The sampling method that defines the marginal gain directly as a measure of the sample value is called Leave-one-out (LOO) [154].

Since the marginal gain of a sample e is defined based on the subset S, it is obvious that different subset S will bring different marginal gains for the same sample e. For a small size of training data, the additional sample may provide a significant increment in the prediction performance of a data-driven model. However, when there is sufficient training data available, the same sample could be ineffective in improving the prediction accuracy of the model. Therefore, the LOO methodology does not provide a fair and objective valuation of the sample.

As analysed above, increasing the amount of training data does not consistently lead to a linear increase in the model's performance, i.e. the marginal gain of the sample e

on the set function $\varphi(S)$ decreases as the size of the set S increases. The diminishing return property of marginal gain is defined as sub-modularity.

Definition 2 (Sub-modularity) *A set function $\varphi : 2^{|N|} \to \mathbb{R}$, is submodular if for every $A \subseteq B \subseteq N$, and $e \in N \backslash B$ it holds that:*

$$\Delta_{\varphi}(e \mid A) \geq \Delta_{\varphi}(e \mid B) \tag{2.3}$$

The sub-modularity of the model training describes the influence of existing samples on the valuation of additional samples. In the following sections, sub-modularity will be the basis of the Shapley value theory and also the basic assumption of the proposed sampling method. Note that, the data-driven model training can not always hold sub-modularity because of the uncertainty and performance difference of machine learning algorithms. This situation will be discussed in Section 2.6.1.2.

2.3.2 Valuation of data based on Shapley theory

2.3.2.1 The definition of Shapley value

In order to fairly assess the contribution of each player in a cooperative game, Shapley L.S [163] proposed an axiomatic criterion for the fair valuation in 1953 and accordingly introduced the concept of Shapley value, defined as the average marginal gain under all subsets. The Shapley theory is widely used in economics and has also been introduced into the machine-learning field in recent years. Sundararajan et al. [164] used the Shapley value to quantitatively estimate the contribution of each feature of the data to explain the feature extraction ability of a neural network. Duval et al. [165] tried to estimate the contribution of each node in a graph neural network based on the Shapley value. Ghorbani et al.[153] first proposed to estimate the sample value based on Shapley theory, and the effectiveness of the Shapley value was also validated through a series of experiments.

Based on the definition of indicator function $\varphi(S, \mathcal{A}), \varphi(S)$ for short, and the dataset N, Shapley value of datapoint $x_* \in N$ can be given as:

$$v\left(x_*\right) = \frac{1}{n} \sum_{S \subseteq N \backslash \{x_*\}} \frac{\varphi(S \cup \{x_*\}) - \varphi(S)}{\left(\begin{array}{c} n-1 \\ |S| \end{array} \right)} \tag{2.4}$$

where S represents a subset of N without sample $\left(x_*, y_*\right), \left(\varphi\left(S \cup \{x_*\}\right) - \varphi(S)\right)$ is the difference in the predictor's performance when trained on the $S \cup \{x_*\}$ and S, namely the marginal gain defined above. $\left(\begin{array}{c} n-1 \\ |S| \end{array} \right)$ is the number of subsets of size $|S|$ in $N \backslash \{x_*\}$. Therefore, the Shapley value of each training data will be determined by the training dataset, the learning algorithm as well as the indicator function.

Fig. 2.4 gives an intuitive example of the effect of Shapley values on a classification task. As shown in Fig. 2.4a, removing high-value samples leads to a rapid decrease in classification accuracy from 0.73 to below 0.6, while the removal of low-value samples even leads to an increase in model accuracy. Considering the presence of noisy samples in the real dataset, removing low-value samples can potentially improve the data quality, and therefore the prediction accuracy can be increased. For the sample addition experiment shown in Fig. 2.4b, adding high-value samples can bring rapid increase to the model accuracy. By comparison, the model accuracy increases very slowly when adding low-value samples.

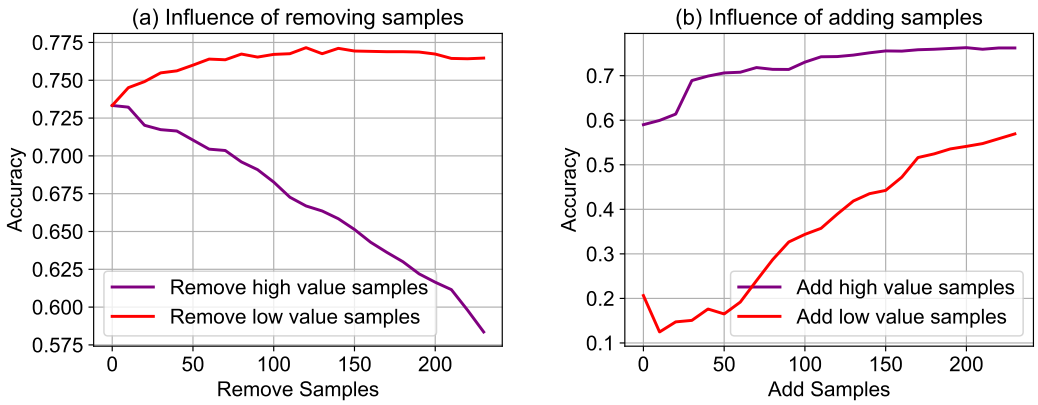


Figure 2.4: Remove or add points based on Shapley value for a classification task.

Fig. 2.5 shows another example of Shapley values on a regression task. Since a smaller MSE means a better regression performance, Fig. 2.5a shows that removing the high-value samples can lead to a significant increase in MSE. Similarly, adding high-value samples can achieve a low MSE effectively compared with adding low-value samples, as shown in Fig. 2.5b.

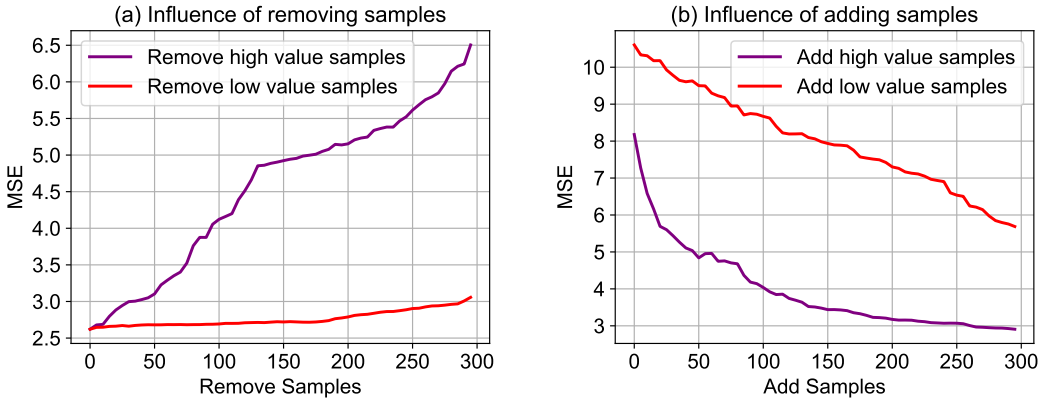


Figure 2.5: Remove or add points based on Shapley value for a regression task.

2.3.2.2 Truncated Monte Carlo for Shapley value approximation

Theoretically, computing the Shapley value in Eq. 2.4 requires all the marginal gains, which are exponentially large in the number of training data. That is an unacceptable computational burden and not practical for real applications. Consequently, Ghorbani et al. [153] proposed an approximate method to estimate the Shapley value, named truncated Monte Carlo Shapley (TMC-Shapley). This section first introduces the basic idea of TMC-Shapley and then proposes a modified version TMC-Shapley.

For the TMC-Shapley method, the Shapley value can be formulated as the following expectation evaluation problem, shown in Eq. 2.5.

$$v\left(x_*\right)=\mathbb{E}_{\pi\sim\Pi}\left[\varphi\left(S_\pi^*\cup\left\{x_*\right\}\right)-\varphi\left(S_\pi^*\right)\right] \tag{2.5}$$

where Π is the uniform distribution over all n ! possible permutations of training dataset N, and S_π^* is the subset of N consisting of all the samples before x_* in permutation π. The basic idea of TMC-Shapley is sampling a limited number of permutations π instead of calculating the marginal gains on all the subsets $2^{|N|}$.

In the beginning, a random permutation of N is first generated. Then, the marginal gain $\varphi\left(S_\pi^*\cup\left\{x_*\right\}\right)-\varphi\left(S_\pi^*\right)$ for each sample x_* can be calculated under the given learning algorithm and indicator function. The indicator function value $\varphi\left(S_\pi^t\right)$ under the permutation π can be used to calculate both the marginal gain of x_t and the marginal gain of x_{t-1}, namely the preceding samples of x_t. Therefore, the computation of the marginal gains for the samples in the permutation π requires n indicator operations, namely $\varphi\left(S_\pi^1\right),\ldots,\varphi\left(S_\pi^n\right)$.

To further reduce the computational efforts, the required n indicator operations can also be reduced, which leads to the proposed modified TMC-Shapley method. Each permutation has a starting point $S_\pi^{n_0}$ and a truncating point S_π^{trunc}. Besides, a special convergence criterion is defined for the iteration of different permutations.

Starting point in one permutation: For a machine learning task, the performance of the trained model is unstable when the size of the training set is very small. For the regression task, the marginal gains of the top few samples might have significant orders of magnitude difference, which will bring significant error to the following data valuation. Therefore, a minimum initial data size n_0 can be pre-defined when calculating marginal gain, namely starting from $\varphi\left(S_\pi^{n_0}\right)$ instead of $\varphi\left(S_\pi^0\right)$.

Truncating point in one permutation: Considering the sub-modularity of model training described in the previous section, the larger the number of samples in the set S_π^*, the smaller the marginal gain $\varphi\left(S_\pi^*\cup\left\{x_*\right\}\right)-\varphi\left(S_\pi^*\right)$. Since the marginal gain always approaches zero as the number of training data increases, the indicator operations can be truncated, and then the marginal gain of the following data points in this permutation can be directly set to zero. That means, the originally required indicator operations $\left[\varphi\left(S_\pi^1\right),\ldots,\varphi\left(S_\pi^n\right)\right]$ can be simplified to $\left[\ldots,\varphi\left(S_\pi^{n_0}\right),\varphi\left(S_\pi^{n_0+1}\right),\ldots,\varphi\left(S_\pi^{trunc}\right),\ldots\right]$, where $\varphi\left(S_\pi^{trunc}\right)$ refers to the truncated indicator value. The tolerance for the indicator truncation is defined as $\alpha\varphi(N)(0<\alpha<0.1)$. When $\varphi(N)-\varphi\left(S_\pi^t\right)<\alpha\varphi(N)$, the

following marginal gain will be set to zero. $\varphi(N)$ refers to the prediction accuracy of the model training on the total dataset N.

Convergence criterion for permutation iterations: After repeating the above steps under more permutation π, each sample will have more calculated marginal gains, and the Shapley value of each sample can be simply approximated by the average of all marginal gains. The number of permutations can be determined by the convergence of the marginal gains. In this research, the convergence criterion is defined as the maximum variation of the last k iterations of Shapley values:

$$\max\left\{\frac{1}{k}\sum_{t=m-k}^{m}\left(v_t\left(x_i\right)-v_{\min}\left(x_i\right)\right), i=1,2,\ldots,n\right\}<\beta \tag{2.6}$$

where m is the total number of iterations, $0<k<m$ is a constant, and the sum $\sum_{t=m-k}^{m}$ represents the Shapley value's variation of the last k iterations, $v_t\left(x_i\right)$ is the Shapley value of the sample x_i after completing the t_{th} iteration, and $v_{\min}\left(x_i\right)=\min\{v_t\left(x_i\right), t=m-k,\ldots,m\}$. The above criterion means that the Shapley value approximation is considered to achieve convergence when the maximum fluctuation of all Shapley values in the last k iterations is less than the given convergence tolerance β.

With the help of the defined starting points and truncating points, the proposed modified TMC-Shapley can provide more stable marginal gain results while reducing the computation efforts. To ensure the convergence of the Shapley value and reduce the randomness of the training process, the number of iterations increases until reaching the convergence criterion. The details of the proposed modified TMC-Shapley method are described in Algorithm 1.

2.3.2.3 Establish value function from discretised Shapley values

The Shapley theory can give the value of all samples in a given dataset N. However, when new samples are added to the dataset, the marginal gains have to be completely recalculated to update the Shapley values of all samples in the new dataset according to Eq. 2.4. Therefore, after obtaining the Shapley values of all samples in the initial dataset, a general Shapley value function $v(x)$ can be learnt from the paired data $[\{x_1, v\left(x_1\right)\}, \{x_2, v\left(x_2\right)\}, \ldots, \{x_n, v\left(x_n\right)\}]$ to predict the value of samples out of the dataset. Note that, the discrete Shapley values and the learnt value function have different properties.

- According to the Shapley value theory, the sum of the Shapley values of the discrete samples equal to the maximum accuracy indicator that can be achieved in the sample space N, i.e. $\varphi\left(N\right)=\sum_{i=1}^{n} v(x_i)$. However, the continuous value function does not have this property, i.e. the continuous value function only reflects the relative value relationship between samples, but the function values are directly related to the accuracy indicator.
- Because of the noise in the dataset or the approximation error of the Shapley value, the discrete Shapley values of samples have inevitable random errors. However,

Algorithm 1 Modified Truncated Monte Carlo Shapley

Input: Training data N with size n, learning algorithm $\mathcal{A}$, and indicator function $\varphi, n_0, \alpha, \beta, k$

Output: Shapley value of all training data $v_t\left(x_i\right), i=1,2,\ldots,n$

Initialise Shapley value of sample $v_0\left(x_i\right)=0, i=1,2,\ldots,n$

while Convergence criteria not met **do**

$t \leftarrow t+1$

Generate random permutation of training dataset N, denoted as π_t

Initialize $\varphi_0^t \leftarrow \varphi\left(\left\{\pi_t[1],\ldots,\pi_t\left[n_0\right]\right\},\mathcal{A}\right)$ and $k_c=0$.

for $j \in \left\{n_0, n_0+1,\ldots,n\right\}$ **do**

if $\left|\varphi(N,\mathcal{A})-\varphi_{j-1}^t\right| < \alpha\varphi(N,\mathcal{A})$ **then**

$\varphi_j^t \leftarrow \varphi_{j-1}^t$

else

$\varphi_j^t \leftarrow \varphi\left(\left\{\pi_t[1],\ldots,\pi_t[j]\right\},\mathcal{A}\right)$

end if

$v_t\left(\pi_t[j]\right) \leftarrow \frac{t-1}{t} v_{t-1}\left(\pi_t[j]\right)+\frac{1}{t}\left(\varphi_j^t-\varphi_{j-1}^t\right)$

end for

end while

$v_t\left(x_i\right) \leftarrow v_t\left(\pi_t\left[\text{Index}\left(x_i\right)\right]\right), i=1,2,\ldots,n$

$v_t'\left(x_i\right) \leftarrow v_t\left(x_i\right)-\min\left\{v_t\left(x_i\right), i=1,2,\ldots,n\right\}, i=1,2,\ldots,n$

the value function is able to eliminate the random error by fitting the discrete Shapley values.

As the case shown in Fig. 2.6a, the size of the blue circle means the Shapley value of each point when fitting the curve using the Gaussian process regressor. It can be observed that the values of turning points are larger than boundary points. When the model to be learnt is differentiable, the value function $v(x)$ should also be differentiable, which means these points near local maxima are all high-value samples. The Shapley values of all samples are shown in Fig. 2.6b, there exist clear random errors despite the global trend of the values. The fitted value function is shown in Fig. 2.6c. The random errors of discrete Shapley values are eliminated, and the smooth value function can benefit the sampling method in the following section.

2.4 Aggregation-value-based sampling method

This section will endow training data with two non-negligible meanings, namely representativeness for probability density and values for game theory, leading to the proposed AV4Sam method. A novel **Value Aggregation Function (VAF)** is designed to capture the influence of neighbouring values by adding a kernel function to the value function. The aggregation value is then derived to an elegant form, the expectation of the united

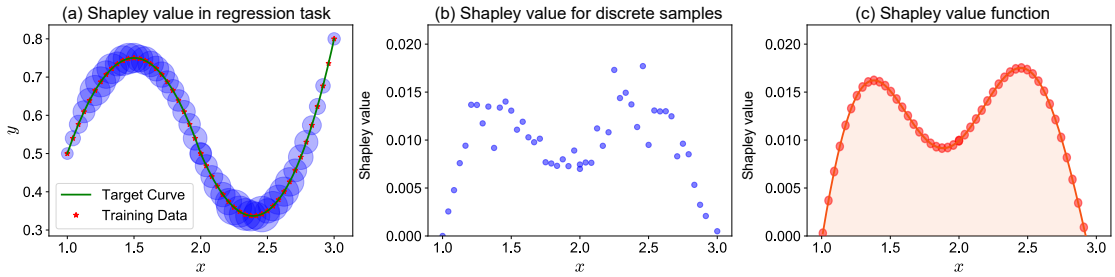


Figure 2.6: The discrete Shapley values and the value function.

VAF. The approximate optimal sample set can be easily obtained by greedy optimisation of the aggregation value.

2.4.1 Value aggregation considering neighbouring influences

For a machine learning task, it is more reasonable to evaluate the contribution of one sample considering the value of its neighbourhoods rather than only its own value, where the neighbourhood refers to the samples with a small distance in the sample space. Suppose the aggregation coefficients are positively correlated with the Euclidean distance in the feature space, the VAF of a given instance x_* is designed by adding a kernel function to the value function as:

$$v'(x, x_*) = v(x)k(x, x_*) \tag{2.7}$$

where $k(x, x_*)$ is a kernel function. The kernel function can influence how much information is aggregated from neighbouring samples. It plays a similar role as filtering in signal processing, convolution in image recognition, and attention mechanism in deep learning [166, 167].

VAF $v'(x, x_*)$ can express the neighbouring influence of instance x_*. Intuitively, the influence degree of the sample x_* to its neighbouring sample should be inversely proportional to the Euclidean distance between the samples. Therefore, the classical kernel functions, including Radial Basis Function (RBF), the Laplace Kernel, and the Polynomial Kernel, are all effective. This research will adopt RBF because of its simplicity. The comparison of the different kernel functions will be analysed in detail in the experimental Section 2.6.2.

Fig. 2.7a-c show the different influence ranges that are controlled by different bandwidth parameters σ of the kernel function. As shown in Fig. 2.7a, it is obvious that the VAF $v'(x, x_*)$ will converge to $v(x_*)$ when bandwidth approaches zero. By comparison, when the bandwidth becomes too large, the VAFs for all samples converges to the value function $v(x)$ itself (as shown in Fig. 2.7c), i.e., the values at each point are no longer distinguishable.

To descript the quantitative contribution of the observation instance x_*, the expec-

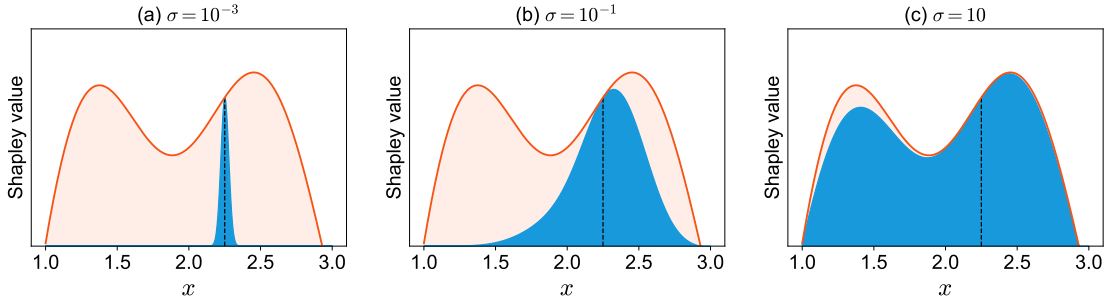


Figure 2.7: Shapley value for a regression toy case.

tation of VAF is defined as the **aggregation value of sample** x_* :

$$v_{agg}\left(x_*\right) = \int p(x) v'\left(x, x_*\right) dx \tag{2.8}$$

where $p(x)$ is the probability density function of the variable x in the sample space. When the potential dataset is given by a limited set N, the discrete version of aggregation value is given as:

$$\hat{v}_{agg}\left(x_*\right) = \frac{1}{n} \sum_{x \in N} v'\left(x, x_*\right) \tag{2.9}$$

Eq. 2.8 and 2.9 can output the aggregation value of one sample. But for a set S with more than one sample, the VAF is not the sum of the individual VAFs. Next section will introduce the VAF of a sample set.

2.4.2 Represent the value of a subset

The neighbouring samples in the sample space might carry similar or redundant feature information, which cannot bring a proportional contribution to the model training. Therefore, the value of a sample set can not be defined as the sum of the individual value. The defined aggregation value can represent the values of the neighbouring samples, thus, can be used to represent the redundant information.

As shown in Fig. 2.8, since the VAF of each instance is defined on the entire feature space, the VAFs of different samples may overlap, which can represent the redundant information explicitly. If two samples are very close to each other, the majority of their VAFs will overlap. The two samples in Fig. 2.8a are far from each other, so there is only a little overlap of the VAFs. In contrast, the two samples close to each other in Fig. 2.8b have greater overlaps of their VAFs, namely more redundant information.

Therefore, the VAF of a set can be represented as the 'union' of the individual VAFs, just like the union in Boolean geometry operations. Consequently, the VAF of the set $S = \{x_1, x_2, \ldots, x_m\}$ is then defined as:

$$v'(x, S) = v(x) \max\left\{k\left(x, x_1\right), \ldots, k\left(x, x_m\right)\right\} \tag{2.10}$$

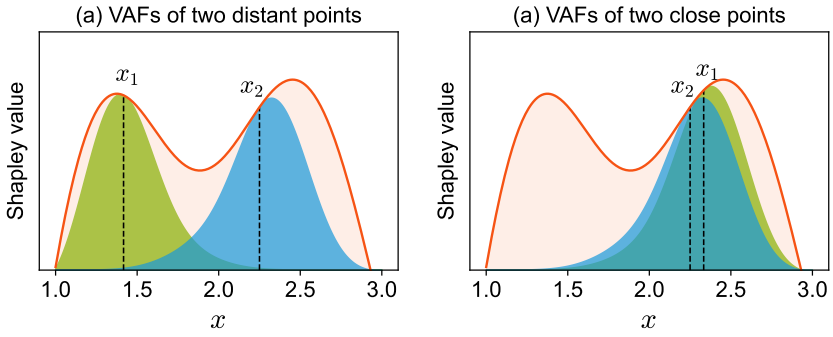


Figure 2.8: Shapley value for a regression toy case.

The function can intuitively represent the neighbouring value distribution and show how much value should be aggregated for a sample s when set S is available. The quantitative expression, namely the **aggregation value of set** S, is therefore the same:

$$v_{agg}(S) = \int p(x)v'(x,S)dx \tag{2.11}$$

Similarly, the estimation under finite number of samples is:

$$\hat{v}_{agg}(S) = \frac{1}{n}\sum_{x\in N} v'(x,S) \tag{2.12}$$

Under this definition, closed high-value instances cannot provide high aggregation value. The optimal sample set selection problem can be defined as a submodular optimisation problem [162]:

$$\max_{S\subseteq N} \hat{v}_{agg}(S) \tag{2.13}$$

It is admitted that the above problem is not a strict submodular optimisation problem because Shapley values of some instances could sometimes be negative. This situation is discussed in Section 2.6.1.2.

2.4.3 Greedy optimisation of the aggregation value

Submodular maximisation is an NP-hard problem, which means that the optimal solution is ordinarily inaccessible. Fortunately, the greedy algorithm can provide an approximation to the optimal solution of the submodular maximisation problem with the guarantee up to a factor of $1-1/e$[168]. Starting from the empty set S_0, the greedy algorithm queries the new sample to maximise the marginal gain $\Delta\left(e \mid S_{i-1}\right) = \hat{v}_{agg}\left(S_{i-1}\cup\{e\}\right) - \hat{v}_{agg}\left(S_{i-1}\right)$. Then iterative set S_i can be obtained as:

$$S_i = S_{i-1} \cup \left\{ \arg\max_e \Delta\left(e \mid S_{i-1}\right) \right\} \tag{2.14}$$

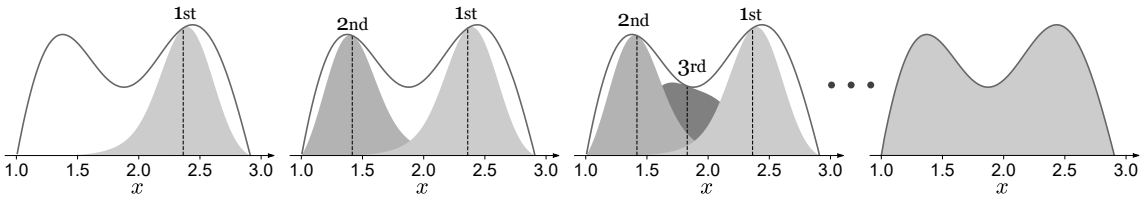


Figure 2.9: The greedy optimisation of the aggregation value.

The illustration of greedy-based sampling is shown in Fig. 2.9. The redundant information can be represented as the overlapped grey regions of VAFs.

Since the defined aggregation value is always nonnegative and monotone with increased samples, it is easy to obtain a basic approximation guarantee of greedy submodular maximisation [162]. Suppose ℓ points are sampled from the set N, let $S^* \in \arg\max_S \{\hat{v}_{agg}(S) : |S| \leq k\}$ be an optimal set of size k. The approximation bound can be given as:

$$\hat{v}_{agg}\left(S_\ell\right) \geq \left(1 - e^{-\ell/k}\right) \hat{v}_{agg}\left(S^*\right) \tag{2.15}$$

The sampling problem has been transferred to a simple greedy optimisation problem. The optimal subset S^* can conveniently be sampled from the potential dataset N. The maximisation of aggregation value can balance the distribution and value of points, which breaks the limitations of the traditional value-based sampling method. The following section will validate the proposed sampling method with detailed experiments.

2.5 Case study

This section will validate and analyse the proposed sampling method in different manufacturing predictive modelling problems. Although the aggregation-value-based method is derived from the Shapley value function, the basic idea of the aggregation value can be generalised to other forms of value function as long as it can provide a positive correlation with the real contribution of samples. Therefore, the case study section generalises the proposed method to more practical scenarios with different value function schemes.

2.5.1 Evaluate value function from direct labelled data

Figure 2.10a-d report the detailed results of different sampling methods on four datasets. Different number of samples are selected from the potential dataset. A machine learning model is then trained on the selected samples and evaluated on the test set. 'HighAV' (High Aggregation Value) is the high valuable dataset sampled by the proposed AV4Sam method. 'HighSV' (High Shapley Value) means sampling the high-Shapley-value samples greedily to construct the sample set [153]. 'Cluster' is the clustering-based core set selection strategy [151]. Lastly, 'Random' means generating the sample set randomly. The detailed data processing and model training are reported in the Supplementary data (S2, S3). The brief description of these tasks is summarised below.

1. **Image classification:** Cifar10 [169] is a widely used classification dataset in the image process field. A small dataset is constructed from Cifar10 to evaluate the generalisability of the proposed method, and the result is shown in Fig. 2.10a.

2. **Rolling bearing fault classification:** The bearing fault dataset from the Case Western Reserve University (CWRU) [170] is a famous benchmark dataset in the fault diagnosis field. The vibration signals of normal and faulty conditions are collected under different motor loads (HP=0,1,2,3). Two 10-way classification problems (HP=0, HP=1) are formulated, and the classification accuracies with varying numbers of samples from all methods are shown in Fig. 2.10b(HP=1).

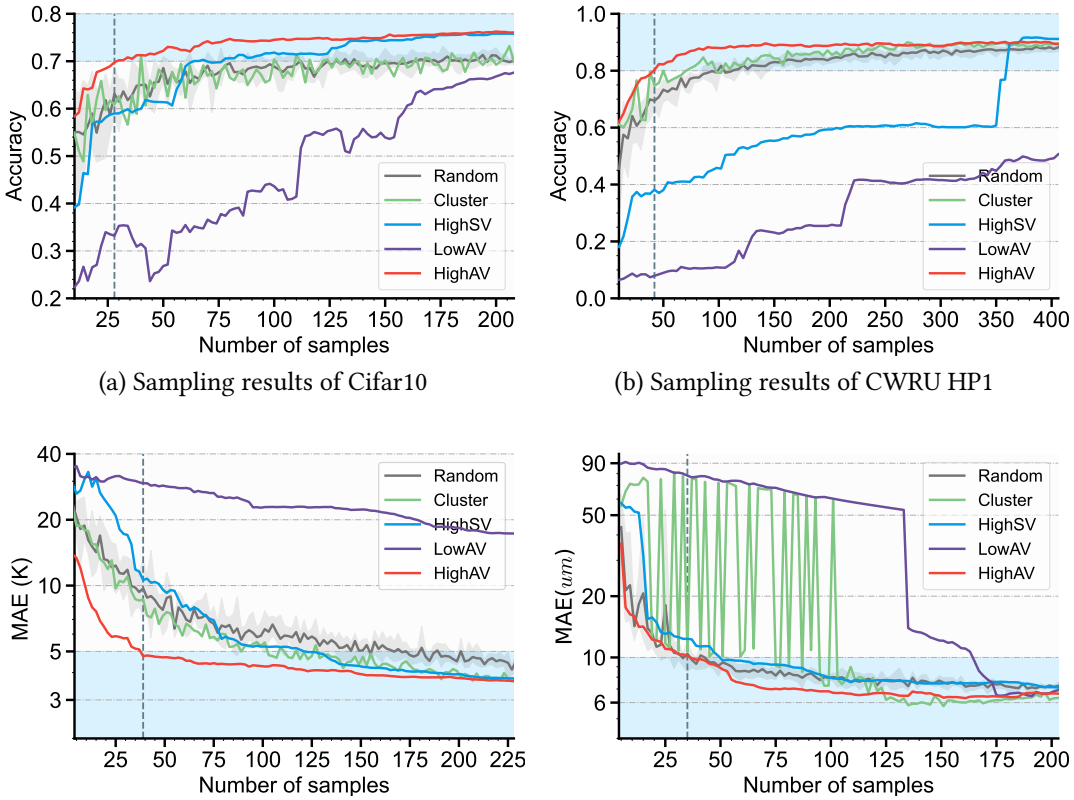
3. **Composite curing:** Predicting the thermal lag of temperature from curing parameters is an important task for the quality control of composite parts. 600 combinations of curing parameters are generated from a reasonable range, and the corresponding thermal lags are simulated using Finite Element (FE) software [85, 10]. The MAE results are shown in Fig. 2.10c.

4. **Tool wear prediction:** The tool wear dataset from the Prognostics and Health Management Society [171] consists of collected monitoring signals during milling and the corresponding tool wear values for three blades of cutting tools. The input of the regression problem is the extracted 64-dimensional features. Two regression tasks are formulated, blade No. 2 of cutting tool No. 4 and blade No. 3 of cutting tool No. 6, B2C4 and B3C6 for short. The MAE results are shown in Fig. 2.10d(B3C6).

Figure 2.10a-d illustrate that HighAV can consistently achieve superior performance, especially when the number is limited. Under most circumstances, HighAV outperforms the uncertainty boundary of Random (grey region) while Cluster is only better than Random occasionally but far more unstable. The Cluster results fluctuate sharply because similar sample sizes (e.g. 49 and 51 in Fig. 2.10d) can lead to totally different clusters, and some high-value samples might be missed. In Fig. 2.10b, the results of HighSV show 'step effect', namely suddenly increasing at some point (around 350 in Fig. 2.10b).

Theoretically, minimising the aggregation value can also provide the worst sample set. As seen in Fig. 2.10, 'LowAV' (Low Aggregation Value), sampled by greedily minimising the aggregation value, can always provide far worse results than the lowest bound of Random. Although the low valuable sample set seems meaningless for real application, it does reveal the importance of the distribution of training data, as well as the magic of AV4Sam.

Table 2.1 summaries the regression and classification results with training data from different sampling methods under different samples size (30, 50, 80, 100). It is clear that the proposed AV4Sam method can provide better sample sets compared to other sampling methods.



(a) Sampling results of Cifar10

(b) Sampling results of CWRU HP1

(c) Sampling results of composite curing task

(d) Sampling results of Tool wear B3C6

Figure 2.10: Comparison of different sampling methods.

2.5.2 Reuse value function from similar tasks

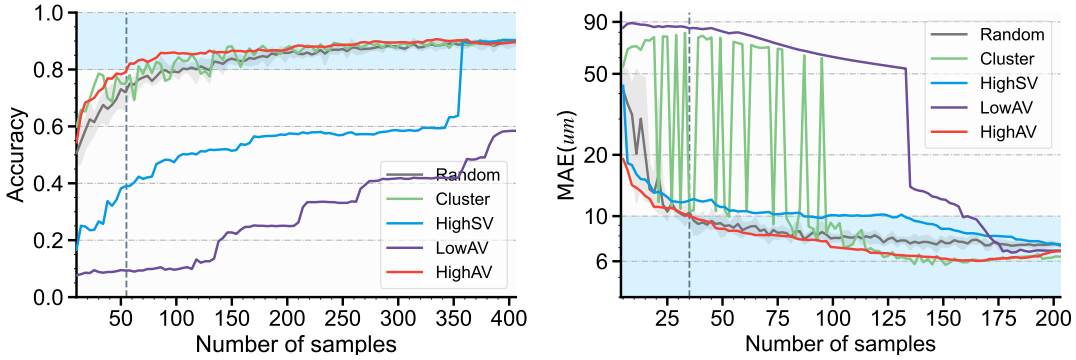
To avoid data labelling for the value function, this section investigates the possibility of reusing the value function learnt from a similar task on the target task without training a new one. As introduced in Section 2.5.1, the fault diagnosis dataset CWRU consists of two classification tasks, HP0 and HP1. Suppose the dataset CWRU HP0 is available, the value function of CWRU HP0 can be calculated first and applied in the sampling of the problem CWRU HP1.

Figure 2.11a presents the cross tasks application of reusing value function from CWRU HP0 on HP1. It can be observed that the accuracy of HighSV is even lower than Random, but HighAV can consistently achieve leading performance. This phenomenon reveals that the effectiveness of HighSV relies heavily on the accuracy of the value function. However, HighAV is more robust, meaning that a less accurate value function can still provide helpful value information. The same conclusions can also be summarised from Fig. 2.11b, which are results of cross tasks application of reusing the value function of B2C4 on B3C6.

Table 2.1: Summary of model performance with training data from different sampling methods.

	CWRU HP1 (ACC/%)		Cifar10 (ACC/%)		Tool (MAE/μm)		Composite (MAE/K)	
Samples	50	100	30	80	50	100	30	80
Random	73.18%	80.52%	61.47%	69.60%	9.32	7.98	10.90	6.54
Cluster	74.80%	81.56%	62.62%	66.56%	72.20	8.16	9.90	5.95
HighSV	48.57%	55.84%	59.00%	70.62%	9.90	7.98	16.54	5.87
HighAV	**83.12%**	**87.79%**	**70.20%**	**74.71%**	**8.88**	**6.70**	**5.70**	**4.41**

The best results are marked with bold.



(a) Reusing $v(x)$ of task CWRU HP0 on HP1.

(b) Reusing $v(x)$ of task B2C4 on B3C6.

Figure 2.11: Comparison of different sampling methods.

2.5.3 Reuse value function from low fidelity data

This section investigates Scheme C for the composites curing case, in which the value function is first calculated from the simplified low-fidelity Finite Difference (FD) model, and then reused for parameters designing in high-fidelity FEM simulations.

2.5.3.1 Problem statement

The illustration of the curing of a 1-D composite-tool system is shown in Fig. 2.12a. The actual temperature of the composite part always lags behind the designed cure cycle (Fig. 2.12 right). Thus, the thermal lag is defined as the maximum difference between the cure cycle and the actual temperature of any point in the part thickness during the heat-up step [85, 10].

The objective here is to establish the data-driven prediction model of the thermal lag, where the input features include the heating rate, the cooling rate, the hold temperature, the hold time, and the heat transfer coefficients of both sides. Since the labelled data comes from the time-consuming high-fidelity FEM simulation, a better sampling method should reduce the number of simulations while maintain the required accuracy of the data-driven model.

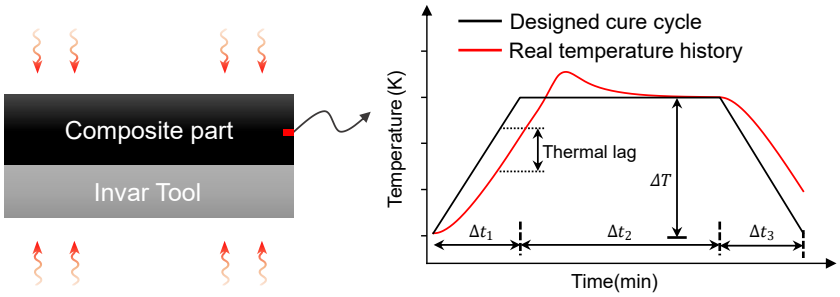


Figure 2.12: Illustration of 1-D composite-tool curing system and one hold cure cycle.

The detailed procedure of AV4Sam on this case is shown in Fig. 2.13. 600 curing parameters are generated first as the potential dataset. The simplified FD model is established first as the low-fidelity model. The thermal lags of the 600 curing parameters are obtained with the low-fidelity model, the simulation data is therefore defined as the low-fidelity data. Although the low-fidelity data is not accurate enough for composite manufacturing, it can be used to calculate the value-function $v(x)$. An optimal parameters sample set S is then determined based on the proposed sampling method for the subsequent complete high-fidelity FEM simulations.

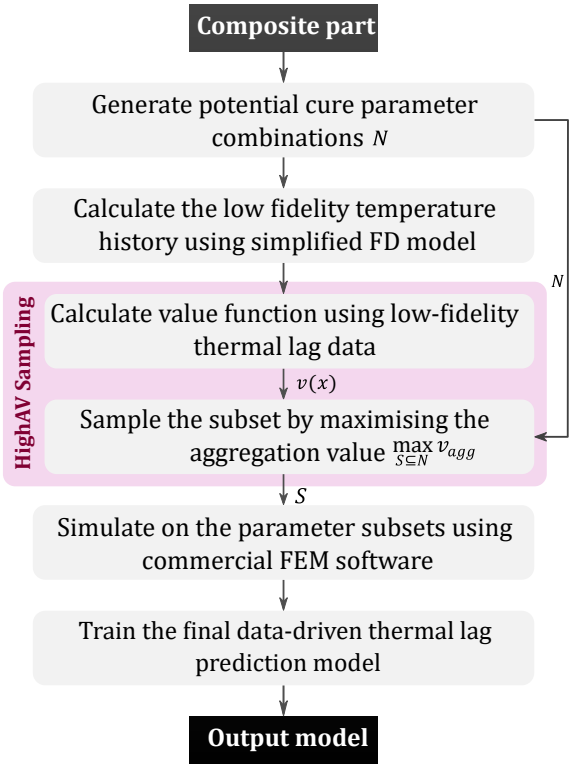


Figure 2.13: The workflow of sampling curing parameters for composites simulation.

2.5.3.2 Experimental results

To compare the influence of the data sampling on the performance of the data-driven model, a parameters sample set with n=40 instances was selected from the potential dataset by different sampling methods. A Gaussian Process Regression (GPR) model was then trained on the simulation results of the selected samples and evaluated on the test set. The MAE of 10 repeated trials of four methods are shown in Fig. 2.14. It can be observed that HighAV achieved a superior and stable performance with MAE around 5K. Conversely, Cluster is slightly better than Random, and HighSV is very unstable, even worse than Random. These results show that the distribution of the designed curing parameter combinations significantly influences the performance of data-driven models, and the proposed HighAV provided a better sample set stably.

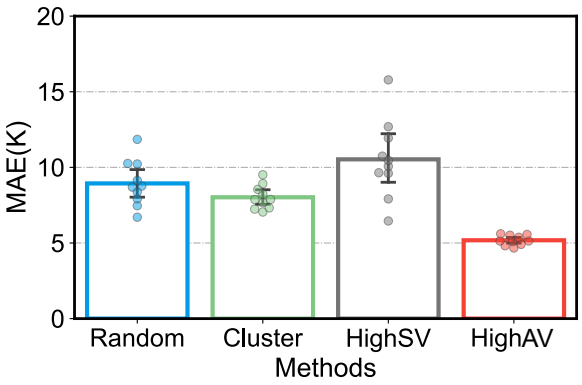


Figure 2.14: MAEs of 10 repeated trails of different samples selection methods with 40 samples for composite task.

Figure 2.15 presents how many samples were required to achieve an MAE of 5K for different sample selection methods. In each independent experiment, a sample set was constructed by increasing instances one by one from an empty set until the MAE became less than 5K stably. The size of the final sample set was recorded as the required size of this trial. As shown in the scatters and box plots of 10 repeated tests in Fig. 2.15, HighAV achieved the MAE of 5K with around 50 samples, much less than the required number of Random and Cluster.

Table 3.2 presents the detailed required samples for different sampling methods to stably achieve an MAE of $5K$ and $6K$. These results demonstrated that the proposed sampling method could reduce the data-collecting effort of FEM simulations in the composite curing problem while maintaining the required accuracy.

2.5.4 Define value function from prior knowledge

This section investigates Scheme D for the case of surface measurement and reconstruction, in which the value function is defined from prior knowledge, the absolute Gaussian curvature of the surface.

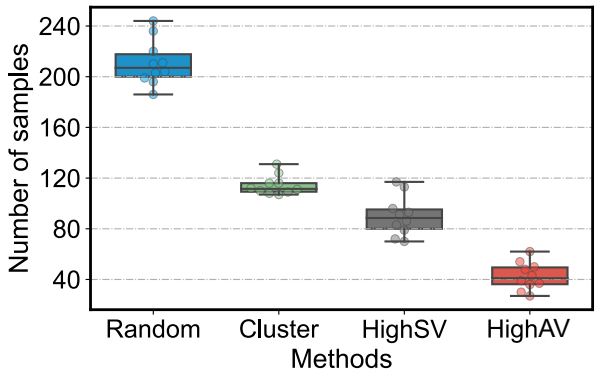


Figure 2.15: Required samples of different samples selection methods to achieve an MAE of 5K for composite task.

Table 2.2: Comparison of the required number of samples.

	Composite [1]		Surface [1]	
MAE	$6K$	$5K$	0.1mm	0.01mm
Random	172±21	234±30	122±10	292±8
Cluster	83±5	124±8	85±3	199±4
HighSV	68±8	104±21	331+8	-
HighAV	25±5	53±11	62±2	147±1

[1] The required number of samples M for different sampling methods to achieve an predefined required MAE. Considering the uncertainties of different methods, the number M is defined as: during the sampling from 20 to 520 points, for any sample set with more than M samples, the MAE is always less than the required one. A±B represents the mean (A) and standard deviation (B) of the required number M in 10 repeated trials.

2.5.4.1 Problem statement

Dimensional inspection and reconstruction of engineering products comprising free-form surfaces require accurate measurement of a large number of discrete points using a coordinate measuring machine with a touch-trigger probe [172]. An efficient sampling method should enable the reconstruction of the surface under the required accuracy with a limited amount of measurement points.

Curvature and other geometric features are widely used prior knowledge for traditional measurement sampling methods. By defining the curvature of the surface as the value function $v(x)$, the proposed aggregation-value-based sampling method can also be extended to sample the measurement points for further surface reconstruction.

Figure 2.16 shows the full workflow of how to design the measurement points through the proposed sampling method. The potential measurement sample set N, can be first generated from the nominal surface uniformly or randomly. Each sample in the set N is represented by the x, y, z coordinates of a point. The curvature function obtained from

the CAD model is then defined as the value function $v(x)$ for performing the HighAV sampling, in which the indicator function should be the reconstruction error of the nominal surface. For a given number of planned samples, an optimal subset $S \in N$ will be obtained by maximising the aggregation value. Theoretically, the points in the subset S could preserve valuable information when reconstructing the nominal surface. Therefore, it is reasonable that these points could also guide the measurements and the reconstruction of the real surface.

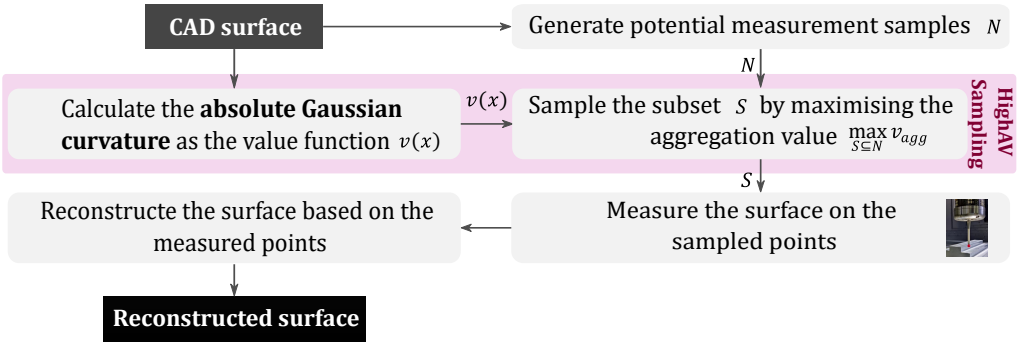


Figure 2.16: The full workflow of sampling measurement points for the surface measurement and reconstruction.

In this case, a Matlab Peak surface shown in Fig. 2.17 was adopted for the simulated measurements and reconstruction. The function of the Peak surface (Eq. 2.16) is defined as the nominal CAD surface without noise. As the colour map shown in Fig. 2.17, the absolute Gaussian curvature evaluated by the python library Pyntcloud is defined as the value function, which could provide information about the importance of different regions when reconstructing the surface. The set of the potential sample space N consists of 900 points uniformly sampled from the Peaks function. The basis learner for surface reconstruction is Gaussian Process Regression (GPR) based on the GPyTorch package and the indicator function is the MAE of the reconstruction result. After the HighAV sampling, the points in the sampled subset S will measured to get the real coordinate of the physical surface. Normally, there will be geometrical and dimensional defects from the manufacturing process. In this case, a Gaussian noise $\sigma = 0.02mm$ is added on the points in the sampled subset S to simulate the measurement points S_{meas}. Another set N_{real} with 1600 points integrating a Gaussian noise $\sigma = 0.02mm$ is assumed as the coordinate of the real surface to evaluate the accuracy of the reconstructed surface.

$$
\begin{aligned}
z = {} & 6\left(1-\frac{x}{16}\right)^2 \cdot \exp\left(-\left(\frac{x}{16}\right)^2 - \left(\frac{y}{16}+1\right)^2\right) - \\
& 20\left(\frac{x}{5\times 16} - \left(\frac{x}{16}\right)^3 - \left(\frac{y}{16}\right)^5\right) \cdot \exp\left(-\left(\frac{x}{16}\right)^2 - \left(\frac{y}{16}\right)^2\right) \\
& - \frac{2}{3}\exp\left(-\left(\frac{x}{16}+1\right)^2 - \left(\frac{y}{16}\right)^2\right), x, y \in [-40, 40]
\end{aligned}
\tag{2.16}
$$

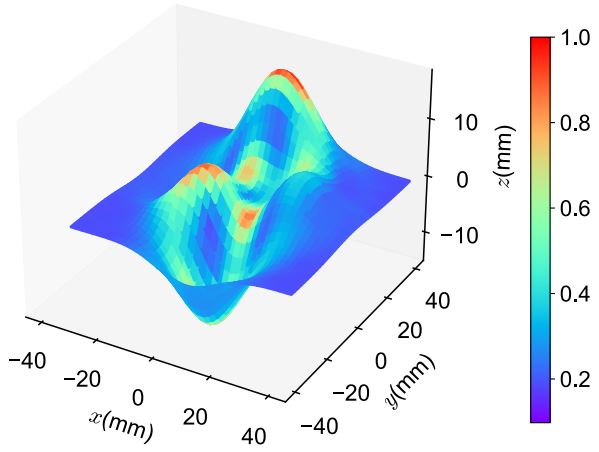


Figure 2.17: The absolute Gaussian curvature function of Peaks surface.

2.5.4.2 Experimental results

Figure 2.18 presents the MAEs of four sampling methods with different number of samples from 20 to 520. MAEs refer to the error between the surface reconstructed by S_{meas} and the simulated real points N_{real}. HighAV has a small MAE for almost any size of samples.

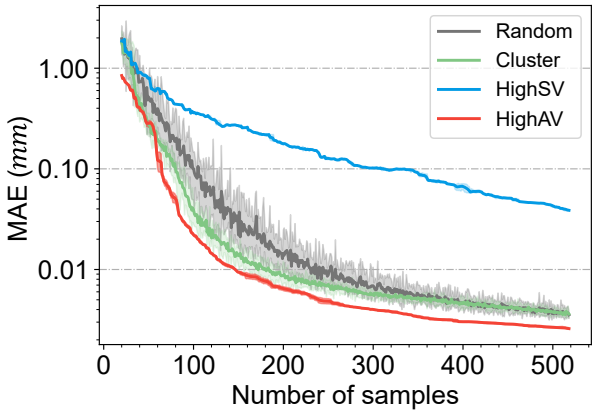


Figure 2.18: The relationship between number of samples and the MAE of the reconstructed surfaces for different sampling methods.

Table 4.3 presents the required samples for different sampling methods to stably achieve an MAE of 0.1mm and 0.01mm. It is clear that HighAV reduced the required measurement points under the predefined MAE.

Figure 2.19a is the error distribution map of the reconstructed surface with 140 measurement points sampled by HighAV. MAE is 0.010mm and the maximum absolute error (marked as MAX) is 0.078mm. Figure 2.19b shows the errors of the surface reconstructed with 140 points sampled by Cluster. MAE and MAX are 0.015mm and 0.139mm respec-

tively. It is clear that HighAV can reduce the error of areas with high curvatures, which plays a similar role as traditional curvature-based sampling.

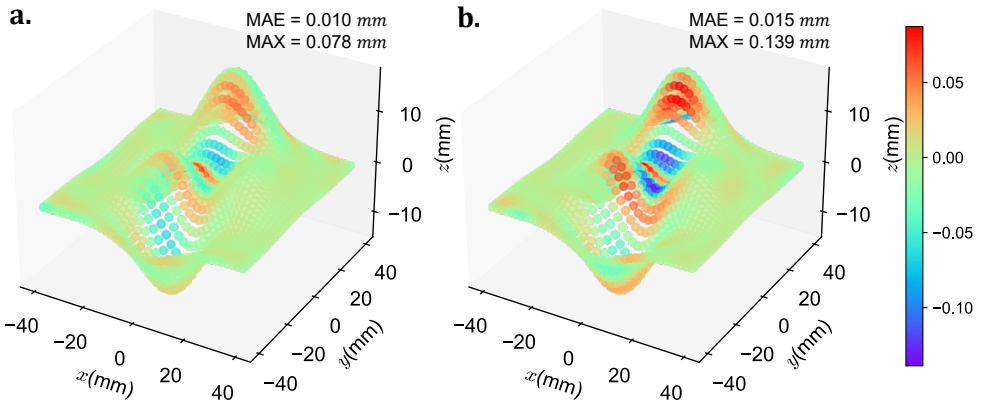


Figure 2.19: The error map of the surface reconstructed.

2.6 Formal analysis of AV4Sam

2.6.1 Characteristic of the method

The above mentioned results show that AV4Sam can provide superior and stable sample sets compared with Shapley-value-based or representativeness-based methods. This section comprehensively analyses the characteristics of AV4Sam and explains why it works.

2.6.1.1 Value distribution analysis

Fig. 2.20a-c show the t-SNE visualised features of the samples in the composite task. These samples are generated by HighSV, HighAV and Cluster, respectively. Green points are part of the dataset, and blue points are samples in the generated sample sets.

The red background in Fig. 2.20a-b represents the Shapley value field, and the darker the shade of red, the larger the Shapley value. Almost all the samples in Fig. 2.20a are concentrated in the upper-left high-value area, showing that this high-value samples clustering phenomenon would result in a severe information redundancy phenomenon. Shapley-Value-based sampling tends to be deficient because the sample set does not represent the dataset.

The blue background in Fig. 2.20c is the kernel density estimation result of the samples' distribution in the dataset. The sample set of Cluster is representative of the probabilistic density of the dataset. Still, samples in the high-value area are random and insufficient, which explains the unstable fluctuation of Cluster in Fig. 2.10d-e. As shown in Fig. 2.20b, the sample set generated by HighAV take both Shapley value and probabilistic density into consideration and provide balanced and reasonable results. The same

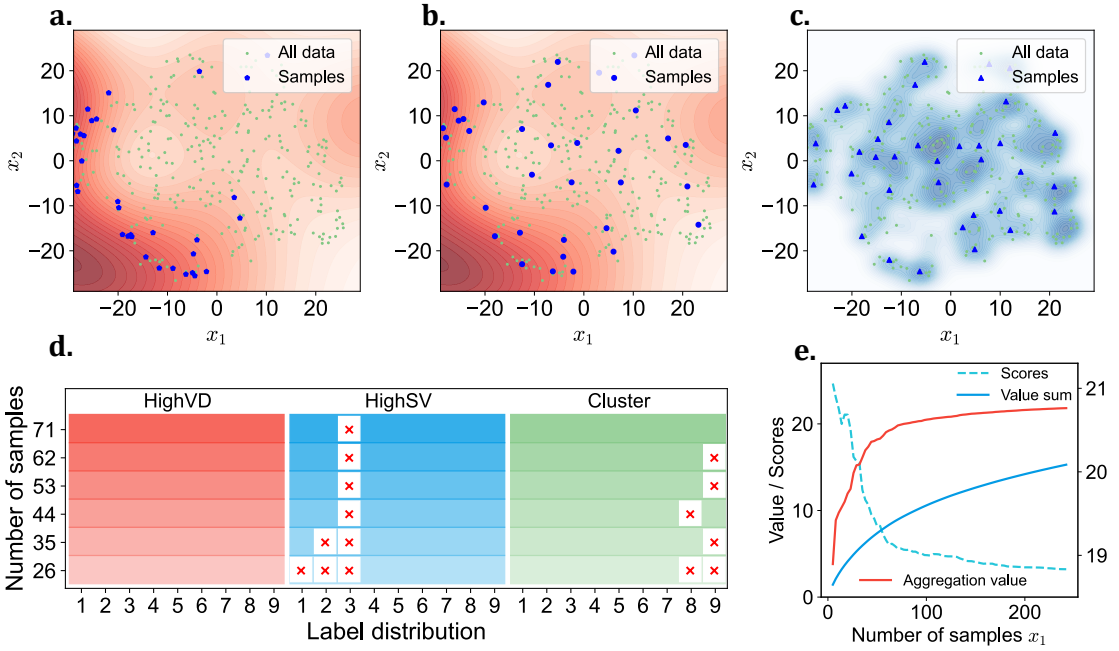


Figure 2.20: Samples distribution analysis of the composite task.

observations can also be summarised from Fig. 2.21a-c, which are features of the CWRU HP1 task.

Due to the information redundancy phenomenon, the functions between the number of samples and the corresponding performance of the data-driven model are usually approximately logarithmic, as shown in the black curves in Fig. 2.20e and Fig. 2.21e, which are MAE of the composite task and classification accuracy of CWRU HP1 respectively. Therefore, the direct sum of Shapley value of all samples (blue curves in Fig. 2.20e and Fig. 2.21e) cannot reflect the variation trend of the performance reasonably. On the contrary, the defined aggregation value (red curves in Fig. 2.20e and Fig. 2.21e) can provide a more correlative evaluation of the actual performance.

Fig. 2.20d and Fig. 2.21d illustrate the label distributions under different number of samples of the two tasks. The abscissa 1-9 in Fig. 2.20d shows that the labels of thermal lag are divided into nine intervals, and the 1-10 in Fig. 2.21d are the ten categories of bearing faults. The red cross represents the vacancy of the label in that interval. It's interesting that HighSV always suffers from the vacancy of specific intervals in the label space, this may come from the low Shapley value of these samples. However, the distribution of the labels will directly influence the generalisability of the data-driven models. The model could not provide reliable prediction results when the training data did not cover the label space. Cluster results show a relatively balanced label distribution, despite a few missing labels, and the situation is improved when with more samples. On the contrary, HighAV can cover all the label intervals with only a few samples in both the regression and the classification tasks.

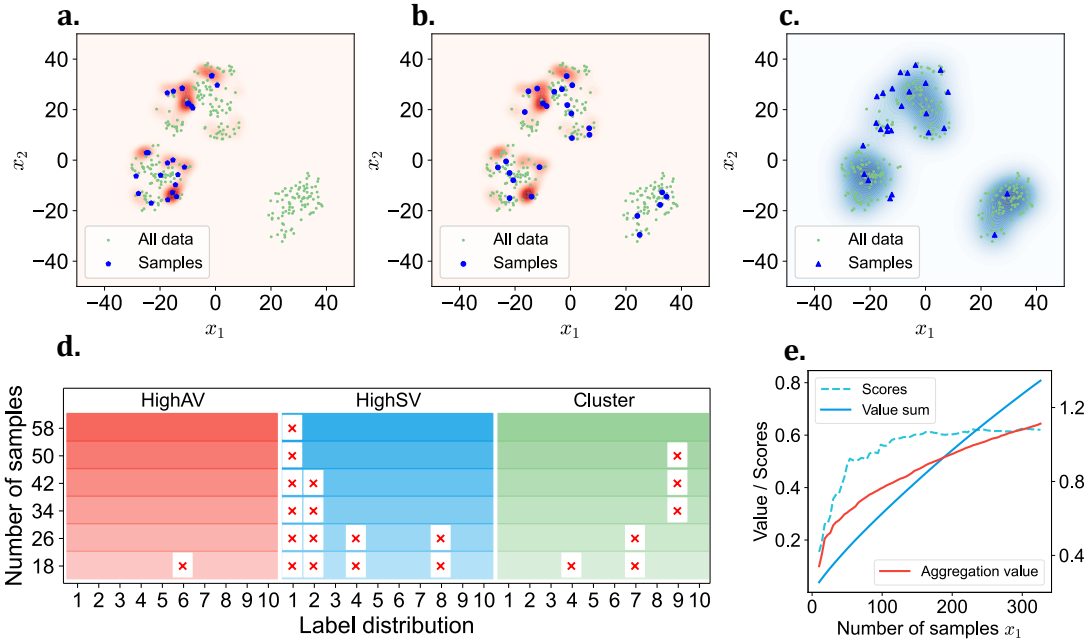


Figure 2.21: Samples distribution analysis of the CWRU task.

2.6.1.2 Sub-modularity analysis

Sub-modularity of the model training is the basic assumption for the proposed sampling methods. This section will analyse the distribution of Shaply value and its influence on the sub-modularity property, which can provide support for the definitions in Section 2.3.1.

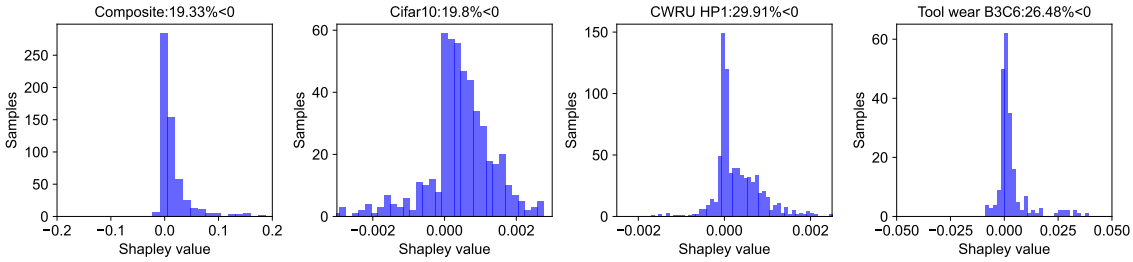


Figure 2.22: Shapley value distribution for the four cases.

Fig. 2.22 shows the distribution of the calculated Shapley value for four tasks. The title of each figure shows the proportion of points with a negative Shapley value. The Shapley value means the contribution of each sample in the model training, so the outlier or noise may have a negative influence on the modelling, thus resulting in a negative Shapley value. In addition, the uncertainties and approximation error of the modified TMC Shapley method may also lead to negative Shapley values. Anyway, these negative Shapley values may have two influences on the basic assumption of the proposed aggregation method.

- As shown in definition 2 in the Section 2.3, the sub-modularity requires $\Delta_\varphi(e \mid$

$A) \geq \Delta_{\varphi}(e \mid B)$ for $A \subseteq B \subseteq N$ [9]. Suppose set A consists of only a few samples, in contrast, set B has sufficient samples to train a high-accuracy model, namely $|A| \ll |B|$. An outlier sample e may reduce the performance of the subset A significantly but only have a few negative influences on the subset B, which means $\Delta_{\varphi}(e \mid A) < \Delta_{\varphi}(e \mid B) < 0$. Therefore, the aggregation value $\hat{v}_{agg}(S)$ is not a strict sub-modular function anymore.

- The defined aggregation value $\hat{v}_{agg}(S)$ is maximised using the greedy algorithm, but greedy submodular maximisation requires the submodular function to be non-negative and monotone [162], namely $\hat{v}_{agg}(B) \geq \hat{v}_{agg}(A) \geq 0$ for any $A \subseteq B \subseteq N$. Therefore, the negative Shapley values will influence the effectiveness of greedy optimisation. Although greedy optimisation can still work for most situations with negative Shapley values, there is a simple way to avoid instability. After the calculation of the modified TMC-Shapley method, each point v_i should subtract the smallest Shapley value in the set to keep it nonnegative: $v_i = v_i - \min(v_i, \cdots, v_n)$, for $i = 1, \cdots, n$. After this operation, the function $\hat{v}_{agg}(S)$ is a strict nonnegative and monotone submodular function, which brings a guarantee for the greedy optimisation of Eq. 2.15.

2.6.2 Sensitivity of the method

This section will analyse the sensitivity of the proposed sampling method in terms of the kernel parameter and the randomness of the Shapley value approximation.

2.6.2.1 Kernel function

The kernel function plays an important role in aggregating the neighbouring values and constructing VAFs. This section investigates the performance of three different kernel functions, Radial Basis Function (RBF) kernel, Laplace kernel and Inverse Multiquadric (IM) kernel.

RBF kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma}\right), \sigma = 1e2 \tag{2.17}$$

where the denominator can also be expressed as the equivalent definition $\sigma = 2l^2$.

Laplace kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right), \quad \sigma = 1e2 \tag{2.18}$$

IM kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{\sqrt{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 + c^2}}, \quad c = 1 \tag{2.19}$$

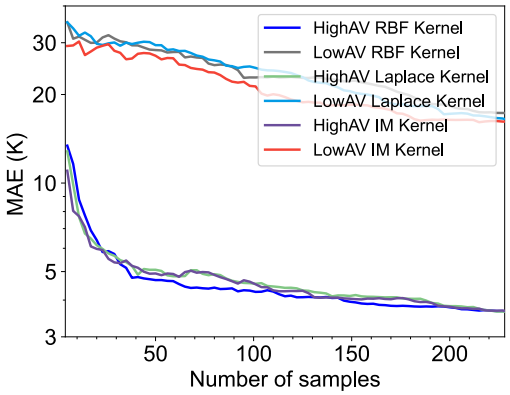


Figure 2.23: The sensitivity of HighAV on different kernel functions.

Figure 2.23 compares the HighAV results for the composite case with different kernel functions. The upper three curves are the results of LowAV, the lowest aggregation value subset. The bottom three curves are the results of HighAV, the highest aggregation value subset. It can be found that different kernel functions have similar convergence curves despite slight fluctuations. Considering the uncertainty of sampling, all three kernels can provide similar sampling performance for the proposed sampling method. The RBF kernel is selected for all the case studies because it is simple and general enough.

2.6.2.2 Kernel width

The bandwidth parameter of the kernel function σ, which determines the influence range, is the one and only parameter in AV4Sam. When σ is too small, the neighbouring values will not be aggregated, and AV4Sam will degenerate into Shapley-value-based sampling. On the other hand, AV4Sam will be less effective when σ is too large because the aggregation value of all samples could be too similar to be distinguished.

Fig. 2.24a and Fig. 2.24c show performance on the composite task and the Cifar10 task concerning σ from 0.05 to 1000. The darker shade means worse performance, and the red line is the selected parameters in the previous experiments. The large light-yellow region implies that the AV4Sam can achieve relatively robust performance in a wide range of σ. For the composite task, the performance is almost consistent for $\sigma \in [1, 1000]$.

Fig. 2.24b and Fig. 2.24d show the degeneration process of the method as σ becomes smaller. For the composite task in Fig. 2.24b, HighAV turns to HighSV gradually from $\sigma = 1$ to $\sigma = 0.1$. The degeneration process of the Cifar10 task (Fig. 2.24d) is more abrupt, which can also be observed in the bottom left corner of Fig. 2.24c.

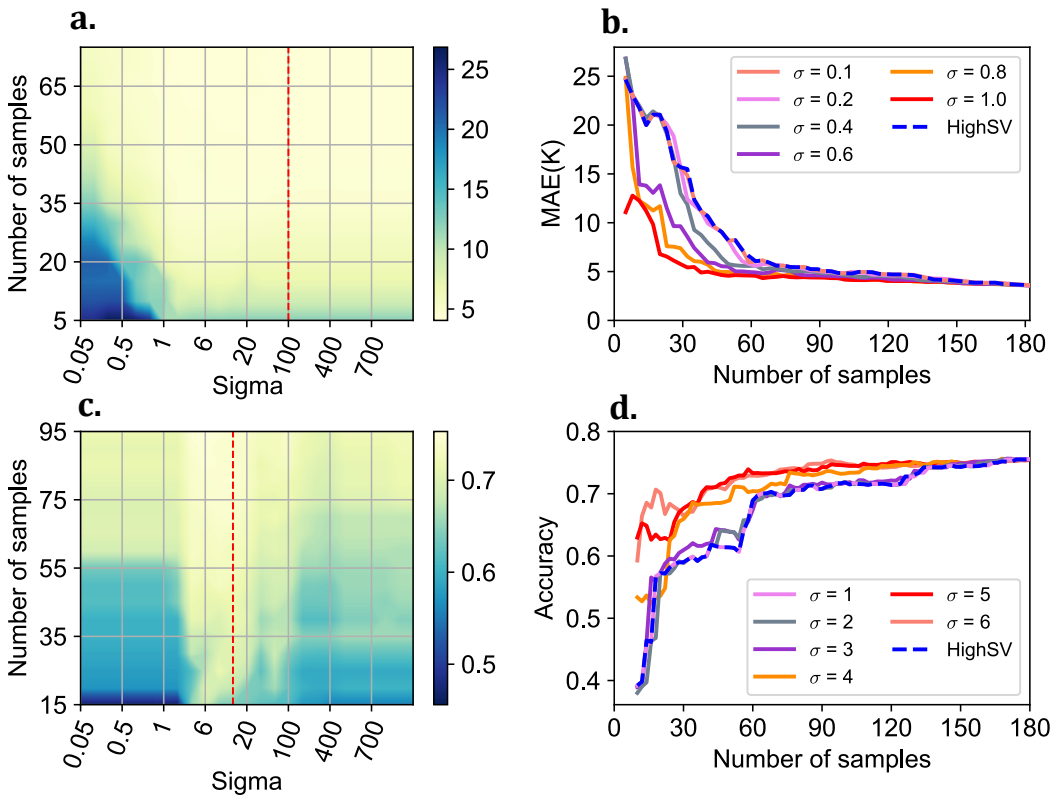


Figure 2.24: Kernel width analysis of HighAV sampling.

2.6.2.3 Robustness

Since the calculation of Shapley value always brings random errors, this section analyses the sensitivity of HighSV, HighAV and LowAV with five random trials of Shapley value function. Fig. 2.25a-b show robustness performance on the composite task and the Cifar10 task, respectively. It is clear that HighAV is far more stable and robust than HighSV. For HighSV, the slight random error of the Shapley value would change the samples significantly, thus reducing the stability and robustness. However, AV4Sam can aggregate the values of neighbouring samples by a kernel function, which plays the role of smoothing filter, so that HighAV can be less sensitive to the random error of Shapley value. The robustness of the proposed method enables the value function reuse and prior-knowledge-based value function in Scheme B,C,D.

2.7 Summary

Considering the expensive and time-consuming data labelling in manufacturing predictive modelling problems, sampling a smaller but informative dataset can potentially reduce labelling efforts. This chapter proposed an aggregation-value-based sampling (AV4Sam) strategy for optimal sample set selection for data-driven manufacturing applications. The proposed method has the appealing potential to reduce labelling efforts for machine learning problems. A novel aggregation value is defined to explicitly rep-

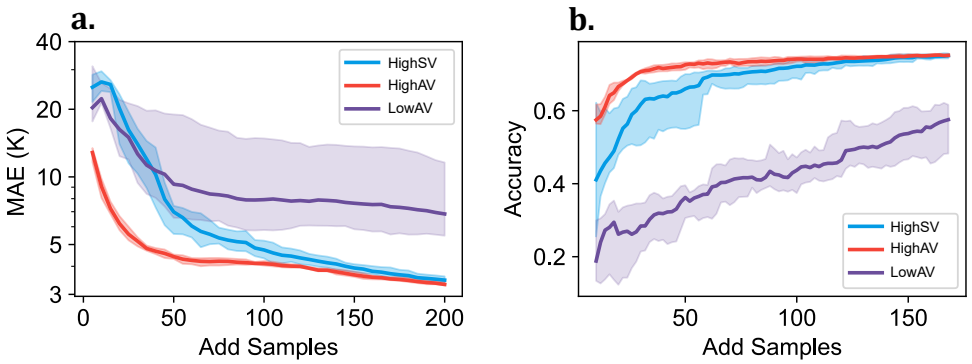


Figure 2.25: Robustness analysis of HighAV sampling.

resent the invisible redundant information as the overlaps of neighbouring values. The sampling problem is then recast as submodular maximisation on the aggregation value, which can be solved using the standard greedy algorithm.

Comprehensive experiments on several manufacturing datasets demonstrated that AV4Sam could achieve superior sampling performance compared with existing representative based sampling and value based sampling methods. Four schemes of value function show the generalisability of the proposed sampling methods. The selected optimised samples could provide more accurate and robust prediction results under the exact size of labelled data. On the other hand, the size of labelled data could be reduced by more than 50% under the fixed accuracy requirements. The detailed analysis of the feature distribution and aggregation value interprets the superiority of aggregation-value-based sampling.

AV4Sam can design a small but informative labelled dataset for further data modelling. When the sampled labelled dataset is insufficient to train a data-driven model, transferring the knowledge from similar or relative datasets can improve the performance of the target model. Chapter 3 will focus on transfer learning to leverage the auxiliary data and compensate for the insufficient information of the direct labelled data.

Chapter 3

Transfer learning based on structured distribution adaptation

La volonté trouve, la liberté choisit.
Trouver et choisir, c'est penser

– Victor Hugo

As mentioned in Section 1.3, when the direct labelled data is insufficient to train a model, leveraging auxiliary data can potentially compensate for the insufficient modelling information and reduce the requirements for a large amount of labelled data. This Chapter will focus on **objective** 2, i.e., knowledge transfer introduced in Fig. 1.30. The conditional distribution shift situation widely existing in MPM problems was investigated, and a structured distribution adaption method was proposed to facilitate the knowledge transfer from auxiliary data to the target task, thus improving the performance of the target model under data scarcity situations.

3.1 Introduction and Challenge analysis

Data-driven MPM continuously suffers from insufficient labelled data because of expensive and time-consuming experiments or simulations. Nevertheless, widely existing auxiliary data in manufacturing systems can afford transferable knowledge for the MPM problem to compensate insufficient direct labelled data. Learning the data-driven model of the MPM problem from a given dataset can be defined as the target task. In the machine learning field, the new paradigm, transfer learning, aims at transferring knowledge from a similar task to enhance the performance of the target task [173]. Therefore, the data from similar tasks can be defined as *auxiliary data* or *source data*, and the direct labelled data can be expressed as the *target data*.

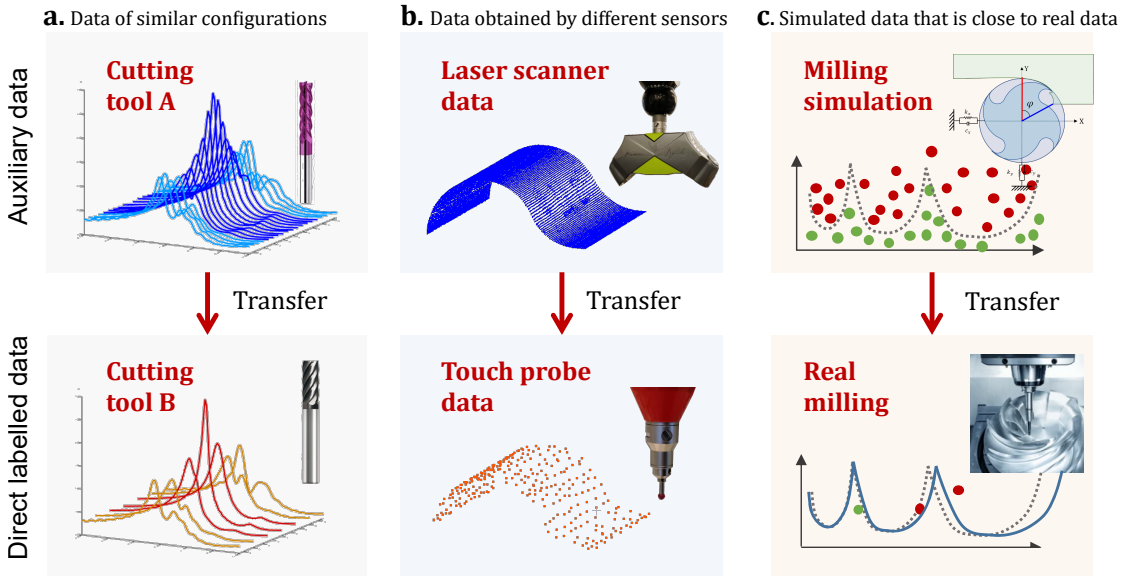


Figure 3.1: Transfer learning examples in manufacturing fields. (a) Transfer learning problem on pose-dependent tool tip dynamics prediction from tool A to tool B. (b) Multi-sensor surface measurement can be defined as the transfer learning problem from a large number of low-precision laser points to a small number of high-precision probe points. (c) Transfer learning problem on milling stability prediction from simulation data to real experimental data.

The auxiliary data in MPM problems can come from similar tasks, such as different but similar configurations, measurement data from different sensors or simulated data. Fig. 3.1a-c show the examples of transfer learning regression problems in manufactur-

ing fields. As shown in Fig. 3.1a, different milling tools have similar pose-dependent dynamics, therefore, transferring knowledge from existing data set of milling tool A can potentially reduce the data requirements of milling tool B. For the surface measurement problem shown in Fig. 3.1b, since it is time-consuming to collect sufficient touch probe points, the laser scanner data can act as the auxiliary data. Therefore, the multi-sensor measurement problem can be defined as a transfer learning setting, where the knowledge of the overall shape of the surface is transferred from the dataset of the laser scanner to the dataset of the touch probe. Fig. 3.1c shows a simulation-to-real transfer setting for the data-driven milling stability prediction problem, where the simulated stability diagram lobe is defined as the auxiliary data to compensate expensive real experimental stability data.

From a probabilistic point of view, transfer learning refers to minimising distribution discrepancy between the two tasks so as to extract common knowledge. Various transfer learning methods have been proposed according to different assumptions about the distribution shift between source and target data, including covariate shift [174, 175], prior probability shift [176, 177], sample selection bias [178, 179], and conditional distribution shift[180]. This research focused on conditional distribution shift regression situations that widely exist in MPM problems. In probability theory and statistics, the conditional distribution of y given $\mathbf{x}$ is the probability distribution of y when $\mathbf{x}$ is known to be a particular value, namely $p\left(y|\mathbf{x}\right)$ [177]. The different conditional distribution means that the two tasks have different probability distributions of y under the particular value of x, i.e. $p\left(y|\mathbf{x}\right)_s \neq p\left(y|\mathbf{x}\right)_t$. For example, the two cutting tools in Fig. 3.1a have the same feature space, namely the posture spaces of the machine tool, but different tip dynamics under any posture. The simulated and real stability diagrams in Fig. 3.1c have similar feature space, i.e. the cutting parameter combinations, but different stable boundaries under the same spindle speed. The above problems with the same feature space but different conditional distributions can be categorised into conditional distribution shift regression situations. As illustrated in Fig. 3.2, transfer learning under conditional shift aims to minimise the distribution discrepancy between the conditional distributions of the source data and target data, namely $\min \mathrm{dis}\left[p\left(y|\mathbf{x}\right)_s - p\left(y|\mathbf{x}\right)_t\right]$.

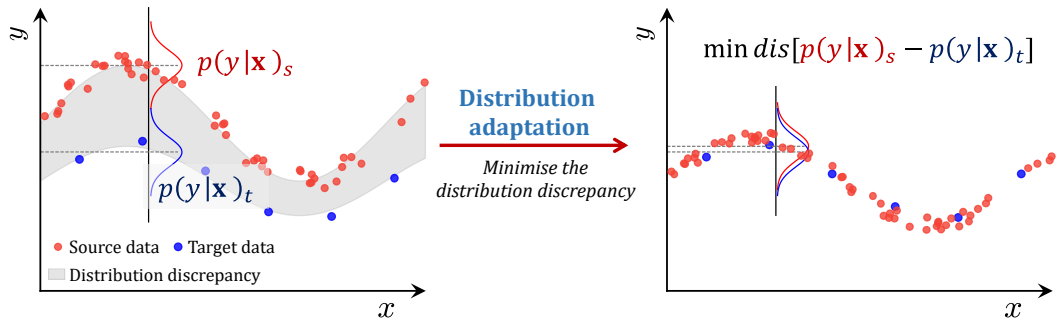


Figure 3.2: Illustration of conditional distribution adaptation.

The core challenges in conditional shift problems lie in the **representation and adaptation of the conditional distribution discrepancy** across tasks [139]. For classification problems, the conditional distribution shift can be represented using pseudo-labels because the label spaces maintain consistency between tasks [181]. However,

the predicted value for regression problems is a continuous variable, pseudo label-based methods are not applicable [182]. Thus, conditional shift regression problems are typically supervised problems, which require labelled target data [183].

Previous research about conditional shift regression problems can be categorised into mapping-based methods, weighting-based methods and parameter-based methods. Mapping-based methods aim to establish the mapping from the hypothesis function of the source task to the target task. The mapping function can be predefined according to the complexity of the distribution discrepancy, such as scale-offset transformation [184, 180], the residual function [185], or nonlinear mapping [186]. However, since the transformation function was trained only on the target data, the final performance highly depends on the amount of the labelled target data. Weighting-based methods adjust the importance of training data in two tasks to reduce the distribution discrepancy across tasks. The weights of instances could be estimated directly by Kullback–Leibler Inductive Transfer Learning (KLITL) [187], or adjusted adaptively during the iteration training such as TLB [188]. Parameter-based methods trained the target model using a few labelled data by adding a new parametric structure based on the reused task-independent framework [80, 189]. Since the information of the source data was contained in the reused framework, the whole model was only updated on limited labelled target data, which means that only the discretised conditional distribution near the training points was aligned, and there are no constraints on regions without target data [187, 188].

To summarise, previous conditional shift research only adapted the discretised distribution discrepancy near the labelled data without constraints on the entire feature space, a global distribution adaptation still required a large amount of labelled data. Therefore, representing and adapting the conditional distribution discrepancy with a small amount of labelled data remains as a significant challenge. This Chapter will describe a proposed new conditional distribution adaptation method by representing the distribution discrepancy in the latent space. The following Sections will first give the general idea of the proposed method, followed by the details of distribution discrepancy representation and adaptation.

3.2 General idea of structured conditional distribution adaptation

As shown in Fig. 3.3a, existing conditional adaptation methods directly adapts the distribution discrepancy of the few labelled data (the blue points). There are not enough constraints on the other region of the feature space that is without labelled data, which would lead to inevitable over-fitting during adaptation. In this research, the conditional distribution discrepancy is assumed to be generated by a finite number of latent variables, then **the discretised discrepancy on the original feature space could be transferred to structured discrepancy defined in the k-dimensional latent space**. After solving the distribution discrepancy of the latent space, the latent variable could be decoded to the original feature space to represent the distribution discrepancy globally. The structured distribution discrepancy representation could provide enough constraints

on the whole feature space, therefore, could lead to a stable adaptation result without requiring for too much labelled data. Fig. 3.3b illustrates the structured representation with Gaussian Mixture Model (GMM), in which the original distribution discrepancy is assumed to be controlled by three Gaussian distributions. Fig. 3.3c shows another structured representation with fuzzy rules, in which the original distribution discrepancy is assumed to be represented by multiple linear rules.

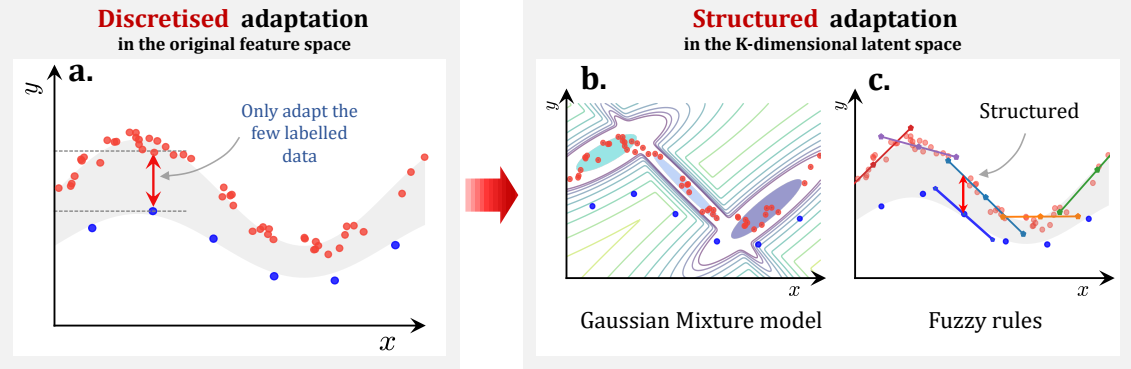


Figure 3.3: (a) Existing method: Discretised adaptation in the original feature space. (b) Structured adaptation in the K-dimensional Gaussian Mixture Model. (c) Structured adaptation in the K-dimensional fuzzy rules.

The proposed structured distribution adaptation consists of two main steps: (a) **the representation** and (2) **the adaptation** of structured distribution discrepancy. The representation step means representing the residual or the distribution discrepancy as the limited number of structured components in the defined latent space. This Chapter describes two structured representation schemes, Gaussian Mixture Model in Section 3.3.2 and fuzzy rules in Section 3.3.3. The adaptation step aims to design an optimisation target to minimise the distribution discrepancy on the latent space. A Conditional Embedding Operator Discrepancy (CEOD) is introduced in Section 3.4, and then generalised to fine-tuning deep learning settings in Section 3.4.1.

3.3 Structured distribution discrepancy representation

3.3.1 Conditional distribution shift problem definition

Assumming that a regression task for manufacturing predictive modelling with a small number of training data $\mathcal{D}_t = \left\{\left(\mathbf{x}_1^t, y_1^t\right), \ldots, \left(\mathbf{x}_{n_t}^t, y_{n_t}^t\right)\right\}$, defined as the target task, where the input feature is $\mathbf{x}_i^t \in \mathbb{R}^{1\times d}$, and $y_i^t \in \mathbb{R}^1$ is the corresponding output label, a continuous variable. The available auxiliary data is defined as the source data $\mathcal{D}_s = \left\{\left(\mathbf{x}_1^s, y_1^s\right), \ldots, \left(\mathbf{x}_{n_s}^s, y_{n_s}^s\right)\right\}$, where $\mathbf{x}_i^s$ is the input and y_i^s is the corresponding output. The size of the source data should be much larger than the target data, i.e. $n_s \gg n_t$. For the example shown in Fig. 3.1a, the tool tip dynamics dataset of cutting tool A is the source data, or auxiliary data while the tool tip dynamics dataset of cutting tool B is the target data. The task refers to learning a regression model from the input fea-

ture, namely the posture coordinates of a machine tool, to the output label, namely the dynamic parameters.

Supposing that the underlying hypothesis functions of the source task and the target task are f_s and f_t, respectively, then the observed training data can be modelled as:

$$y_i^s = f_s\left(\mathbf{x}_i^s\right) + \varepsilon_i^s \quad y_i^t = f_t\left(\mathbf{x}_i^t\right) + \varepsilon_i^t \tag{3.1}$$

It is assumed that the observed labels differ from the functions by additive noise ε_i^s and ε_i^t, and these noises follow a zero-mean Gaussian distribution $\varepsilon \sim \mathcal{N}\left(0, \sigma_n^2\right)$. From the probabilistic viewpoint, each observed training data can be written as conditional distribution as $p\left(y_i^s \mid \mathbf{x}_i^s\right)$ and $p\left(y_i^t \mid \mathbf{x}_i^t\right)$. The whole source and target dataset can be represented as $p\left(\mathbf{y}_s \mid \mathbf{X}_s\right)$ and $p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$ respectively, where $\mathbf{X}_s = \left[\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_{n_s}^s\right]^\top$, $\mathbf{y}_s = \left[y_1^s, y_2^s, \ldots, y_{n_s}^s\right]^\top$, $\mathbf{X}_t = \left[\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_{n_t}^t\right]^\top$ and $\mathbf{y}_t = \left[y_1^t, y_2^t, \ldots, y_{n_t}^t\right]^\top$. This research focused on the transfer learning problem for regression under a conditional shift situation, which means the identical marginal distributions $p\left(\mathbf{X}_s\right) = p\left(\mathbf{X}_t\right)$ but different conditional distributions $p\left(\mathbf{y}_s \mid \mathbf{X}_s\right) \neq p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$. Since the source data size is adequate, the source model f_s can be estimated directly with supervised machine learning algorithms. The goal of transfer learning is to learn the target function f_t to predict the labels $y^t \in Y_T$ for the target task.

Before the distribution adaptation, this Section will represent the distribution discrepancy between $p\left(\mathbf{y}_s \mid \mathbf{X}_s\right)$ and $p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$ by two structured components, i.e. Gaussian mixture model and fuzzy rules.

3.3.2 Discrepancy representation by Gaussian mixture model

Since the feature space and the marginal distributions of two tasks are consistent, the distribution discrepancy between two conditional distributions can be modelled as a residual function $h(\mathbf{x})$:

$$h(\mathbf{x}) = f_t(\mathbf{x}) - f_s(\mathbf{x}) \tag{3.2}$$

From the probabilistic viewpoint, the residual distribution can be represented as:

$$p(h \mid \mathbf{x}) = \mathcal{N}\left(f_t(\mathbf{x}) - f_s(\mathbf{x}), 2\sigma_n^2\right) \tag{3.3}$$

The most intuitive way to adapt the discrepancy is fitting the residual function $h(\mathbf{x})$ with only target data $\mathcal{D}_t$ as Ref [185]. However, the method only ensures that the conditional mean values near the training points are aligned, and there are no constraints on regions without target data. Because the target data is insufficient, directly fitting $h(\mathbf{x})$ can only adapt the discretised conditional discrepancy.

The most straightforward way to discretise the residual function is to solve the residual value of each sample of source data, namely $\mathbf{y}_s^{\text{new}} = \mathbf{y}_s + \mathbf{h}$, where $\mathbf{h} \in \mathbb{R}^{n_s}$ is a residual vector for source data, so that the new source data $\left\{\mathbf{y}_s^{\text{new}}, \mathbf{X}_s\right\}$ follows the similar conditional distribution with the target data $\left\{\mathbf{y}_t, \mathbf{X}_t\right\}$. Therefore, $\mathbf{h}$ can be obtained

by minimising the distribution discrepancy between the new source data and the target data as:

$$\mathbf{h} = \underset{\mathbf{h}\in\mathbb{R}^{n_s}}{\operatorname{argmin}} \operatorname{Dist}\left[p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right), p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)\right] \tag{3.4}$$

where $p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right)$ is the conditional distribution of the new source data after the compensation of the residual. Since the size of target training data is not enough to estimate the target function f_t, it is also fundamentally ill-posed to estimate residual vector $\mathbf{h}$ when n_t is far less than n_s. The optimisation of Eq. 3.4 would be unstable because there are infinitely many possible $\mathbf{h}$ that could have given rise to the observed limited target data. Therefore, it is reasonable to model the residual function in a parametric way to control the solution space. Since the source task and the target task share the same marginal distribution. The common marginal distribution $p(\mathbf{x})$ can be represented by a finite number of Gaussian distributions, where the k-th Gaussian distribution is characterised by the mixing coefficient π_k, mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{3.5}$$

where $\sum_{k=1}^{K} \pi_k = 1$. To illustrate the assignments of data points to specific components of the mixture model, a latent variable $\mathbf{z} \in \mathbb{R}^{K\times 1}$ is introduced, in which only one element equal to 1 and others are zero. For example, if a data point belongs to k-th component of the mixture, then $z_k = 1$ and $z_i = 0$, for $i \neq k$. Therefore, the probability of the assignment can be represented as $p\left(z_k = 1\right) = \pi_k$, therefore:

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k} \tag{3.6}$$

where $\prod$ refers to the product of the sequence $\pi_k^{z_k}$. Based on the latent variable $\boldsymbol{z}$, the marginal distribution $p(\mathbf{x})$ can be obtained by integrating the joint distribution as:

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x} \mid \mathbf{z}) \mathrm{d}\mathbf{z} \tag{3.7}$$

Note that the mixing coefficient π_k, mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$ can be solved by the Expectation Maximisation (EM) algorithm [17]. After building the Gaussian mixture model on $\mathbf{x}$, the distribution discrepancy on the feature space can be represented by marginalising the joint distribution over the latent variable $\mathbf{z}$ as:

$$\begin{aligned} p(h \mid \mathbf{x}) &= \int_{\mathbf{z}} p(h \mid \mathbf{z}) p(\mathbf{z} \mid \mathbf{x}) \mathrm{d}\mathbf{z} \\ &= \sum_{k=1}^{K} p\left(h \mid z_k = 1\right) p\left(z_k = 1 \mid \mathbf{x}\right) \end{aligned} \tag{3.8}$$

Denote w_k as the residual of the k-th component of the mixture, then the residual function $h(\mathbf{x})$ can be characterised as weighted superposition as:

$$h(\mathbf{x}) = \sum_{k=1}^{K} w_k p\left(z_k = 1 \mid \mathbf{x}\right) \tag{3.9}$$

where the posterior probability $p(\mathbf{z} \mid \mathbf{x})$ can be obtained by Bayes's theorem as:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} = \frac{\mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \prod_{k=1}^{K} \pi_k^{z_k}}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)} \tag{3.10}$$

Denote $\gamma\left(z_{ik}\right) = p\left(z_k = 1 \mid \mathbf{x}_i\right)$, then:

$$\gamma\left(z_{ik}\right) = \frac{\pi_k \mathcal{N}\left(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \Sigma_j\right)} \tag{3.11}$$

Now the residual function $h(\mathbf{x})$ can be represented as:

$$h(\mathbf{x}) = \sum_{k=1}^{K} w_k \gamma\left(z_k\right) \tag{3.12}$$

Till now, the problem of solving $\mathbf{h} \in \mathbb{R}^{n_s}$ is transferred to the problem of solving the component residual vector $\mathbf{w} \in \mathbb{R}^K$ on the structured Gaussian mixture model, which is more tractable because the marginal distribution can always be approximated using a limited number of Gaussian distributions with $K \ll n_s$. The new representation can significantly reduce the degree of freedom of the residual function. Then the properties of the latent variables $\mathbf{w}$ can be optimised by matching the kernel embedding conditional distributions between two tasks.

3.3.3 Discrepancy representation by fuzzy rules

This section will investigate another structured distribution discrepancy representation method based on Takagi-Sugeno-Kang (TSK) fuzzy systems. As shown in Fig. 3.4a, the source data is firstly represented by the TSK fuzzy rules. The distribution discrepancy can be defined as the residual terms on the fuzzy rules (Fig. 3.4b). Therefore, the distribution adaptation problem becomes solving the residual terms on the fuzzy rules. The target model can be represented based on the adapted fuzzy rules as shown in Fig. 3.4c. The background of the TSK fuzzy system will be introduced first, followed by the two steps of the fuzzy rules-based distribution representation method.

3.3.3.1 Takagi-Sugeno-Kang fuzzy system

TSK fuzzy system is an intelligent model defined with fuzzy logic and fuzzy rules [190]. Because of the interpretability and powerful approximation capabilities, TSK fuzzy system has been widely used for modelling complex non-linear systems [191]. Supposing that a TSK fuzzy system has d inputs $x_1 \in X_1, \ldots, x_d \in X_d$, one output y $\in Y$ and K rules,then the structure of K-th TSK fuzzy rule for the system consists of IF-THEN in the form:

$$\begin{array}{l} \text{IF} x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \ldots \wedge x_d \text{ is } A_d^k \\ \text{THEN } y \text{ is } r_k(x) \quad k = 1, 2, \ldots, K \end{array} \tag{3.13}$$

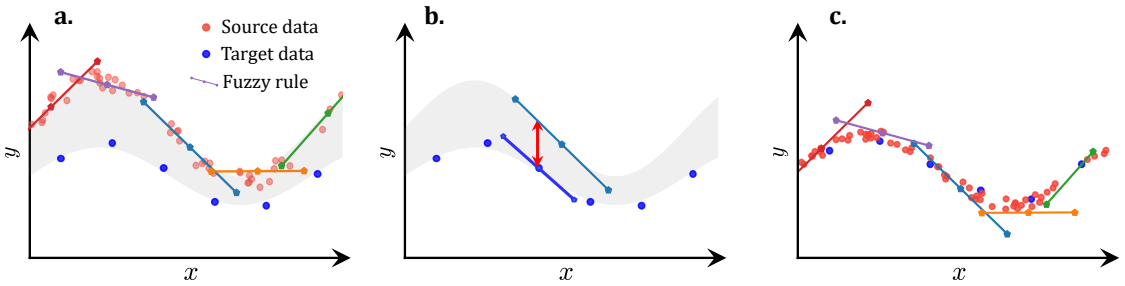


Figure 3.4: Transfer learning based on fuzzy rules. (a) Represent the source data with fuzzy rules. (b) Evaluate the residual terms on the fuzzy rules. (c) Reconstruct the target model.

Where A_j^k is the fuzzy set of j-th input dimension under the k-th. A fuzzy set is described by its membership function $\mu_{A_j^k}(x)$ as:

$$A_j^k = \left\{ \left(x_j, \mu_{A_j^k}(x_j) \right) | x_j \in X_j \right\} \tag{3.14}$$

The shape of membership function is determined up to the designer. The commonly used membership function is the Gaussian membership function:

$$\mu_{A_j^k}(x_j) = \exp\left(\frac{-\left(x_j - c_j^k\right)^2}{2\sigma_j^k} \right) \tag{3.15}$$

Where c_j^k and σ_j^k are the center value and the spread of the fuzzy set A_j^k. These two parameters are defined as the antecedent parameters, which can be calculated by fuzzy partition algorithms such as Fuzzy c-means (FCM) [192].

The membership degree of fuzzy set A^k can be the product of membership degrees of each dimension as:

$$\mu_k(\mathbf{x}) = \prod_{j=1}^{d} \mu_{A_j^k}(x_j) \tag{3.16}$$

The normalized membership degree is given by:

$$\lambda_k(\mathbf{x}) = \frac{\mu_k(\mathbf{x})}{\sum_{k=1}^{K} \mu_k(\mathbf{x})} \tag{3.17}$$

Where λ_k means the degree of activation of the K-th rule. Then the output of TSK fuzzy system can be formulated by a combination of sub-models as [190]:

$$y = \sum_{k=1}^{K} \lambda_k(\mathbf{x}) r_k(\mathbf{x}) \tag{3.18}$$

3.3.3.2 Constructing the TSK fuzzy system of the source task

As defined in Section 3.3.1, for the source data $\mathcal{D}_s$, denote $\mathbf{X}_s \in \mathbb{R}^{n_s \times d}$ as the matrix having the vectors of input features and $\mathbf{y}_s \in \mathbb{R}^{n_s \times 1}$ as the vector containing the output

label:

$$\mathbf{X}_s = \left[\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_{n_s}^s\right]^\top, \mathbf{y}_s = \left[y_1^s, y_2^s, \ldots, y_{n_s}^s\right]^\top \tag{3.19}$$

TSK fuzzy system can represent nonlinear dynamical systems with TSK fuzzy rules with high precision. Denote the TSK fuzzy system of the source task as FS_s, then the output of FS_s can be represented by a combination of series sub-models as:

$$\mathrm{FS}_s(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k(\mathbf{x}) r_k(\mathbf{x}) \tag{3.20}$$

where K is the number of fuzzy rules, $\lambda_k(\mathbf{x})$ is the normalized membership degree which can be calculated with antecedent parameters as Eq. 3.17. The antecedent parameters are obtained by fuzzy partition [193], which can be regarded as reflections of the marginal distribution of $\mathcal{X}$. Because of the assumption that $p(\mathbf{X}_s) = p(\mathbf{X}_t)$, the antecedent parameters can be shared across tasks to preserve the distribution information of source data. $r_k(\mathbf{x})$ is the k-th fuzzy rule, a linear function of the input variables.

Denote $\mathbf{p}^k \in \mathbb{R}^{(d+1)}$ as the linear coefficient vector of $r_k(\mathbf{x})$, then the consequent parameters of the TSK fuzzy system is $\mathbf{p} = \left[\mathbf{p}^1\ \mathbf{p}^2\ \cdots\ \mathrm{p}^K\right]^\top \in \mathbb{R}^{(d+1)K\times 1}$. The consequent parameters of FS_s can be solved with labelled dataset $\mathcal{D}_s$.

To solve the consequent $\mathbf{p}$, donate $\mathbf{X}_x^p \in \mathbb{R}^{n_s\times(d+1)K}$ as:

$$\mathbf{X}_x^p = [\Gamma_1\mathbf{X}_s^e, \Gamma_2\mathbf{X}_s^e, \ldots, \Gamma_K\mathbf{X}_s^e] \tag{3.21}$$

where $\mathbf{X}_s^e = [\mathbf{X}_s,\ \mathbf{1}] \in \mathbb{R}^{n_s\times(d+1)}$ is the extended matrix by appending a unitary column to $\mathbf{X}_s$. $\Gamma_k \in \mathbb{R}^{n_s\times n_s}$ is a diagonal matrix having the normalized membership degree $\lambda_k(\mathbf{x}_i)$ as its i-th diagonal element:

$$\Gamma_k = \begin{bmatrix} \lambda_k(\mathbf{x}_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_k(\mathbf{x}_{n_s}) \end{bmatrix} \tag{3.22}$$

Then the consequent parameters $\mathbf{p} \in \mathbb{R}^{(d+1)K}$ can be solved with the least square method as :

$$\mathbf{p} = \left[\left(\mathbf{X}_x^p\right)^\top \mathbf{X}_x^p\right]^{-1} \left(\mathbf{X}_x^p\right)^\top \mathbf{y}_s \tag{3.23}$$

After that, the TSK fuzzy system of the source data, FS_s, is completely constructed and can be used to represent the distribution discrepancy.

3.3.3.3 Constructing the target model with fuzzy residual

Given that there are only small amounts of labelled target data, the TSK fuzzy system of the target task FS_t cannot be constructed directly. The antecedent parameters of the source fuzzy system to can be reused preserve the marginal distribution properties of

the source data. For the labelled target data $\mathcal{D}_t$, denote $\mathbf{X}_t \in \mathbb{R}^{n_t \times d}$ and $\mathbf{y}_t \in \mathbb{R}^{n_s \times 1}$ as the input and output respectively:

$$\mathbf{X}_t = \left[\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_{n_t}^t\right]^\top, \cdot \mathbf{y}_t = \left[y_1^t, y_2^t, y_{n_t}^t\right]^\top \tag{3.24}$$

For an instance $(\mathbf{x}_i^t, y_i^t)$ in $\mathcal{D}_t$, the output of FS_s is

$$\mathrm{FS}_s\left(\mathbf{x}_i^t\right) = \sum_{k=1}^{K} \lambda_k\left(\mathbf{x}_i^t\right) r_k\left(\mathbf{x}_i^t\right) \tag{3.25}$$

Since conditional distributions of the two datasets are different, i.e. $p\left(\mathbf{y}_s \mid \mathbf{X}_s\right) \neq p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$, thus $h_{\mathrm{FS}_s}\left(\mathbf{x}_i^t\right) \neq y_i^t$. Assuming the distribution discrepancy can be represented by the residual term defined on the fuzzy rules. Therefore, the target fuzzy system FS_t can be constructed by appending a residual term w_k to each fuzzy rule of FS_s as:

$$\mathrm{FS}_t(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k(\mathbf{x})\left(r_k(\mathbf{x}) + w_k\right) \tag{3.26}$$

where the antecedent parameters $\lambda_k\left(\mathbf{x}\right)(k = 1, \ldots, K)$ are reused to preserve the distribution properties of the source data while the residual term $z_k(k = 1, \ldots, K)$ can be used to represent the conditional distribution discrepancy between$p\left(\mathbf{y}_s \mid \mathbf{X}_s\right)$ and $p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$, therefore, $h_{\mathrm{FS}_t}\left(\mathbf{x}\right)$ could be decomposed into the following representation as

$$\mathrm{FS}_t(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k(\mathbf{x}) r_k(\mathbf{x}) + h(\mathbf{x}) \tag{3.27}$$

The former component is the TSK fuzzy system of source task, but it can be easily replaced by any supervised regression model. The latter term can be defined as the residual function that represents the distribution discrepancy:

$$h(\mathbf{x}) = \sum_{k=1}^{K} \lambda_k\left(\mathbf{x}\right) w_k \tag{3.28}$$

where w_k is the to-be-solved residual term of each fuzzy rule.

The fuzzy residual function has the consistent form as the Gaussian Mixture Model residual function in Eq. 3.12. The residual function $h(\mathbf{x})$ can be easily represented by solving the component residual vector $\mathbf{w} \in \mathbb{R}^K$ on the structured fuzzy rules.

Although Gaussian Mixture Model and fuzzy system schemes have similar structured distribution discrepancy representation results, they have different characteristics. Implementing the fuzzy system-based method requires fuzzy partition, which is generally more stable for problems with small-size data and low-dimensional feature spaces. The tool tip dynamics problem of the case study section will adopt the fuzzy rules solution. By comparison, Gaussian Mixture Model has more powerful implementation approaches even for large-size noisy data, and there are also more open-source libraries, such as scikit-learn [194]. The multi-sensor measurement case will adopt Gaussian Mixture Model because the dataset consists of more than 30000 noisy points.

3.4 Conditional distribution discrepancy adaptation

After the distribution discrepancy representation based on Gaussian Mixture Model or fuzzy rules, the structured residual terms can be solved by adapting the conditional distribution. It is known that kernel embedding of marginal distribution is widely used in transfer learning problems under covariate shift situations [195]. However, kernel embedding for conditional shift problems has not drawn much attention. This Section describes a proposed new measure, Conditional Embedding Operator Discrepancy (CEOD), to adapt the conditional distribution discrepancy based on the kernel embedding theory. After that, the new measure is generalised to deep learning scenarios by defining a hybrid loss function.

3.4.1 Embedding representation of distribution

Distribution adaptation in transfer learning aims to minimise distribution discrepancy. The quantified criterion that can measure the discrepancy between distributions is defined as the distance between distributions [177]. In machine learning and statistics, many criteria are developed to measure the distribution distance, such as KL-divergence and JS-divergence [196]. However, most methods rely heavily on density estimation, a well-known complicated mathematical statistics problem [197]. Kernel embedding emerges as a new distance estimation method that can avoid density estimation [198]. The key idea is to map the conditional distribution into a reproducing kernel Hilbert space, and then the distance between distributions is available via simple feature space operations [199]. This section will first introduce the basic idea of kernel embedding of distribution.

In statistical hypothesis testing, a two-sample test aims to determine whether the discrepancy between these two distributions is statistically significant. The kernel embedding can quantify the discrepancy between two sets of distributions. Supposing that dataset X_p and X_q are sampled from P and Q, respectively, the distance between the distribution P and Q can be evaluated directly by comparing X_p and X_q. For example, if P and Q are the same distribution, they should have the same expectation value, namely the first moment of distribution:

$$P = Q \quad \Rightarrow \quad \|\mathbb{E}[X_p] - \mathbb{E}[X_q]\| = 0 \tag{3.29}$$

However, the same expectation does not mean the same distribution.

$$\|\mathbb{E}[X_p] - \mathbb{E}[X_q]\| = 0 \quad \not\Rightarrow \quad P = Q \tag{3.30}$$

which means that the equal first moment is necessary but insufficient. The norm distance between the first moment cannot represent the actual distance between the two distributions. However, if both the first and second moments, namely $\mathbb{E}[(X)], \mathbb{E}[(X^2)]$, are equal, the two distributions may have a closer distance. Furthermore, if the infinite number of moments of the two distributions are the same, it can be proved that $P = Q$

[198]. Fig. 3.5 illustrates the distribution embedding. The left figure means two different distributions, P and Q. It is challenging to measure the distance between the two distributions directly. After embedding the distributions into a new space, the norm between the new representations, namely the expectations, can be used to evaluate the distance between distributions. The space that carries the infinite number of moments is a Reproducing Kernel Hilbert Space.

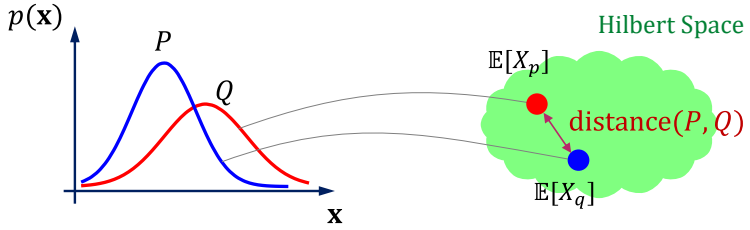


Figure 3.5: Embedding of distributions into a Hilbert space via an expectation operation, revised from. [198]

Suppose a feature map $\varphi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ refers to an reproduced kernel Hilbert space. Therefore $\varphi(X)$ refers to the infinite-dimensional feature of X, for example, $\varphi(X) = [X, X^2, X^3, \cdots]$. If the expectations of the infinite-dimensional representations $\varphi(X)$ of the two distributions are the same, it can also be proved that $P = Q$ [198], namely:

$$\|\mathbb{E}[\varphi(X)] - \mathbb{E}[\varphi(X_2)]\|_{\mathcal{H}}^2 = 0 \quad \Longrightarrow \quad P = Q \tag{3.31}$$

where $\|\cdot\|_{\mathcal{H}}^2$ refers to the 2-norm defined on the space $\mathcal{H}$. Therefore, the norm distance between the map expectation can be naturally defined as the distribution distance. Maximum Mean Discrepancy (MMD), one of the most famous measures for adapting the marginal distribution discrepancy, is then defined as:

$$\begin{aligned} \text{MMD}(P, Q) &= \|\mathbb{E}[\varphi(X_p)] - \mathbb{E}[\varphi(X_q)]\|_{\mathcal{H}}^2 \\ &= \left\|\mathcal{U}_{X_p} - \mathcal{U}_{X_q}\right\|_{\mathcal{H}}^2 \end{aligned} \tag{3.32}$$

where $\mathcal{U}_{X_p}$ and $\mathcal{U}_{X_q}$ are known as distribution embedding operator. The distribution discrepancy is now represented as a scalar value, which can then be directly defined as the optimisation target for the distribution adaptation.

MMD is widely used in the transfer learning field for marginal distribution adaptation [195, 136]. In contrast, there is still a lack of effective measures for the conditional distribution shift regression problems.

3.4.2 Conditional embedding operator discrepancy

In this Section, a Conditional Embedding Operator Discrepancy (CEOD) was designed to measure the conditional distribution discrepancy based on the theory of kernel embedding of conditional distributions.

The distribution discrepancy between the source and target tasks is represented by the component residual vector $\mathbf{w}$ in the previous section. To adapt the distribution discrepancy, $\mathbf{w}$ can be estimated by minimising the distribution distance between the new source data and the target data as:

$$\mathbf{w} = \underset{\mathbf{w}\in\mathbb{R}^K}{\operatorname{argmin}} \operatorname{Dist}\left[p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right), p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)\right] \tag{3.33}$$

where $\mathbf{y}_s^{new} = \mathbf{y}_s + \boldsymbol{\gamma}_s\mathbf{w}, \boldsymbol{\gamma}_s \in \mathbb{R}^{n_s\times K}$ is the posterior probability matrix whose element is $\gamma\left(z_{ik}\right) = p\left(z_k = 1 \mid x_i^s\right)$ that defined in Eq. 3.11.

Similar to MMD, the inspiration for conditional distribution discrepancy comes from the field of hypothesis testing, a conditional embedding operator is proposed by Song et al. [200], which embeds $p(\mathbf{y} \mid \mathbf{X})$ into RKHS and makes it possible to measure the distance between two conditional distributions.

According to the distribution embedding theory [198], the conditional distribution $p(\mathbf{y} \mid \mathbf{X})$ can be represented by conditional embedding operator as [199]:

$$\mathcal{U}_{\mathbf{y}|\mathbf{X}} = \mathbb{E}_{\mathbf{y}\mathbf{X}}[\varphi(\mathbf{y}) \otimes \phi(\mathbf{X})]\mathbb{E}_{\mathbf{X}\mathbf{X}}^{-1}[\phi(\mathbf{X}) \otimes \phi(\mathbf{X})] \tag{3.34}$$

where $\otimes$ is the tensor product, $\mathbb{E}$ is the expectation operation, $\varphi(\cdot)$ and $\phi(\cdot)$ are feature maps of variables $\mathbf{x}$ and y respectively. This operator is very similar to the embedding operator of the marginal distribution shown in Eq. 3.32.

Then the empirical estimation of the conditional embedding operators of $p(\mathbf{y}_t \mid \mathbf{X}_t)$ can be represented as:

$$\begin{aligned}\widehat{\mathcal{U}}_{\mathbf{y}_t|\mathbf{X}_t} &= \frac{1}{n_t}\sum_{i=1}^{n_t}\varphi\left(y_i^t\right)\phi\left(\mathbf{x}_i^t\right)\frac{1}{n_t}\sum_{i=1}^{n_t}\phi\left(\mathbf{x}_i^t\right)\phi\left(\mathbf{x}_i^t\right) \\ &= \varphi\left(\mathbf{y}_t\right)\left(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\phi^\top\left(\mathbf{X}_t\right)\end{aligned} \tag{3.35}$$

where $\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t}$ is the kernel matrix of $\mathbf{X}_t$, λ_c is the regularisation parameter to ensure the invertibility of the kernel matrix. Similarly, the empirical estimation of $p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right)$ can be given by:

$$\begin{aligned}\hat{\mathcal{U}}_{\mathbf{y}_s^{\text{new}}|\mathbf{X}_s} &= \frac{1}{n_s}\sum_{i=1}^{n_s}\varphi\left(y_i^S\right)\phi\left(\mathbf{x}_i^t\right)\frac{1}{n_s}\sum_{i=1}^{n_s}\phi\left(\mathbf{x}_i^S\right)\phi\left(\mathbf{x}_i^S\right) \\ &= \varphi\left(\mathbf{y}_s^{new}\right)\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\phi^\top\left(\mathbf{X}_s\right)\end{aligned} \tag{3.36}$$

The distance between the two distributions can be represented as the Conditional Embedding Operator Discrepancy (CEOD) [199]:

$$\operatorname{CEOD}(p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right), p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)) = \left\|\hat{\mathcal{U}}_{\mathbf{y}_s^{\text{new}}|\mathbf{X}_s} - \hat{\mathcal{U}}_{\mathbf{y}_t|\mathbf{X}_t}\right\|_{\mathcal{H}}^2 \tag{3.37}$$

It can be further simplified as:

$$\begin{aligned}
\text{CEOD} &= A + B - 2C \\
A &= \text{Tr}\left[\phi\left(\mathbf{X}_s\right)\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\phi^{\top}\left(\mathbf{X}_s\right)\right] \\
&= \text{Tr}\left[\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s}\right] \\
B &= \text{Tr}\left[\phi\left(\mathbf{X}_t\right)\left(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}^t\left(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\phi^{\top}\left(\mathbf{X}_t\right)\right] \\
&= \text{Tr}\left[\left(\mathbf{K}_{\mathbf{X}_t\mathbf{x}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}^t\left(\mathbf{K}_{\mathbf{X}_t}\mathbf{X}_t + n_t\lambda_c\mathbf{I}\right)^{-1}\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t}\right] \\
C &= \text{Tr}\left[\phi\left(\mathbf{X}_s\right)\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}^c\left(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\phi^{\top}\left(\mathbf{X}_t\right)\right] \\
&= \text{Tr}\left[\left(\mathbf{K}_{\mathbf{X}_s\mathbf{X}_s} + n_s\lambda_c\mathbf{I}\right)^{-1}\tilde{\mathbf{K}}^c\left(\mathbf{K}_{\mathbf{X}_t\mathbf{X}_t} + n_t\lambda_c\mathbf{I}\right)^{-1}\mathbf{K}_{\mathbf{X}_t\mathbf{X}_s}\right]
\end{aligned} \tag{3.38}$$

where $\tilde{\mathbf{K}} = \mathbf{K}_{\mathbf{y}_s^{\text{new}}\mathbf{y}_s^{\text{new}}}, \tilde{\mathbf{K}}^t = \mathbf{K}_{\mathbf{y}_t\mathbf{y}_t}, \tilde{\mathbf{K}}^c = \mathbf{K}_{\mathbf{y}_s^{\text{new}}\mathbf{y}_t}$ are kernel matrix.

To achieve a stable optimisation of $\mathbf{w}$, the final loss function is defined as the combination of the CEOD and the traditional MSE loss of regression:

$$\mathcal{L}(\mathbf{w}) = \beta\,\text{CEOD}(p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right), p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)) + (1-\beta)\mathcal{L}_{MSE} + \lambda_r\mathbf{w}^{\top}\mathbf{w} \tag{3.39}$$

where λ_r is the penalty term, and β is the trade-off parameter to control the influence of different terms. The MES loss $\mathcal{L}_{MSE}$ is defined as:

$$\mathcal{L}_{MSE}^t = \frac{1}{n_t}\sum_{i=1}^{n_t}\left(f\left(\mathbf{x}_i^t\right) - y_i^t\right)^2 \tag{3.40}$$

The component residual vector $\mathbf{w}$ can then be optimised by minimising the defined loss function with gradient descent. The conditional distributions $p\left(\mathbf{y}_s^{\text{new}} \mid \mathbf{X}_s\right)$ and $p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)$ are aligned so that the new source data can be merged with the target data as $\{\mathbf{X}_s, \mathbf{y}_s^{\text{new}}\} \cup \{\mathbf{X}_t, \mathbf{y}_t\}$. Then the target model can be trained on the merged data sets using a general supervised learning algorithm.

The main contribution of this Section is the defined loss function in Eq. 3.39, which can be used to adapt the conditional distribution discrepancy for different machine learning models, including deep transfer learning. For a given pre-defined source model, the model parameters can be fine-tuned by minimising the CEOD loss function on the target data. The case study Section 3.5 will validate the proposed CEOD in different machine learning models.

3.5 Case study

The proposed transfer learning method consists of structured conditional distribution representation and distribution adaptation. Therefore, several case studies were carried out to demonstrate the effectiveness of both representation and adaptation.

3.5.1 Case study one: Tool dynamics prediction

3.5.1.1 Problem statement

This case study focused on the pose-dependent tool tip dynamics (TCP) of a five-axes machine center. The dynamic parameters of the cutting tool are necessary inputs to the milling dynamics milling model. They have a significant impact on the stability of the milling process as well as the surface quality of the machined part. However, the dynamics typically vary with the changing postures of the machine tool axis during the machining process. The accurate prediction of the pose-dependent tool tip dynamics has become one of the critical challenges for the milling process stability control [24]. Considering the time-consuming experimental data collection, transferring TCP data from a similar tool to the target task can reduce labelling efforts.

Fig. 3.6a shows the impact test experiments for collecting the tool tip dynamics. The tool tip was first excited by an impact hammer, and the response signal was then collected by the accelerometer to calculate the Frequency Response Function (FRF). After that, the key dynamic parameters, including natural frequency ω, damping ratio ξ, and stiffness K, were identified from the FRF, as shown in Fig. 3.6b.

Fig. 3.6c illustrates transferring the dynamics from one cutting tool to another. By leveraging the knowledge from the source data, the data-driven TCP dynamics model of the target can be built with only a limited number of labelled data.

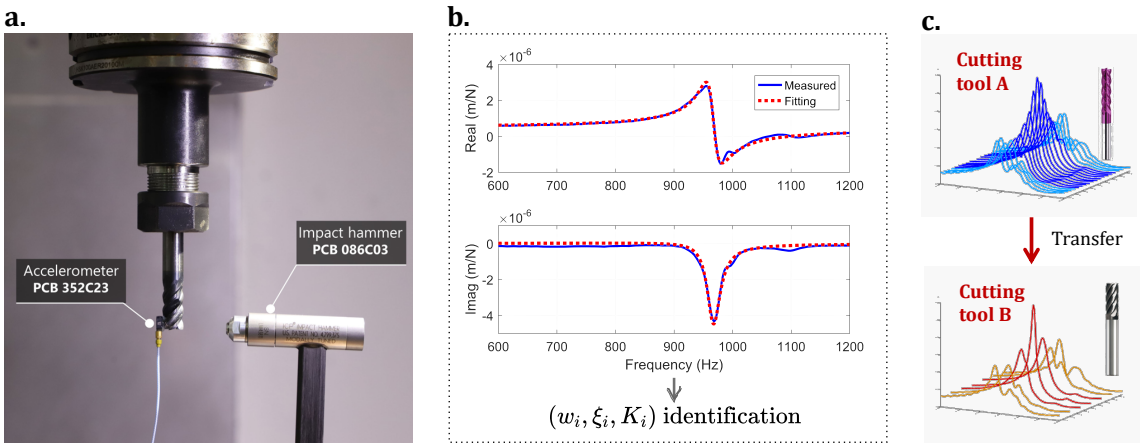


Figure 3.6: The problem statement of the TCP case study. (a) The impact test experiments for TCP dynamics collection. (b) Frequency response function. (c) Transfer learning settings.

Tool tip dynamics usually change with the coordinates of the machine tool axes, including three linear axes (X, Y, Z) and two rotating axes (A, C). Therefore, the modelling task aims to learn the map from (X, Y, Z, A, C) to (ω, ξ, K). The dataset TCP was constructed in the author's laboratory by collecting the dynamics data of two milling tools (T_A, T_B) for 381 coordinates of a five-axes machining centre manually [173]. T_A is a 3-flute cutter with a diameter of 8mm and a length of 65mm, while T_B is a 2-flute cutter with a diameter of 10mm and a length of 85mm. The multivariate random variables $\mathbf{x}$ is the coordinates of the machine centre and three labels (ω, ξ, K) constitute three

prediction tasks: $\omega = f_\omega(\mathbf{x}), \xi = f_\xi(\mathbf{x}), K = f_K(\mathbf{x})$. The two tools and three prediction tasks can construct six transfer learning tasks: $f_\omega : T_A \to T_B, f_\omega : T_B \to T_A, f_\xi : T_A \to T_B, f_\xi : T_B \to T_A, f_K : T_A \to T_B, f_K : T_B \to T_A$.

Fig. 3.7 shows the FRFs of T_A and T_B with axis A from $-90°$ to $90°$. As Aix A goes from $90°$ to $0°$, both sets of FRFs change from flat to steep and reach their maximum amplitude at $0°$. The similar changing trends of the T_A and T_B can provide explainable transferability for transfer learning performance.

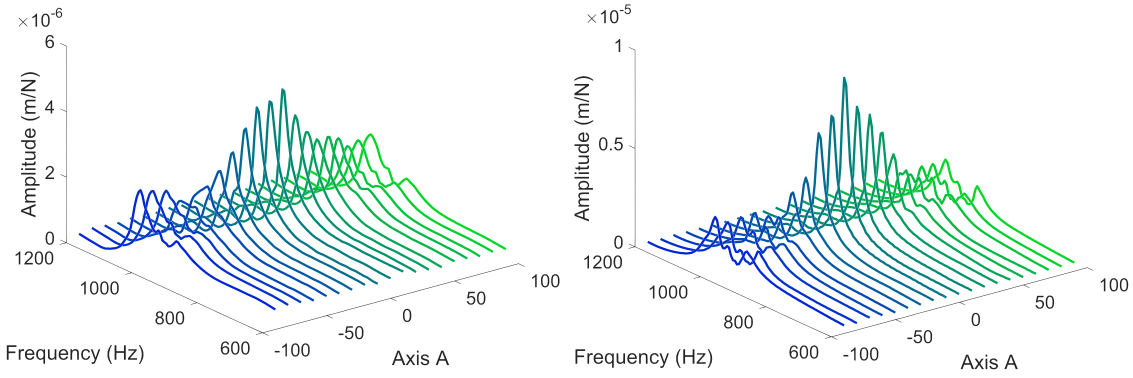


Figure 3.7: The similar changing trends of the two TCP tasks.

3.5.1.2 Experimental settings

In this case, the conditional distribution discrepancy between two datasets was represented and adapted based on the fuzzy system scheme introduced in Section 3.3.3. The hyperparameter, including the number of fuzzy rules K, the kernel width, and others, were selected based on the five-fold cross-validation of the source data. The number of fuzzy rules was set to $K = 8$. Because of the similarity in TCP dynamics, it is possible to reduce the labelled data requirements for the target task. Therefore, the target data size is very insufficient, namely $n_t = 10\% \times n_s$. The proposed method is uniformly marked as CDA (Conditional Distribution Adaptation) for simplicity of expression. To evaluate the performance of the proposed method, several state-of-art methods were implemented for comparison:

- **Residual Approximation** (RA)[185]: The RA method builds the target model by computing the offset between the target data and the source model with GP. This method is widely used in condition shift situations, such as the multi-source fusion problems [185] or multi-sensor surface measurement [201]. The residual function of RA was trained on the target labelled data directly without using the distribution information of the source data.

- **Transfer learning by boosting** (TLB)[188]: The TLB method, named as Two-stage TrAdaBoost.R2, which is a famous boosting-based regression transfer algorithm, trains the source data and target data together by adaptively adjusting the instance weights.

- **General Transformation Function** (GTF)[184]: GTF is an algorithm-dependent hypothesis transfer learning method which characterises the relationship between the source and the target tasks by establishing a general transformation function.
- **Domain adaptation under generalised target shift** (GeTarS) [180]: GeTarS proposes to resample the source instances by reweighting or transforming to reproduce the distribution on the target task. And the marginal distribution and conditional distribution are embedded in a reproducing kernel Hilbert space.
- **Target**: GP prediction using only target data.
- **Source**: GP prediction using only source data.

3.5.1.3 Experimental results

Tab. 3.1 shows the experimental results, the mean absolute error (MAE) and the corresponding Standard Deviation (STD) of 6 transfer learning tasks with target data size $n_t = 10\% \times n_s$. All results were the average values of 20 repeated trials with randomly selected target data. The lowest MAEs of each transfer task is marked in bold. CDA performed better than other methods in most tasks (5/6 tasks). For the two learning tasks of ω, CDA achieved an MAE of 5.24 Hz and 6.82 Hz, which was much less than other methods.

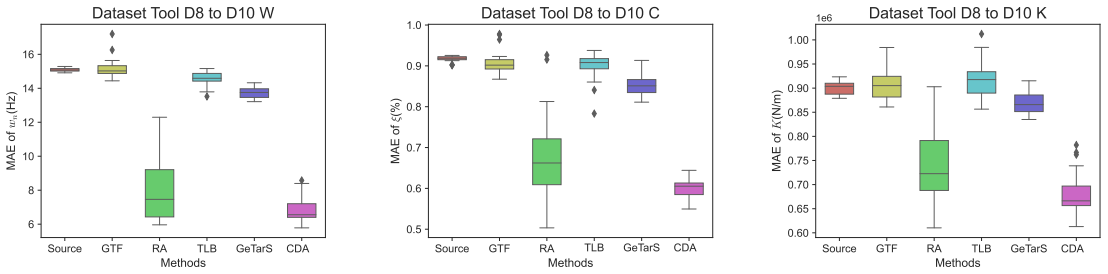
Table 3.1: Transfer learning performance for TCP case.

Data	GTF	RA	TLB	GeTarS	CDA
$f_\omega : T_B \to T_A$	13.98 ± 0.72	6.06 ± 0.58	13.25 ± 0.51	12.24 ± 1.09	**5.24 ± 0.22**
$f_\xi : T_B \to T_A$	0.71 ± 0.03	0.55 ± 0.05	0.84 ± 0.05	0.64 ± 0.05	**0.53 ± 0.04**
$f_K : T_B \to T_A$	0.76 ± 0.04	**0.51 ± 0.07**	0.90 ± 0.05	0.70 ± 0.04	0.56 ± 0.04
$f_\omega : T_A \to T_B$	15.22 ± 0.44	8.26 ± 1.84	14.64 ± 0.36	13.76 ± 0.25	**6.82 ± 0.57**
$f_\xi : T_A \to T_B$	0.91 ± 0.03	0.67 ± 0.09	0.91 ± 0.01	0.85 ± 0.03	**0.60 ± 0.02**
$f_K : T_A \to T_B$	0.91 ± 0.03	0.73 ± 0.06	0.92 ± 0.02	0.87 ± 0.02	**0.68 ± 0.04**

Note: The best results for each transfer learning task are marked in bold. The MAE unit for the natural frequency ω is Hz. The MAE unit for the damping ratio ξ is %. The MAE unit for the stiffness K is 1e6N/m.

To compare the robustness of different methods, Fig. 3.8 shows the boxplots of MAEs constructed from 20 trials with random target data. Each boxplot consists of the lower to upper quartile values of the data, with a line at the median. In Fig. 3.8a-c, RA can achieve the minimum MAE by the lower quartile values, but the long box means that RA is more sensitive and unstable. In contrast, CDA can provide lower MAE and maintain robustness.

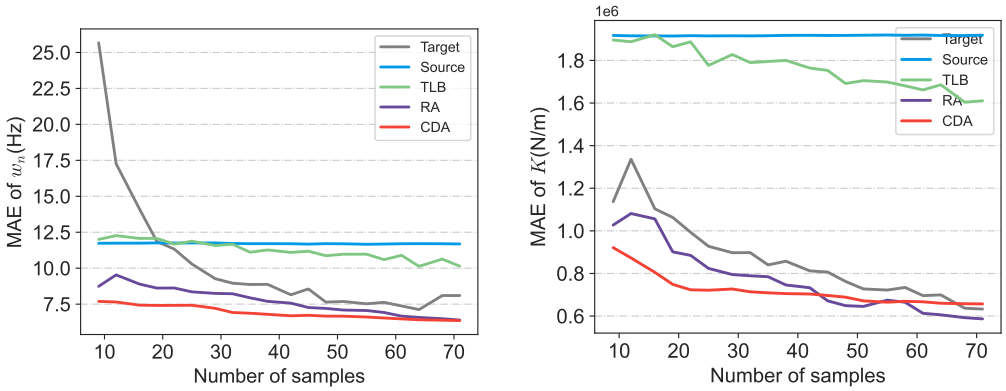
To further reveal the relationship between the transfer learning performance and the size of the target data, all transfer learning methods were carried out with the size of the target data from 20 to 70. Fig. 3.9 shows the relationship between the size of the target data and the MAEs for the task $f_\omega : T_A \to T_B$ and $f_W : T_A \to T_B$.



(a) MAEs of $f_\omega : T_A \to T_B$. (b) MAEs of $f_\xi : T_A \to T_B$. (c) MAEs of $f_K : T_A \to T_B$.

Figure 3.8: The robustness analysis of different methods.

The MSEs for all transfer learning methods decreased as the target data increased, indicating that more target data is beneficial for improving transfer learning performance. In the case of limited target data size, the MSEs learned with only the target data were much larger than those of the transfer learning method, implying that transfer learning is effective for data-scarce scenarios. For Fig. 3.9b, the source model had a significantly large MAE of more than 1.8e6 N/m, which means that a large distribution discrepancy exists between the source data and target data. TLB in Fig. 3.9b follows the MAE value of Source and decreases slightly with the increasing target data size. It is reasonable because TLB directly trains the model with weighted source data and target data, the model performance will turn toward the source model because $n_s \gg n_t$. In both Fig. 3.9a and Fig. 3.9b, CDA could achieve significantly low MAEs even with a limited number of target data.



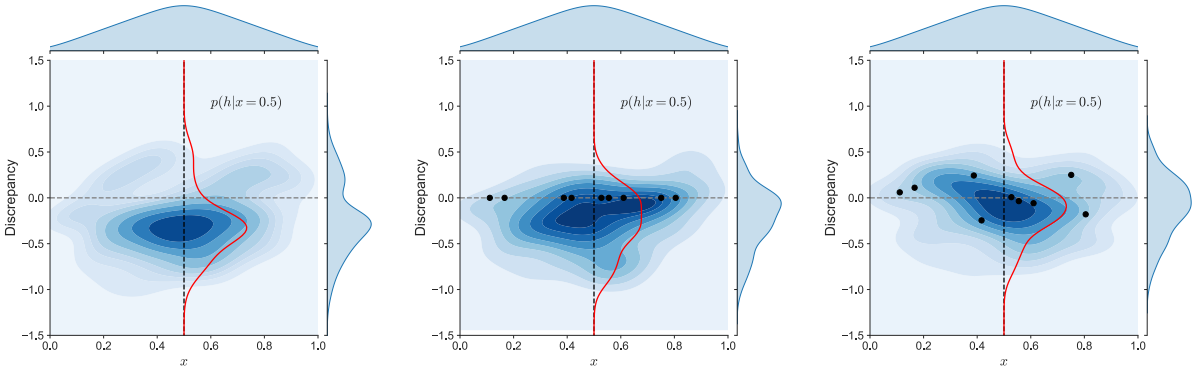
(a) Task $f_\omega : T_A \to T_B$. (b) Task $f_K : T_A \to T_B$.

Figure 3.9: MAEs with different size of target data.

3.5.1.4 Distribution discrepancy analysis

Although the above Sections have verified the performance of the transfer learning, it is still necessary to analyse the performance of the distribution discrepancy adaptation. This section will investigate whether the global distribution discrepancy was reduced

and what the difference was between minimising the embedding distribution discrepancy and minimising the loss of the training target data.



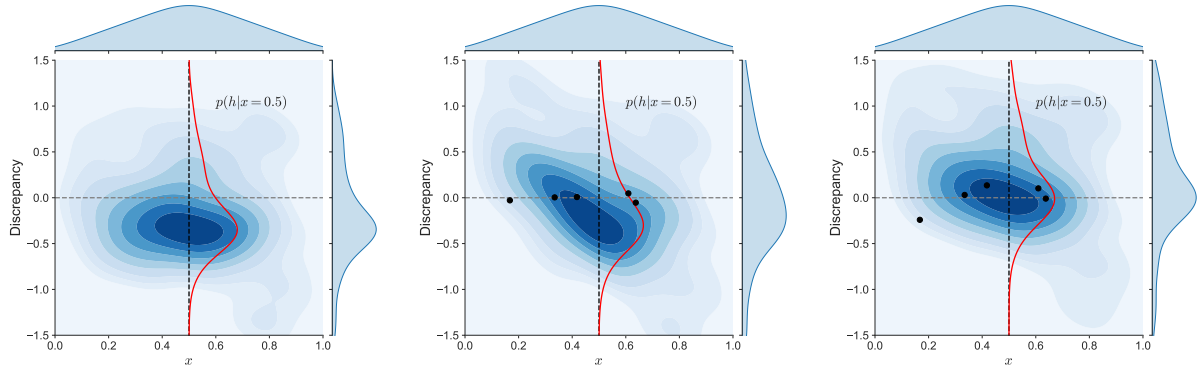
(a) Without adaptation. (b) Adaptation by TLB. (c) Adaptation by CDA.

Figure 3.10: Distribution discrepancy after adaptation for the task $f_\omega : T_B \to T_A$.

Figs. 3.10,3.11 ,3.12 show the comparison results of distribution discrepancy of three transfer tasks, $f_\omega : T_B \to T_A$, $f_\xi : T_B \to T_A$ and $f_K : T_B \to T_A$. Each figure consists of three joint distribution plots, including original discrepancy across tasks (Figs. 3.10a,3.11a ,3.12a), discrepancy after adaptation using TLB (Figs. 3.10b,3.11b ,3.12b), and discrepancy after adaptation using the proposed CDA (Figs. 3.10c,3.11c ,3.12c). The abscissa of these plots $\mathbf{x}_{pca}$ is the feature space reduced to 1-dimension using Principal Components Analysis (PCA). The vertical axis is the residual between the hypotheses of two tasks, i. e. $h(\mathbf{x}) = f_t(\mathbf{x}) - f_s(\mathbf{x})$. The colour maps are the joint probability densities $p(h(\mathbf{x}), \mathbf{x}_{pca})$ estimated using Kernel Density Estimation (KDE) based on the residual data to represent the distribution discrepancy. The darker areas in the colour maps mean greater probability. The upside subplots are the marginal probability density curve p (x) obtained by integrating along the Y direction of the joint distribution graphs. The subplots on the right side are the marginal probability density curve p (h) obtained by integrating along the X direction of the joint distribution graphs. And the blue curves are density curves of conditional distribution for specific $x = C$, for example $p(h|x = 0.5)$ as shown in Figs. 3.10(a)(b)(c). The black points in Figs. 3.10bc,3.11bc ,3.12bc are part of target training data.

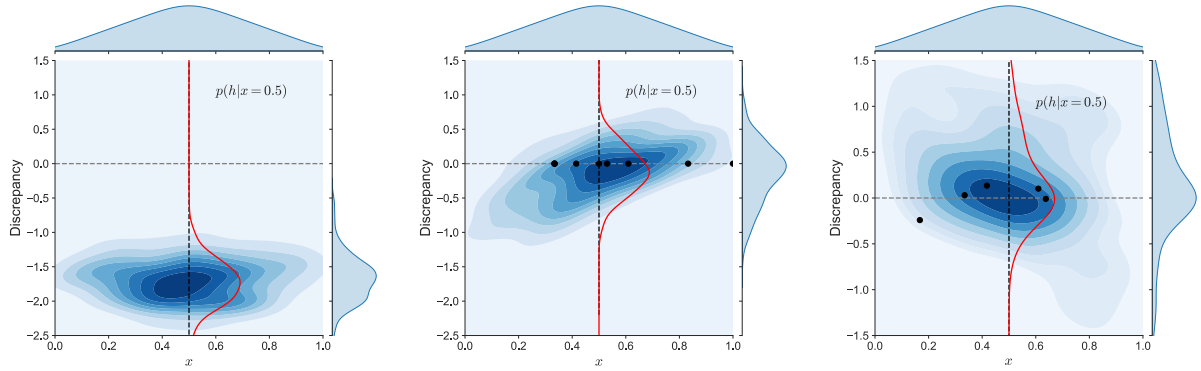
Figs. 3.10a, 3.11a, 3.12a show that there exists a clear conditional distribution discrepancy for all three tasks, where the red distributions $p(h \mid x = 0.5)$ are all far away from the middle $h = 0$. According to the joint distribution and the conditional distribution $p(h \mid x = 0.5)$ in Figs. 3.10, both TLB and CDA can reduce the distribution discrepancy, but the conditional distribution of CDA is closer to $h = 0$. The black data points in CDA are more distributed than in TLB, which means the prediction error on these points is bigger. That is because CDA can learn a more general representation of the global distribution rather than the prediction loss on the labelled target data.

As shown in Fig. 3.11b , almost all black points lie on the curve $h = 0$, but there is still an apparent shift for $p(h \mid x = 0.5)$. This means TLB cannot adapt the global conditional shift by only considering the loss of the target data. Meanwhile, the mean value of the blue curve in Fig. 3.11c is closer to $h = 0$ although there are no training



(a) Without adaptation. (b) Adaptation by TLB. (c) Adaptation by CDA.

Figure 3.11: Distribution discrepancy after adaptation for the task $f_\xi : T_B \to T_A$.



(a) Without adaptation. (b) Adaptation by TLB. (c) Adaptation by CDA.

Figure 3.12: Distribution discrepancy after adaptation for the task $f_K : T_B \to T_A$.

data in $x = 0.5$, which demonstrates that CDA can adapt the conditional distribution globally because of the structured discrepancy representation. The same conclusion can also be obtained in the density curve $p(h \mid x = 0.5)$ in Fig. 3.12b-c and 3.12b-c.

The above results indicated that the proposed CDA could achieve a significant improvement the accuracy, especially when with a small size of training data. Further analysis shows that the conditional distribution discrepancy could be reduced globally with the proposed structured representation.

3.5.2 Case study two: Multi-sensor measurement

3.5.2.1 Problem statement

With the development of modern manufacturing technology, complex surfaces have been increasingly used in aerospace, optics, moulds and other fields [202, 203]. Geometry measurement of these surfaces plays an essential role in different aspects of the manufacturing industry, including quality control, reverse engineering, and specifications verification [204]. Touch probes can provide high-precision measurement results,

but it is usually time-consuming and requires a specific environment that might limit their application scenarios [101]. By comparison, non-contact measurement sensors, such as structured light and laser scanners, can acquire high-resolution point clouds of the target object fastly, but the relatively low precision cannot meet the practical requirements. Considering the complementary measurement quality and speed of different sensors, multi-sensor data fusion has become a significant development trend for complex surface measurement [205, 201]. The general definition of multi-sensor data fusion refers to combining data from several sensors into a common representational format so that the metrological evaluation can benefit from all available sensor information and data [205].

This case study aims to fuse the measurement data from the laser scanner and the touch probe so that the metrological evaluation can benefit from all available sensor information and data [101]. Since the probe is much more accurate than the laser scanner, the insufficient probe data is considered the true reference of the surface, and the laser scanner data can be treated as the auxiliary data. Therefore, the multi-sensor fusion problem here can be defined as a transfer learning setting, where the knowledge of the laser scanner data, including the overall shape of the target surface, is transferred to the domain of the touch probe. As shown in Fig. 3.13, the two measurement datasets are defined as source and target data, respectively.

- **Source data** $\mathcal{D}_s$: Laser scanner dataset with high density, low metrological performance, consists of around 30,000 points measured by the Kreon Aquilon laser scanner shown in Fig. 3.13b.
- **Target data** $\mathcal{D}_t$: Touch probe dataset with low density but high metrological performance, consists of $n_t = 96$ points measured by the touch probe sensor mounted on a Kreon Ace measuring arm, as shown in Fig. 3.13c.

It is assumed that the difference between the laser scanner data and the touch probe data consists of systematic and random errors. The distribution adaptation in this transfer learning task refers to estimating and compensating the error function, namely the residual, between the two measurement datasets. This experiment focused on the nearly cylindrical surface that has sharply varying gradients. As the arrows shown in Fig. 3.13d, some points on the surface have normal vectors almost perpendicular to the z-axis, which brings a significant challenge for the residual modelling.

3.5.2.2 Experimental settings

In this case, the conditional distribution discrepancy between two measurement datasets is represented based on the Gaussian mixture model introduced in Section 3.3.3. The two best-known methods in multi-sensor fusion are weighting fusion and residual approximation fusion [206, 207], corresponding to transfer learning method RA and TLB mentioned in the previous sections. Therefore, these two methods were implemented to compare the transfer learning results.

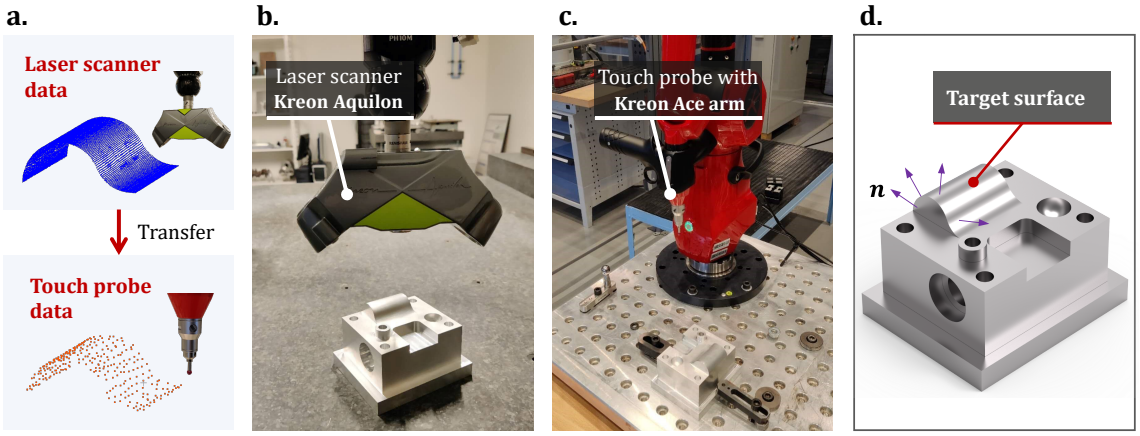


Figure 3.13: The problem statement of the multi-sensor measurement case. (a) The transfer learning settings. (b) The laser scanner Kreon Aquilon. (c) The touch probe with Kreon Ace arm. (d) The target surface for the multi-sensor measurement.

First, assume that all laser scanner points $\mathcal{D}_s$ are generated by a K Gaussian distribution, where the point clouds belonging by each distribution constitute a point cluster. Therefore, the residual function $h(\mathbf{w})$ can be represented by the Gaussian mixing of all the component residual w_k defined on each point cluster.

$$h(\mathbf{x}) = \sum_{k=1}^{K} w_k \gamma\left(z_k\right) * \mathbf{n}_k \tag{3.41}$$

Note that the above equation is almost the same with Eq. 3.12, the difference is the $\mathbf{n}_k$ in the end. The component residual defined in each point cluster has a primary normal vector $\mathbf{n}_k$. Therefore, the final residual function should be the Gaussian mixing of all component residual w_k on the corresponding normal vector.

Considering the density of the touch probe points $\mathcal{D}_t$ is much smaller than that of the laser scanner point $\mathcal{D}_s$, which means each touch probe point cloud corresponds to a subset of the laser scanner point cloud. Therefore, the number of the Gaussian distribution can be set as the size of touch probe data, namely $K = n_t = 96$. Each probe point can intuitively be defined as the center of the corresponding Gaussian distribution, i.e. $\boldsymbol{\mu}_k$ defined in Eq. 3.5.

Fig. 3.14 shows the point clusters of different Gaussian distributions for the experimental surface, where the points belonging to each cluster are marked with the same colour. After the clustering of $\mathcal{D}_s$, each point cluster can approximate the local region of the target surface linearly, which means the component residual w_k can also be evaluated conveniently.

3.5.2.3 Experimental results

Considering the high accuracy of the touch probe, this experiment used the 190 probe points as the reference to evaluate the error of the fused point cloud. Fig. 3.15 shows the

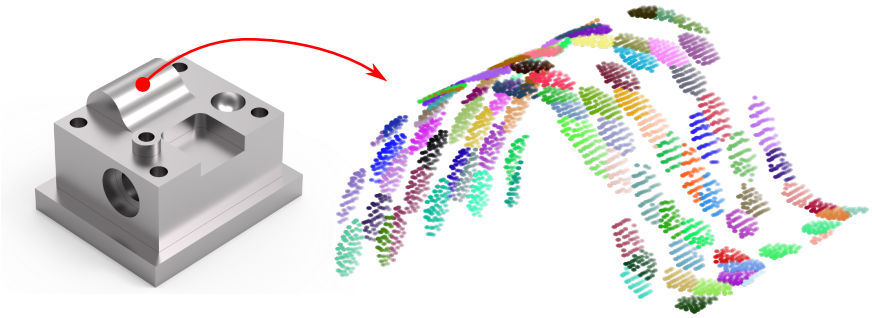


Figure 3.14: The point clusters of different Gaussian distributions.

error maps of the transfer learning results. The 3D plot shows the errors of all 190 test points, where the deep blue and deep red points mean large errors and green points have relatively small errors. The statistical results are also provided below the 3D figures. For Fig. 3.15a, the result of TLB, the laser scanner data and touch probe data are combined together with different weights to learn the surface model. Since the number of laser points is much larger than the number of probe points, this result in Fig. 3.15a is almost identical to fitting the laser points directly. The left side of the surface with greater slopes has significantly larger errors of over 0.1mm, while the smooth area at the top of the surface has a relatively smaller error. Fig. 3.15b is the error map of RA transfer learning method. There still exist significant errors in the left region despite the overall error reduction. Fig. 3.15c shows the error map after fusion by the proposed CDA method. The most region of the surface is green, which means that the error is greatly reduced compared to Fig. 3.15a-b.

The error statistics for the above three comparisons are shown in Tab. 3.2, where σ is the standard deviation of the Gaussian distribution fitted from the error distributions in Fig. 3.15. E_{avg} and E_{max} denote the mean and maximum value of the absolute errors at the test points. RA can reduce the overall error slightly with σ from 0.045mm to 0.042mm. In contrast, CDA can reduce E_{avg} from 0.039mm to 0.017mm and E_{max} from 0.124mm to 0.078mm, significantly outperforming the RA method.

Table 3.2: The comparison of error statistics on the measurement case.

Method	σ (mm)	E_{avg}(mm)	E_{max} (mm)
TLB	0.045	0.039	0.124
RA	0.042	0.032	0.138
CDA	0.022	0.017	0.078

3.5.2.4 Distribution discrepancy analysis

As analysed in Section 3.1, existing transfer learning method, such as RA, can only adapt discretised distribution discrepancy near the labelled data. At the same time, the pro-

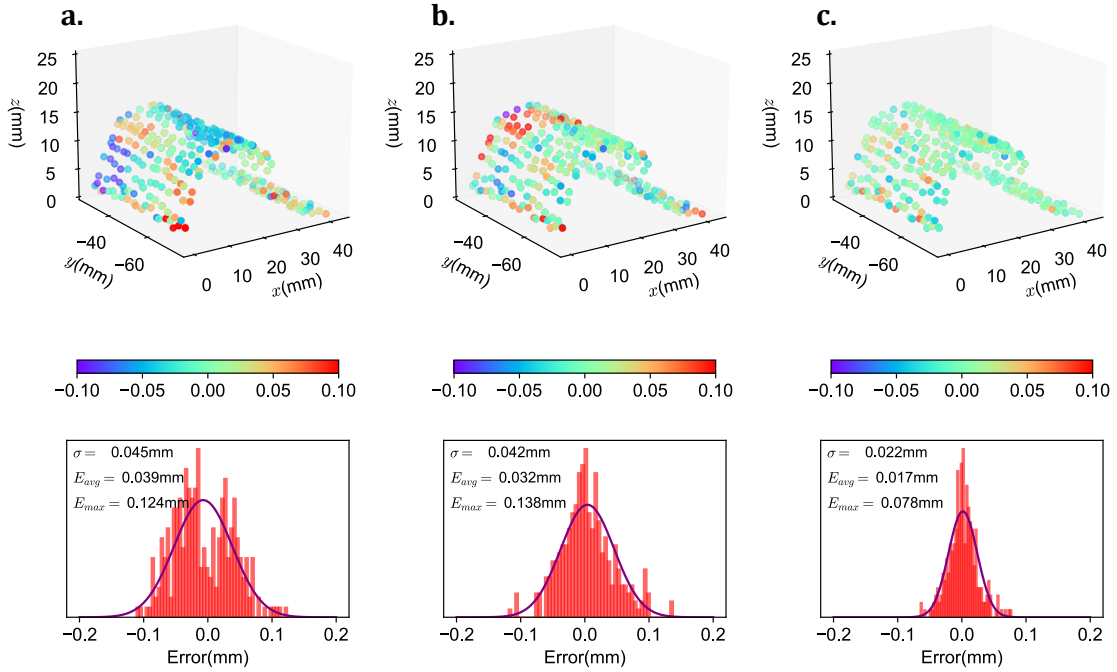


Figure 3.15: Comparison of the transfer learning results for the multi-sensor measurement case. (a) The error map of TLB. (b) The error map of RA. (c) The error map of CDA.

posed CDA can represent the distribution discrepancy structurally based on GMM. Fig. 3.16 compares the residual modelling of RA and CDA to explain the difference between discretised and structured representations. The z direction residual of all training touch probe points is shown in Fig. 3.16a, where the complex and sharply varying distribution significantly challenges RA. By comparison, for the proposed CDA method, the distribution discrepancy of the two data sets can be represented by the residual defined on each cluster, as shown in Fig. 3.16b. The simplicity of the sub-residuals means that the CDA model can provide an accurate and efficient representation of the residuals between different measurement data. Because of the simple structured representation, the distribution adaptation can be easily carried out by solving the sub-residuals with a least square method.

3.5.3 Case study three: Tool wear prediction

3.5.3.1 Problem statement

This case study validated the proposed CEOD loss function in deep transfer learning settings. The conditional distribution adaptation was realised by fine-tuning the network parameters with the CEOD-based loss function defined in Eq. 3.39.

The healthy condition of cutting tools directly influences the machining process stability, and the final quality of the product [78]. Due to the complexity of the cutting process, it is difficult to predict the tool wear accurately using mechanism models. Since

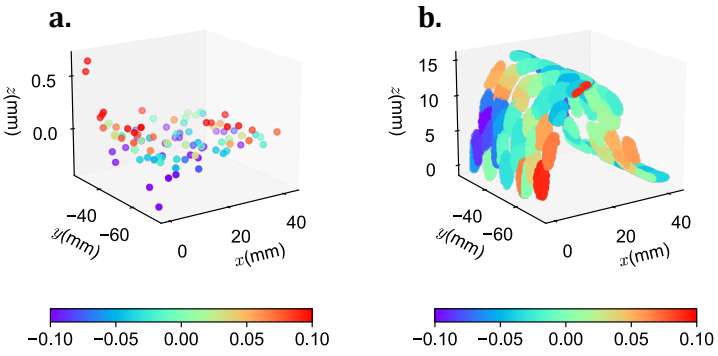


Figure 3.16: Comparison of the distribution discrepancy representation. (a) The discretised distribution discrepancy represented by RA. (b) The structured distribution discrepancy represented by CDA.

the tool wear will result in inconsistent cutting width, further leading to the fluctuation in the cutting force and other signals, building data-driven models from the monitoring cutting force to the corresponding tool wear becomes a potential solution. Fig. 3.17a illustrates the input feature and output label of the tool wear prediction problem.

As reviewed in Section 1.2.2.1, collecting tool wear data requires expensive instruments and time-consuming operations. It is almost impossible to collect sufficient labelled tool wear data for the target tool [80]. Therefore, transferring knowledge from existing source tool can potentially reduce the labelled data requirements for the target tool. The tool wear dataset from the Prognostics and Health Management Society conference data challenge [171] consists of seven types of monitoring signals and tool wear values for three blades of two cutting tools (C4 and C6). This case study selects x-axis milling signal as the input feature, and the output label is the measured tool wear value (um). Chapter 2 also includes the tool wear prediction problem, but the input data were pre-extracted high-level features to simplify the regression problem. In this case, the input cutting force is the sequence signal window with 20000 discrete values, namely with the shape of 20000×1. Four transfer learning tasks are constructed: C4 $\rightarrow$ C6-Blade2, C4 $\rightarrow$ C6-Blade3, C6 $\rightarrow$ C4-Blade2, C6 $\rightarrow$ C4-Blade3.

3.5.3.2 Experimental settings

The deep learning structures of the source and tasks are the same, as shown in Fig. 3.17b. Four 1-D convolution modules are introduced to extract features from the input monitoring signal series. Each convolution module consists of a convolution layer, a batch normalisation layer and a max-pooling layer. For each task, the source data size is $n_s = 315$, and the target data size is 30. The transfer learning procedure consists of two steps, pre-training the source model and fine-tuning part of the model parameters to adapt the distribution discrepancy.

Step 1: Pre-training the source model. The structure of the source model is shown in Fig. 3.17b left. The detailed configurations of the neural network are listed in Tab. 3.3.

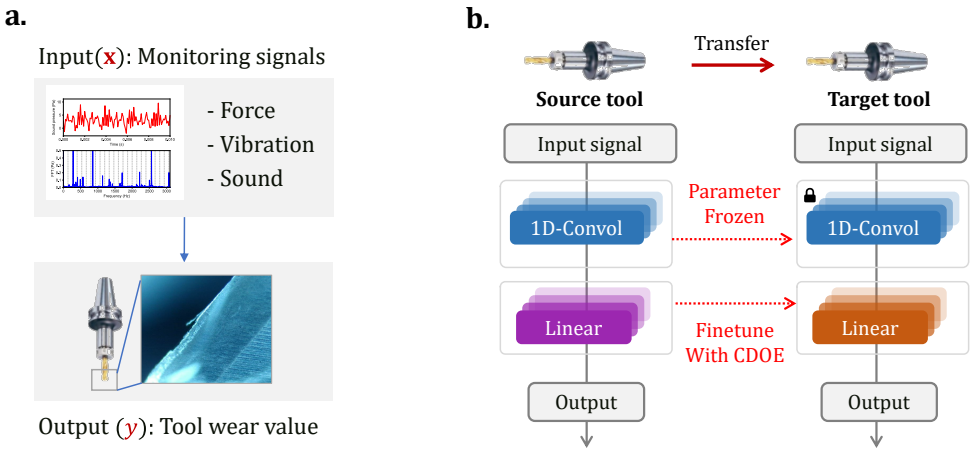


Figure 3.17: The problem statement of the tool wear prediction case. (a) The input feature and output label of the tool wear prediction problem. (b) The deep learning model and the transfer learning settings.

The loss function for training is the L2 loss on the source data:

$$\mathcal{L}^s_{MSE} = \frac{1}{n_s}\sum_{i=1}^{n_s}\left(f\left(\mathbf{x}^s_i\right) - y^s_i\right)^2 \tag{3.42}$$

Step 2: Distribution Adaptation based: After the pre-training the model of the source tool, the parameters of the feature extraction layers were frozen and reused for the target tool. As shown in Fig. 3.17b right, three linear layers that map the features to the output tool wear values were fine-tuned to adapt the distribution discrepancy between the two datasets. The parameters of the three linear layers $\boldsymbol{\theta}_t$ were updated based on the following loss function defined in Section 3.4, where the hyperparameters were as follows: $\beta = 1e-2$, $\lambda_r = 1e-5$.

$$\mathcal{L}(\boldsymbol{\theta}_t) = \beta\,\mathrm{CEOD}(p\left(\mathbf{y}^{\text{new}}_s \mid \mathbf{X}_s\right), p\left(\mathbf{y}_t \mid \mathbf{X}_t\right)) + (1-\beta)\mathcal{L}^t_{MSE} + \lambda_r \boldsymbol{\theta}_t^\top \boldsymbol{\theta}_t \tag{3.43}$$

Table 3.3: The configurations of the neural network

Item	Configuration
Convolution modules	Cov(1, 64, 5, stride=5) + BN + Relu
	Cov(64, 64, 5, stride=5) + BN + Relu
	Cov(64, 64, 5, stride=5) + BN + Relu
	Cov(64, 64, 3, stride=5) + BN + Relu + flatten
Linear layer	$384 \rightarrow 64 \rightarrow 16 \rightarrow 1$
Learning rate	1e-3
Epoch	100

To evaluate the performance of the proposed method in deep learning settings, the state-of-art deep transfer learning method DAN (Deep Adaptation Networks) [181] was implemented for comparison. DAN aims to learn transferable features by minimising the MMD of two tasks in a reproducing kernel Hilbert space. It is a classical method for marginal distribution adaptation of classification problems.

3.5.3.3 Experimental results and analysis

Tab. 3.4 presents the MAE and corresponding STD of all transfer learning tasks, where all results are the mean values of 20 repeated trials with different random seeds. The proposed CDA significantly reduced the MAE, and outperformed all comparison methods in 3/4 of tasks. Although TLB shows superior performance in task C6 $\rightarrow$ C4-Blade2, CDA still achieved satisfactory stable results compared with other methods. In summary, transferring the knowledge from the source tool improved the prediction of the target tool, which then reduced the label acquisition cost for the target task.

Table 3.4: Comparison results of different transfer learning methods for tool wear case.

Data	C4 $\rightarrow$ C6-Blade2	C4 $\rightarrow$ C6-Blade3	C6 $\rightarrow$ C4-Blade2	C6 $\rightarrow$ C4-Blade3
Target	36.50 ± 0.61	43.52 ± 0.67	39.22 ± 14.13	45.04 ± 0.27
RA	63.79 ± 3.21	65.16 ± 2.28	36.768 ± 13.87	26.29 ± 9.43
TLB	15.59 ± 2.37	16.28 ± 4.05	15.35 ± 2.14	**15.69 ± 2.74**
DAN	34.85 ± 3.82	43.62 ± 2.34	47.17 ± 3.91	45.48 ± 4.98
CDA	**13.03 ± 1.01**	**15.37 ± 1.20**	**15.10 ± 1.05**	17.60 ± 1.20

(um) The best results are marked with bold.

3.6 Summary

This Chapter focused on transferring knowledge from auxiliary data to compensate for the insufficient modelling information and reduce the requirements of labelled data. Since existing transfer learning methods only adapt discretised distribution discrepancy of limited target data, this research proposed a structured conditional distribution adaptation, in which the discretised discrepancy on the original feature space could be transferred to structured discrepancy defined in the k-dimensional latent space, including Gaussian Mixture Model and fuzzy rules. After that, a conditional distribution discrepancy adaptation method was proposed based on the defined conditional embedding operator discrepancy.

The proposed transfer learning method was validated in several manufacturing problems, including tool top dynamics prediction, multi-sensor measurement, and tool wear predictions. The results indicated that the proposed method could achieve a significant improvement in both accuracy and precision. Further analysis showed that the latent variables could learn a more general residual representation. Thus the conditional distribution discrepancy can be reduced globally. The experimental results in this Chapter demonstrated that leveraging the available auxiliary data could enhance the performance of the task under data scarcity.

Transfer learning can effectively utilize auxiliary data to compensate for the limited modelling information. Similarly, the widely-exists physics priors of manufacturing problems, although not in the data form, also have the potential to enhance the performance of the data-driven model. Chapter 5 will explore the feasibility of combining the

data-driven method with physics knowledge to address the challenging task of predicting high-dimensional part property fields.

Chapter 4

Data physics combination: Physics-guided low-dimensional neural operator

Great acts are made up of small deeds.

– Laozi

The quality properties or manufacturing process properties of workpieces or products are typically represented by high-dimensional discretising points of their part geometry. How to build a data-driven prediction model of high-dimensional part properties with a limited number of labelled data remains a challenge for data-driven intelligent manufacturing. This Chapter investigates the **objective** 3 of the framework introduced in Fig. 1.30, data-physics combination, i.e., integrates the physics priors to enhance the learning performance of the data-driven models. To address the complex high-dimensional field mapping MPMs problems, this Chapter proposes a physics-guided low-dimensional neural operator, which integrates physics-guided basis functions into the data-driven model to reduce the complexity and improve the ability to extract explainable features. The proposed model can predict the high-dimensional property field of the workpiece accurately with only a small number of labelled data.

4.1 Introduction and Challenge analysis

Modelling the properties of workpieces that can reflect the manufacturing state is essential for quality control and process optimisation. For example, the residual stress fields during the milling of metal parts significantly affect the final machining deformation and fatigue performance [90]. For composite parts curing, temperature, stress, and degree of cure fields during solidification directly determine the mechanical performance and service life of the parts [149].

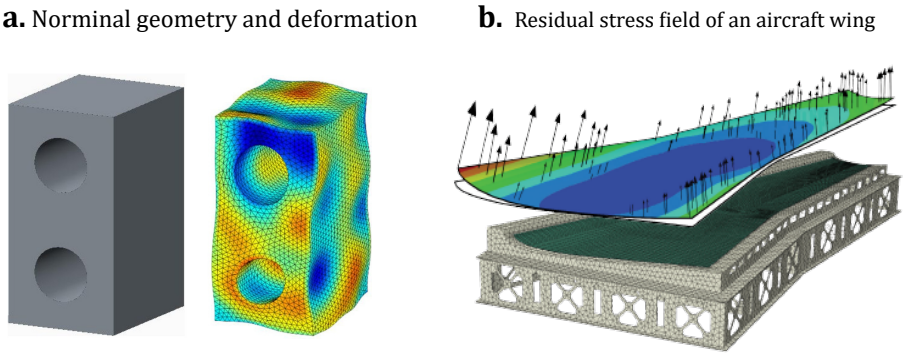


Figure 4.1: Examples of part properties that are represented by high-dimensional discretised mesh points. (a) The nominal geometry and deformation [208]. (b) The residual stress field of an aircraft wing.

Part properties, such as the residual stress fields of metal parts or curing deformation of composites parts, whether the intermediate process properties or the final quality properties, are typically represented on a large size of discretising points of the part geometry [88, 209]. Fig. 4.1a-b shows part property examples of deformation and stress field, which are both defined on a large size of discretised mesh points. Therefore, the output label of the data-driven part property prediction model is not a simple scalar, but a high-dimensional part property field, which means that the data-driven predictive modelling would require complex parametric models and extensive labelled data [90, 210]. However, obtaining high-dimensional labelled data part property field is usually very expensive, whether through simulation or experimental measurement [85]. Therefore, building a predictive model of high-dimensional part properties with a limited number

of labelled data remains a challenge for data-driven intelligent manufacturing.

Predicting high-dimensional property fields of parts requires **geometry representation** and **feature extraction from high-dimensional data**. Convolutional neural networks and neural operators are currently widely used models for predicting part property fields.

Deep convolutional neural networks (CNNs) and deconvolution techniques have shown promise in learning mappings between high-dimensional images, and thus can be used to approximate high-dimensional part property fields discretely [211]. For example, U-Net, a special CNN for image segmentation in biomedical tasks [212], has been applied in surface textured defects modelling [213], strain field of composite parts [214] and other high-dimensional property field predictions. However, due to the limitations in the image convolution operations, CNNs require image-like regular grid as input, which restricts their applicability for 2D or 3D complex geometries that widely exist in engineering fields. Another limitation comes from the inductive bias of image feature extraction. CNN involves sliding fixed-size kernels over the images to extract local features, which means the basic assumption that the advanced features of figures only depend on the neighbouring pixels [215]. However, many engineering problems involve global features rather than local features, such as mechanical vibration mode [216], fluid mode decomposition [217] and structural parts processing deformation [218]. To sum up, traditional CNN-based models remain inapplicable in function mapping learning problems of complex geometries.

Recently, neural operators [219] (or operator networks [220]), that could directly learn the mappings between function spaces, have shown promising results in high-dimensional property field prediction. Neural operators aim to build generalised models that are independent of the domain discretisation, which means that, a model with a limited number of parameters can be generalised to the geometric domain with high discretisation resolution[219]. Lu et al. [221] proposed the first neural operator framework, DeepOnet, based on the universal approximation theorem. Li et al. [222] formulated the approximation of the infinite-dimensional function mapping with multiple integral operators, and proposed the Graph Neural Operator (GNO) by the massage passing mechanism of graph neural networks. The parameters of GNO are defined on the graph kernel that is independent of the domain discretisation, which means that the model complexity will not increase with the high-dimensional discretisation of the part geometry, enabling a resolution-independent neural operator. However, GNO suffers from unstable training, and the $O(K^2)$ complexity (where K is the number of edges in the graph) [223]. Based on this neural operator structure, the same authors then proposed Fourier Neural Operator(FNO) [211], which parameterised the integral operators in the Fourier domain. The high-dimensional mapping is then transferred to the low-dimensional discretisation-invariant parameterisation of few frequency modes. Despite the significant success of FNO, the fast Fourier transform of FNO requires the input and output functions to be defined on a regular domain with a lattice grid mesh, which means that FNO is inherently inapplicable to complex geometry, including irregular domain boundaries or irregular meshing [220].

As discussed previously, both CNN and FNO are only suitable for regular domains.

For complex 2D geometries, irregular grids need to be transformed into Cartesian grids using techniques such as elliptic coordinate transformation [224], grid resampling [225], or grid interpolation [220]. However, these techniques are still limited to simple 2D irregular domains due to their poor intrinsic geometrical representation. Since the structure and parameter complexity of CNN and FNO depend on the data grid, the interpolation of 3D complex geometries will significantly increase the burden of model training and the requirement for labelled data [226].

In summary, the challenge for modelling part property fields lies in how to **represent high-dimensional complex geometry of workpiece** and **extract low-dimensional features** without increasing the complexity of the data-driven model.

4.2 General idea of physics-guided low-dimensional neural operator

To predict the high-dimensional part property fields, existing deep learning methods have to interpolate complex geometry into a high-resolution regular domain (Fig. 4.2a), which significantly increases the complexity of the model and the number of labelled data required. Just like the feature extraction in CNN that is designed based on the physics prior of the image recognition problem, predicting high-dimensional property fields of complex parts can also leverage the physics priors when designing the model structure. This Chapter describes a proposed physics-guided low-dimensional neural operator. As shown in Fig. 4.2b, A series of **physics-guided basis** is constructed and embedded in the neural network layers to extract the low-dimensional information from the part property fields. The **high-dimensional mapping in the spatial geometric domain** is then transferred to **the low-dimensional mapping in the basis domain**, thus significantly reducing the complexity of the model and the requirements for labelled data.

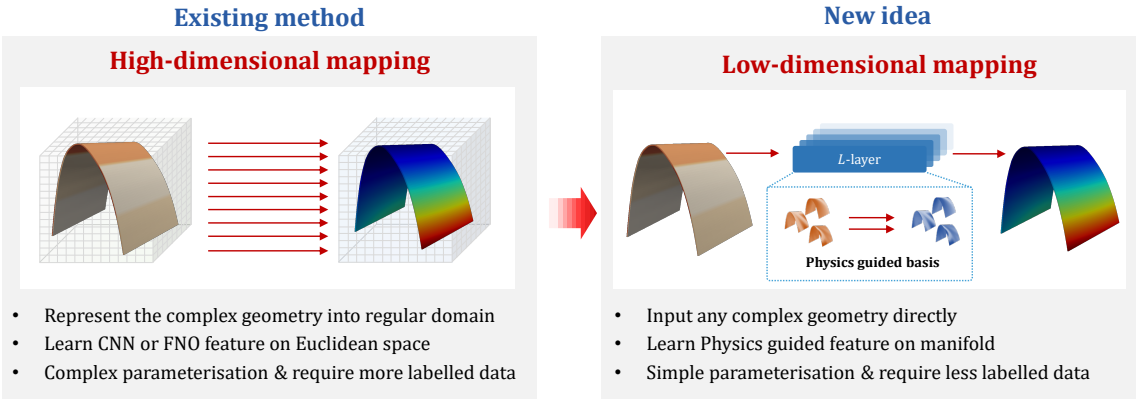


Figure 4.2: The general idea of physics-guided low-dimensional neural operator.

Based on the above ideas, the detailed physics-guided low-dimensional neural operator (LNO) is shown in Fig. 4.3. The basic architecture of the LNO consists of multi-layer iterative L-layers and two traditional linear layers P and Q at the beginning and the

end. Physics-guided basis functions, including **geometry-related basis** and **attribute-related basis**, are constructed before training the model. Each L-layer forms an interpretable low-dimensional representation by encoding the input property field of the complex geometry into the physics-guided basis functions, which could significantly reduce the model complexity by parameterising it in the basis space.

Section 4.3 will first provide the mathematical problem definition and introduce the basic framework of the neural operator. Section 4.4 will construct the physics-guided basis functions and then define the iterative L-layer.

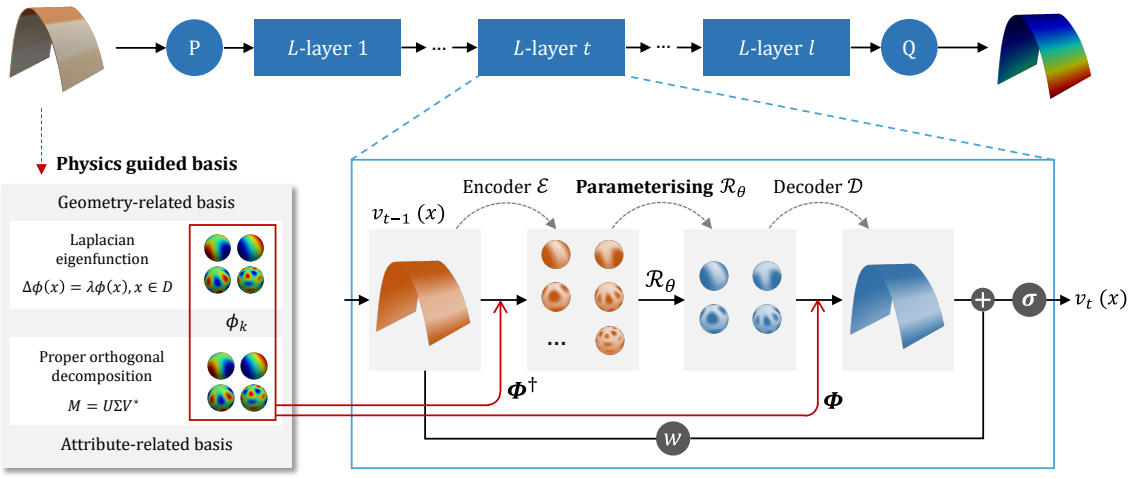


Figure 4.3: The framework of LNO

4.3 Problem definition and background

4.3.1 Problem definition

Traditional neural networks are commonly used for predicting scalars or low-dimensional vectors, such as in classification or regression tasks. In contrast, predicting high-dimensional part property fields of parts can be framed as predicting a function over a geometry domain. The mapping between function spaces is known as an operator. Neural operators are a novel concept in the field of machine learning, aimed at learning mappings between input and output functions based on training data [219].

This section first provides the problem definition of predicting part property fields from the operator perspective, then defines the learning objective. Suppose that the target property functions are defined on a bounded domain D on the d-dimensional Euclidean space $\mathbb{R}^d$. Considering that the computational domain in engineering and simulations are often complex geometries, more generally, D is assumed to be a manifold embedded in a d-dimensional Euclidean space, such as surfaces (2d manifolds in $\mathbb{R}^3$) or solids (3d manifolds in $\mathbb{R}^3$) [227]. For numerical implementation, the computation domain is usually discretised into L points, namely $\{p_\ell\}_{\ell=1}^{L} \subset D$. The boundary definition

and discretisation of the domain will influence the complexity of the numerical computation. The regular geometric domains can be represented by matrixes conveniently, while the complex geometric domains are typically represented by mesh defined with vertices and polygons. In this research, the regular domain refers to the rectangle domain discretised into lattice grid mesh, as shown in Fig. 4.4a. The complex geometric domain consists of both irregular boundaries and irregular mesh discretisation, as shown in Fig. 4.4b-d.

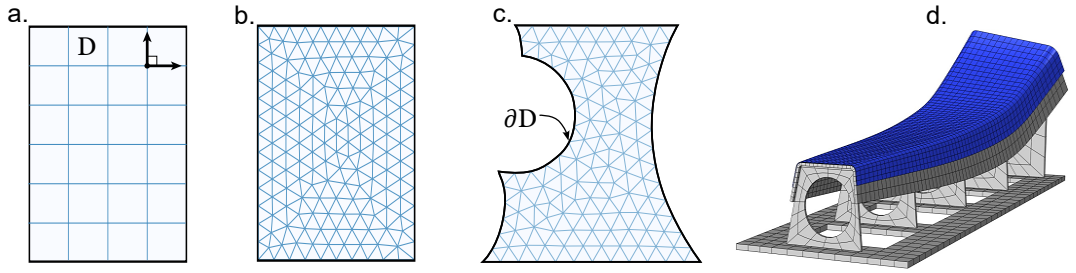


Figure 4.4: The examples of the regular domain and irregular domain. (a) Rectangle domain with regular mesh (b) Rectangle domain with irregular mesh (c) 2d irregular geometric domain (d) 3d irregular geometric domain.

Suppose that the input function and the output function are both defined on domain D, and take values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$, namely $a(x) : D \to \mathbb{R}^{d_a}$, $u(x) : D \to \mathbb{R}^{d_u}$. For example, considering the mapping from the temperature field to the deformation field in the composites curing. The input temperature field is denoted as $a(x)$, and the output deformation field is denoted as $u(x)$. Note that x only refers to one sample in the domain of definition, and its actual meaning depends on the specific problem. It can represent the spatial coordinates of a grid node or the index of nodes. $\mathcal{A}$ and $\mathcal{U}$ are defined as the Banach spaces for the input and output functions, so that $a \in \mathcal{A}(D; \mathbb{R}^{d_a})$ and $u \in \mathcal{U}(D; \mathbb{R}^{d_u})$. For most engineering problems, both $a(x)$ and $u(x)$ are scalar functions with specific physical meaning, that is, $d_u = d_a = 1$, such as the deformation field or temperature field. But from a more general perspective, $a(x)$ and $u(x)$ can also be defined as attribute vectors on spatial node x, thus dimension d_u and d_a can be greater than one, such as stress field and other vector fields.

To construct the machine learning loss function, it is necessary to introduce the norm of functions $\|\cdot\|$. It is assumed that both $\mathcal{A}$ and $\mathcal{U}$ are Banach spaces, i.e., complete normed spaces. Furthermore, $\mathcal{G}$ is defined as the underlying map between the input and output functions, so that, $\mathcal{G} : \mathcal{A}(D; \mathbb{R}^{d_a}) \to \mathcal{U}(D; \mathbb{R}^{d_u})$. The neural operator aims to approximate $\mathcal{G}$ by constructing a parametric operator $\mathcal{G}_\theta$ using a neural network parameterised by $\theta \in \mathbb{R}^p$, i.e. $\mathcal{G}_\theta \approx \mathcal{G}$. The learning target can then be defined as an empirical-risk minimisation problem:

$$\min_{\theta \in \mathbb{R}^p} \mathbb{E} \left\| \mathcal{G} - \mathcal{G}_\theta \right\|_{\mathcal{U}} \tag{4.1}$$

where $\|\cdot\|_{\mathcal{U}}$ denotes the norm defined in the Banach space $\mathcal{U}$, typically using the L^2 norm. Therefore, both spaces $\mathcal{A}$ and $\mathcal{U}$ are assumed to be L^2 spaces, which are function spaces

consisting of functions that are square integrable in domain D. Like classical supervised learning, the neural operator trains model parameters θ using the data set consisting of input functions a and output functions u. Supposing that the training data consists of N samples, $\{a^{(i)}, u^{(i)}\}_{i=1}^{N}$, that is, $u^{(i)} = \mathcal{G}(a^{(i)})$, then Eq. 4.1 can be expressed as

$$\min_{\theta \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \left\| u^{(i)} - \mathcal{G}_\theta \left(a^{(i)} \right) \right\|_{L^2} \tag{4.2}$$

4.3.2 Neural network structure

This research took the kernel integral operator scheme, a family of infinite-dimensional operators proposed by Kovachki et al. [219]. As shown in Eq. 4.3, the neural operator consists of two feature mapping layers P, Q and l kernel integral operator layers. The shallow network P maps input function $a(x)$ to get $v_0(x) = P(a(x))$, where $P : \mathbb{R}^{d_a} \to \mathbb{R}^{d_v}, d_v \geq d_u$, so as to expand the dimension of node features to increase the representation ability, similar to the convolution channel expansion in CNN. Multiple kernel integral operators will update the input function iteratively as $v_0(x) \mapsto v_1(x) \mapsto \ldots \mapsto v_l(x)$. After that, the final shallow network Q will project the node features to the output dimension, namely $u(x) = Q\left(v_l(x)\right)$, where $Q : \mathbb{R}^{d_v} \to \mathbb{R}^{d_u}$. The iterative structure can be represented as:

$$\left(\mathcal{G}_\theta(a)\right)(x) = Q \circ v_l \circ v_{l-1} \circ \cdots \circ v_1 \circ P(a)(x) \tag{4.3}$$

where the iteration from v_t to v_{t+1} can be represented as:

$$v_{t+1}(x) := \sigma\left(W v_t(x) + \left(\mathcal{K}_\theta(v_t)\right)(x)\right), \quad \forall x \in D \tag{4.4}$$

where W and σ are linear transformations and non-linear activation function as in the traditional neural network. $\mathcal{K}_\theta(v_t) : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ parameterised by θ is the kernel integral operator, which plays the key component of the neural operator that maintains the discretisation-invariant property field. It is defined as:

$$\mathcal{K}_\theta(v)(x) = \int_D \kappa_\theta(x, y) v(y) \mathrm{d}y \quad \forall x \in D \tag{4.5}$$

where κ_θ refers to the parameterised kernel function. Different kernel integral operator definitions will lead to different instantiations of the neural operator [219].

The above derivations present the theoretical form of the neural operator, where both input and output are represented as functions. In practice, the input field function $a(x)$ and the output field function $u(x)$ are discretely represented as high-dimensional property matrices consisting of n_x nodes, that is, $\mathbf{a} \in \mathbb{R}^{n_x \times d_a}$ and $\mathbf{u} \in \mathbb{R}^{n_x \times d_u}$. The discrete form of the network architecture is shown in Fig. 4.5. The shallow network P increases the property dimension d_a to $P(\mathbf{a}) \in \mathbb{R}^{n_x \times d_v}$. Since the number of geometric discrete nodes n_x is usually very large, direct parameterisation in the n_x dimension would lead to overly complex network parameters. Therefore, the iterative kernel integration module of the neural operator will simplify the parameterisation form, and the output field

$\mathbf{v}_l$ after the iterative operation remains the same size as $\mathbb{R}^{n_x \times d_v}$. Finally, the shallow network Q transforms the node features to the dimension of the original output, that is, $P(\mathbf{v}_l) \in \mathbb{R}^{n_x \times d_u}$.

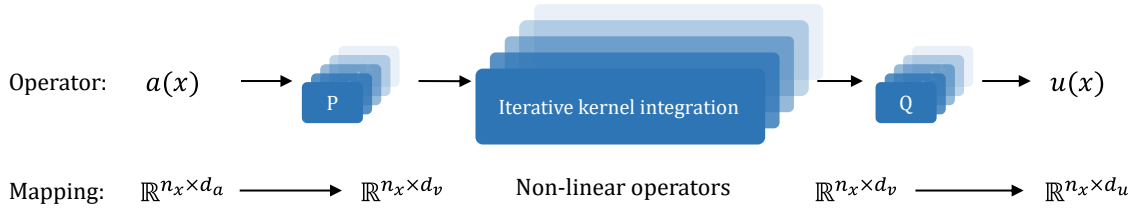


Figure 4.5: The basic structure of neural operator.

4.3.3 Fourier iterative kernel integration operator

The key procedure for the neural operator is to construct the iterative kernel integration module $\mathcal{K}_\theta(v)$, which will be parameterised in a low-dimensional pattern space rather than the original high-dimensional input space. Existing FNO can only handle function mapping problems on regular domains, which is incapable of addressing the property field prediction of complex parts. This section will first analyse the limitations of Fourier iterative kernel integration operators, and introduce the inspiration of low-dimensional iterative kernel integration operators for complex geometric domains.

By introducing the convolution theorem, FNO parameterises the kernel integral operator directly in the Fourier domain. Denoting $\kappa_\theta(x, y) = \kappa_\theta(x - y)$, then the kernel integral operator can be transferred into the convolution operation form:

$$\mathcal{K}_\theta(v)(x) = \int_D \kappa_\theta(x - y)v(y)\mathrm{d}y = (\kappa_\theta * v)(x) \quad \forall x \in D \tag{4.6}$$

where $\kappa_\theta * v$ is the convolution operation. By the convolution theorem, namely $\mathcal{F}\{\kappa * v\} = \mathcal{F}\{\kappa_\theta\} \cdot \mathcal{F}\{v\}$, then:

$$\mathcal{K}_\theta(v_t)(x) = \mathcal{F}^{-1}\{\mathcal{F}\{\kappa_\theta\} \cdot \mathcal{F}\{v_t\}\}(x) \quad \forall x \in D \tag{4.7}$$

where $\mathcal{F}\{\cdot\}$ is the Fourier transform and $\mathcal{F}^{-1}\{\cdot\}$ refers to the inverse Fourier transform. FNO is proposed to directly parametrise κ_θ in the Fourier domain, that is $\mathcal{R}_\theta = \mathcal{F}\{\kappa_\theta\}$. Therefore, the parameterisation of matrix $\mathcal{R}_\theta$ is only related to the number of frequency modes involved, but independent of the resolution of the original computational domain. The k-th frequency mode of input $v_t(x)$ is the complex-valued function $\mathcal{F}\{v_t\}(k) \in \mathbb{C}^{d_v}$, which can then be passed to the complex-valued parameterised matrix $\mathcal{R}_\theta(k) \in \mathbb{C}^{d_v \times d_v}$, followed by the inverse Fourier transform back to the original computational. FNO picks a finite-dimensional parameterisation by truncating the top k_l frequency modes, so the size of the parameterisation matrix $\mathcal{R}$ is $k_l \times d_v \times d_v$.

FNO can significantly reduce the model complexity because it is parameterised in the truncated Fourier domain rather than the original high-dimensional computational

domain. However, harnessing the fast Fourier transform also means that it requires a regular domain as input, and cannot deal with complex geometry, including irregular domain boundaries or irregular meshing.

Fourier transform and modes truncating on input function $v(x)$ can be treated as decomposing $v(x)$ into the first k_l Fourier bases:

$$v(x) \approx \sum_{k=1}^{k_l} \alpha_k \omega_k(x) \quad \forall x \in D \tag{4.8}$$

where $\omega_k(x) = e^{2\pi i k x}$ is the Fourier basis, and x represents the Euclidean coordinates of domain discrete points. Note that this equation is only valid for domain D of Euclidean space. FNO requires the regular domain because the Fourier basis is defined in Euclidean space. Therefore, the key to implementing operator learning on complex geometric domains lies in how to construct a set of basis functions of the geometric domain. After that, the high-dimensional learning problem on the original computational domain can be transferred to a low-dimensional learning problem in the basis function space.

4.4 Physics-guided low-dimensional neural operator for complex geometric domains

The challenge for implementing neural operators on complex parts is constructing the basis functions of the geometric domain. The mechanism of the manufacturing process may lead to many inherent modes of the part property fields, which are typically explainable and exhibit low-frequency characteristics in the geometric space, such as the overall distortion trend of the deformation field during the machining of structural components [90], or the uneven distribution of temperature fields caused by thermal conduction during the curing of composite parts [9]. Therefore, constructing the basis functions of the geometric domain can help the neural operator model learn the inherent features and potentially reduce the complexity of the model.

This Section will first describe the solution of the frequency-domain basis functions of complex geometry based on the Laplace operator and extract attribute-related basis functions based on the proper orthogonal decomposition. Then, it will describe how the low-dimensional iterative kernel integration operator is constructed based on the two groups of basis functions.

4.4.1 Laplacian spectrum for complex geometric domains

Considering that FNO can only extract frequency information of the regular domain, an intuitive question lies in how to extract the frequency information of arbitrarily complex geometry, more specifically, how to find a set of orthogonal bases of the irregular domain D that reflect its frequency information. The eigenfunctions of the Laplace operator can also provide the frequency basis in Euclidean space, and can even be extended

to Riemannian manifolds. This Section will first introduce the Laplacian spectrum for complex geometric domains, which will be used to construct the low-dimensional kernel integral operator.

The Laplace operator, also known as Laplacian, occurs in a wide range of differential equations describing engineering problems, such as heat transfer function, Poisson's equation, diffusion equation and wave equation. [228]. For the Euclidean space $\mathbb{R}^d$, the Laplacian Δf is a second-order differential operator defined as the divergence $\nabla\cdot$ of the gradient ∇f, that is $\Delta f = \nabla^2 f = \nabla \cdot \nabla f$. It can be explicitly represented with the d-dimensions as:

$$\Delta f = \sum_{j=1}^{d} \frac{\partial^2 f}{\partial x_j^2} \tag{4.9}$$

The eigenvalue problem for the Laplacian, also known as the Helmholtz equation [229], can be defined as:

$$\Delta\phi(x) = \lambda\phi(x), \quad x \in D \tag{4.10}$$

where λ and $\phi(x)$ that satisfy this equation are defined as the eigenvalues and the corresponding eigenfunctions. In fact, the Fourier basis $\psi_k(x) = e^{2\pi \mathrm{i} kx}$ is also an eigenfunction of the Laplacian in the Euclidean space. Following Eq. 4.10, thus:

$$\Delta\left(\psi_k(x)\right) = \frac{\mathrm{d}^2}{\mathrm{d}x^2} e^{2\pi \mathrm{i} kx} = -(2\pi k)^2 e^{2\pi \mathrm{i} kx} = -(2\pi k)^2 \psi_k(x) \tag{4.11}$$

It is clear that the Fourier basis $e^{2\pi \mathrm{i} kx}$, also known as the plane wave functions, are the eigenfunction of Laplacian with the eigenvalue $\lambda = -(2\pi k)^2$. **More generally, plane waves will be eigenfunctions for any linear operation which commutes with translations [230], where the Laplacian is the most representative one.** Therefore, FFT in FNO can be treated as projecting the data from the original computational domain to the Laplacian basis. That means that the orthogonal frequency basis of the domain, the Laplacian spectrum, can play the same role as the FFT in FNO but without the limitation of regular-domain requirements. **In general, the equivalent resolution-independent neural operators can be implemented as long as the Laplacian spectrum of the domain $[\phi_1(x), \phi_2(x), \ldots, \phi_{k_m}(x)]$ can be constructed**.

This research focused on parts with complex geometric domain, namely the manifolds embedded on $\mathbb{R}^3$. But Eq. 4.9 only gives the continuous Laplacian in the Euclidean space. In fact, the Laplacian can be extended to the Riemannian manifold (also known as the Laplace–Beltrami operator [231]), such as the graph Laplacian defined on the structure of a graph [232], or the mesh Laplacian defined on a geometric mesh [233].

The Laplacians of different geometric meshes are strictly defined in the differential geometry field [234], including triangular mesh, quadrilateral mesh or tetrahedral mesh. Taking the example of the triangular mesh, as shown in Fig. 4.6 and Eq. 4.12, the discrete Laplacian of a scalar function f on a vertex i is defined by the cotangent function of the adjacent nodes, where $\mathcal{N}(i)$ is the vertex i on the geometric mesh. The Laplacian of

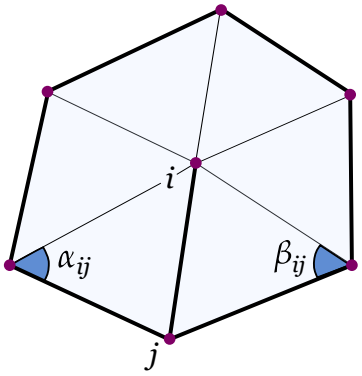


Figure 4.6: Cotangent Laplace operator of the triangular mesh.

triangular mesh is also called the cotangent Laplace operator, which can be derived in many different ways, including finite analysis, finite volume method, or discrete exterior calculus [235].

$$(\Delta f)_i \approx \frac{1}{2} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (f_i - f_j) \tag{4.12}$$

After defining the mesh Laplacian of complex geometries, the geometry-aware Laplacian spectrum $[\phi_1(x), \phi_2(x), \ldots, \phi_{k_m}(x)]$ can be solved by Galerkin method, power iteration, or other numerical methods [236, 237].

Fig. 4.7 shows the Laplace spectrum of three basic geometries, including the quadrilateral lattice sphere, quadrilateral lattice blank and triangular lattice torus. It can be observed that the Laplace spectrum exhibits the frequency information from low to high, which is similar to Fourier transformation in the regular domain. Therefore, the low-frequency information of the part property can be obtained by projecting the property field onto the low-frequency Laplace spectrum.

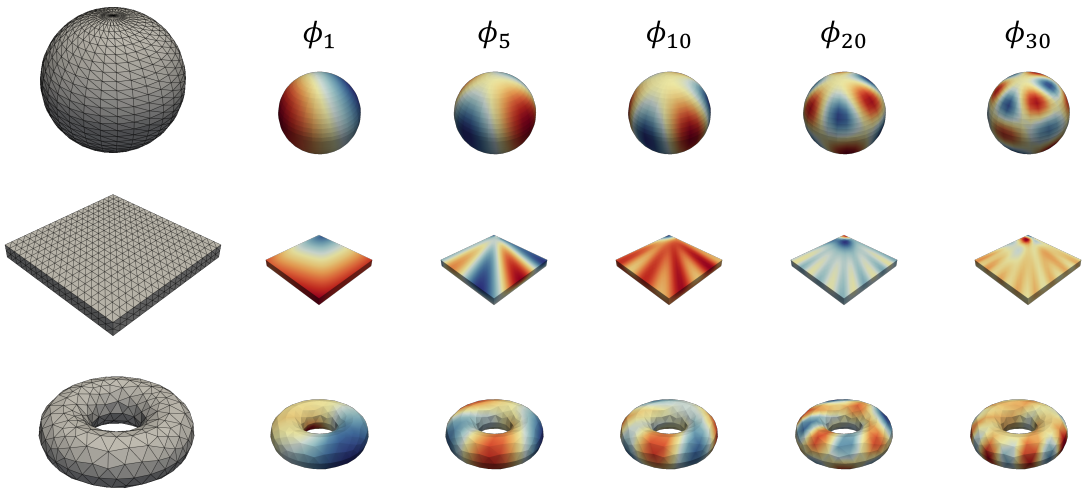


Figure 4.7: The Laplace spectrum of the basic geometries.

4.4.2 Proper orthogonal decomposition for attribution field

Proper orthogonal decomposition (POD) is a statistical dimensional reduction method commonly used in fluid dynamics, and structural analysis [238]. It can obtain low-dimensional embedding information from observed high-dimensional dataset, which can potentially be applied in the neural operator. POD calculation only involves the property information of the grid nodes in domain D without involving the Euclidean coordinates of the grid nodes. Therefore it is valid for both regular and irregular domains. This section will extract the low-dimensional modes from the to-be-predicted property field of the training dataset.

For a neural operator learning problem, both the input and output can be the property fields defined on the geometric domain. But the POD of the output field has a more significant influence on the prediction performance. For the output property field $u(x)$, POD aims to solve a set of basis functions $\psi_k(x)$ from the training dataset, thus representing the original high-dimensional data as a linear superposition of a finite number (k_p) of basis functions:

$$u(x) \approx \sum_{k=1}^{k_p} \alpha_k \psi_k(x) a \tag{4.13}$$

where α_k is the superimposed weight of each basis function.

Assuming that the training dataset consists of N samples with the output property field $\left\{u^{(i)} \in \mathbb{R}^{n_x \times d_u}\right\}, i = 1 \cdots N$. Since $k_p \ll n_x$, vector $\mathbf{a} = [\alpha_1, \cdots, \alpha_{k_v}]$ can be regarded as a low-dimensional representation of the property field u. To simplify the subsequent computation, let $d_u = 1$, then the overall output property fields of the N group of training samples is represented as $\mathbf{U} \in \mathbb{R}^{n_x \times N}$. If d_u is larger than 1, the POD of each dimension of d_u can be obtained separately in the following way.

A set of $\psi_k(x)$ for all training samples satisfying Eq. 4.13 is called a POD basis function, basis for short. The decomposition problem can be defined as an optimal least-squares approximation problem [239], and is commonly solved by Singular-Value-Decomposition (SVD), i.e. :

$$\mathbf{U} = \Psi \mathbf{\Sigma} \mathbf{V}^\top \tag{4.14}$$

where $\Psi = [\psi_1(x), \psi_2(x), \cdots, \psi_N(x)] \in \mathbb{R}^{n_x \times N}$ is a set of orthogonal POD bases, $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ is a square matrix of corresponding singular values. The original property field data can then be approximated by a finite number (k_p) of POD bases, i.e:

$$\mathbf{U} \approx \Psi_p \mathbf{\Sigma}_p \mathbf{V}_p^\top \tag{4.15}$$

where $\Psi_p = \left[\psi_1(x), \psi_2(x), \cdots, \psi_{k_p}(x)\right] \in \mathbb{R}^{n_x \times k_p}$ refers to the POD bases corresponding to the first k_p maximal singular values. $\mathbf{\Sigma}_p$ and $\mathbf{V}_p^\top$ are the truncated matrices of $\mathbf{\Sigma}$ and $\mathbf{V}^\top$, respectively.

Fig. 4.8 shows the POD bases of the deformation and temperature fields of a composite part in the experimental validation. The vertical coordinates are the singular values for the different basis, representing the corresponding influential weight in the original

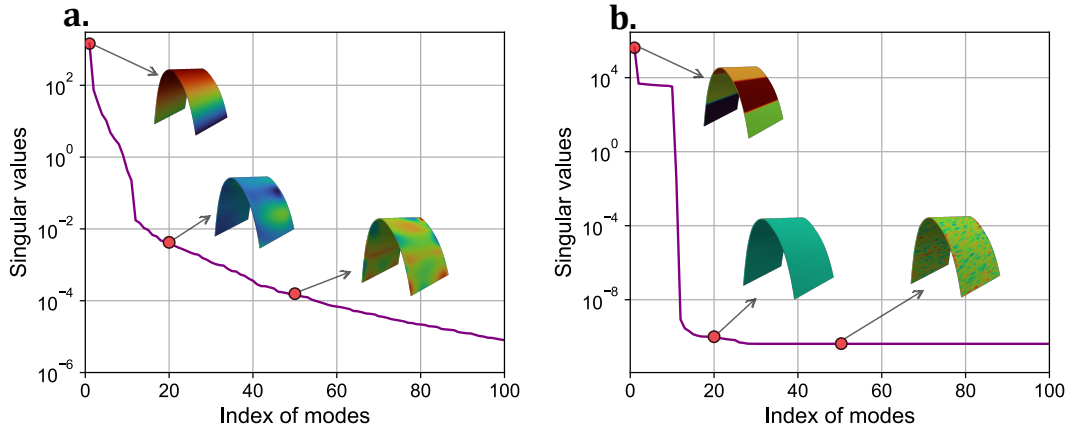


Figure 4.8: The proper orthogonal decomposition of the deformation field and temperature field.

field. As shown in Fig. 4.8a, the maximum singular value of the deformation field exceeds 100, while the weight after the 10-th basis is considerably less than 1. A similar trend is also shown in Fig. 4.8b. This result indicates that a few POD bases are sufficient to represent the main patterns of the property field, which means the integration of POD bases can potentially enhance the feature extraction of the neural operator. However, as POD is a data-driven dimensional reduction method, the accuracy and generalisability are affected by the given dataset, namely $\mathbf{U}$. The size and noise of the training set $\mathbf{U}$ can influence the generalisability of the obtained POD basis. Additionally, the maximum POD basis is equivalent to the number of training samples, namely N, which may lead to inadequate POD bases when the number of training samples is limited.

4.4.3 Low-dimensional kernel integral operator

The POD bases can reflect the embedding modes of the property field, while the Laplace spectrum is a group general basis that can represent the frequency information of the geometric domain. Combining the two sets of bases (property-related and geometry-related) can enhance the feature extraction capability of the neural operator model.

In this research, a low-dimensional iterative kernel integral operator was constructed based on the combined bases. Supposing that the combined bases consist of k_l Laplace bases and k_p POD bases, i.e., are denoted as $\Phi = \left[\phi_1(x), \ldots, \phi_{k_l}(x), \psi_1(x), \ldots, \psi_{k_p}(x)\right] \in \mathbb{R}^{n_x \times k_m}$, where $k_m = k_l + k_p$ is the total number of basis functions.

The low-dimensional kernel integral operator, namely the L-layer in Fig. 4.3, consists of encoder $\mathcal{E}$, parameterisation mapping $\mathcal{R}_\theta$ and decoder $\mathcal{D}$ [240], namely:

$$\mathcal{K}_\theta(v_t) := \mathcal{D} \circ \mathcal{R}_\theta \circ \mathcal{E} \tag{4.16}$$

where encoder $\mathcal{E}$ projects input function $v_t : L^2(D; \mathbb{R}^{d_v})$ to the Laplacian spectrum $[\phi_1(x), \phi_2(x), \ldots, \phi_{k_m}(x)]$. After that, parameterisation mapping $\mathcal{R}_\theta$ can be defined in the finite-dimensional space $\mathbb{R}^{k_m}$, and the size of parameters only depends on the size of

the truncated mode k_m. Lastly, the new feature is mapped back to the output function space $v_{t+1} : L^2(D; \mathbb{R}^{d_v})$ by the decoder $\mathcal{D}$. The three-step procedures can be represented in the following diagram:

$$\begin{array}{ccc} L^2(D;\mathbb{R}^{d_v}) & \xrightarrow{\mathcal{K}_\theta} & L^2(D;\mathbb{R}^{d_v}) \\ \downarrow \mathcal{E} & & \mathcal{D} \uparrow \\ \mathbb{R}^{k_m \times d_v} & \xrightarrow{\mathcal{R}_\theta} & \mathbb{R}^{k_m \times d_v} \end{array} \tag{4.17}$$

When the input space is discretised into n_x nodes, the input function $v_t(x)$ can be represented discretely as a $\mathbf{V}_t \in \mathbb{R}^{n_x \times d_v}$. Note that input matrix $\mathbf{V}_t$ does not contain the coordinate of the discrete points or other geometric information in the domain, but only lists the function values of n_x discrete points. When performing FFT on the regular domain, the input function is defined in the grid coordinates of the Euclidean space, two-dimensional or three-dimensional spacial coordinates will bring extra dimensions for the input data. However, for the manifolds represented by mesh grids, the input and output data only contain the function value of each node, x in $v_t(x)$ actually expresses the index of the node rather than the node coordinates. Both the Laplace and POD bases are discretised into a vector form as $\phi_k \in \mathbb{R}^{n_x \times 1}$, $\psi_k \in \mathbb{R}^{n_x \times 1}$, and all k_m bases comprise the basis matrix $\Phi \in \mathbb{R}^{n_x \times k_m}$. The complex geometric information of the domain has been embedded in the Laplacian spectrum. The entire implementation of the low-dimensional kernel integral operator is defined on the node index rather than the Euclidean coordinates, which means the 2D or 3D geometric domain shares the same structure without increasing the complexity of the model.

The encoder of the discretised input matrix $\mathbf{V}_t$ can be expressed as:

$$\mathcal{E}(\mathbf{V}_t) = \Phi^\dagger \mathbf{V}_t \tag{4.18}$$

where $\Phi^\dagger \in \mathbb{R}^{k_m \times n_x}$ refers to the pseudo inverse of the Laplacian spectrum matrix Φ, defined as:

$$\Phi^\dagger = (\Phi^\top \Phi)^{-1} \Phi^\top \tag{4.19}$$

Denoting the parameterisation mapping $\mathcal{R}_\theta$ as the matrix $\mathbf{R} \in \mathbb{R}^{k_m \times d_v \times d_v}$, the mappings on the encoded frequency information can then be represented as:

$$\mathcal{R}_\theta \circ \mathcal{E}(\mathbf{V}_t) = \left(\mathbf{R} \cdot \left(\Phi^\dagger \mathbf{V}_t\right)\right) \tag{4.20}$$

where the tensor operation is defined as:

$$\left(\mathbf{R} \cdot \left(\Phi^\dagger \mathbf{V}_t\right)\right)_{k,l} = \sum_{j=1}^{d_v} \mathbf{R}_{k,l,j} (\Phi^\dagger \mathbf{V}_t)_{k,j}, \quad k = 1, \ldots, k_m, \quad l = 1, \ldots, d_v \tag{4.21}$$

The decoder process is simply the linear transformation with the Laplacian spectrum matrix:

$$\mathcal{D} \circ \mathcal{R}_\theta \circ \mathcal{E} = \Phi \left(\mathbf{R} \cdot \left(\Phi^\dagger \mathbf{V}_t\right)\right) \tag{4.22}$$

The neural operator with the above-defined low-dimensional kernel integration layer constitutes the proposed Low-dimensional Neural Operator (LNO). LNO is similar to FNO in that both are discretisation-invariant by the frequency-domain parametrisation. But LNO has two notable advantages:

- **The generalisability on the complex geometries:** Fourier transform is only applicable to rectangular domains with regular grids, while the Laplacian spectrum can be generalised from Euclidean space to Riemannian geometry, including 2D, 3D surfaces or 3D solid models that are discretised by triangular mesh, quadrilateral mesh or tetrahedral mesh.
- **The generalisability on the domain dimensions:** The Laplace integral operator implementation is defined in the node index rather than the Euclidean coordinates of the nodes, thus the LNO framework is general for two or three-dimensional, regular or irregular geometries.

4.5 Case studies

This Section describes the verification of the proposed LNO on the 2D Darcy flow problem with complex domain and the 3d composite part curing deformation prediction problem. The proposed LNO was compared with the popular neural operator DeepOnet and its variant POD-DeepOnet. For the 2D Darcy case, the complex domain was interpolated to a regular domain for the implementation of FNO. For the 3d composite part case, FNO was not implemented because of the prohibitive complexity of 3d spatial interpolation.

4.5.1 Darcy flow

4.5.1.1 Problem definition

Darcy flow equation is a classical law for describing the flow of fluid through a porous medium, widely used in various engineering fields, such as resin flow simulation in composites manufacturing [241]. Darcy problem is also adopted as the benchmark for the performance verification of various neural operators [211, 220]. This case study focused on the Darcy equation on 2D irregular geometric domain:

$$-\nabla \cdot (a(x,y)\nabla u(x,y)) = f(x,y) \quad x,y \in D \tag{4.23}$$

where $a(x,y)$ is the diffusion coefficient field, $u(x,y)$ is the pressure field and $f(x,y)$ is the source term to be specified.

The experiments on Darcy flow consisted of two cases, involving different geometric meshes and different boundary conditions. The geometric domain had an irregular boundary with a thin rectangle notch inside, which could increase the complexity of the boundary condition. As shown in Fig. 4.9a, the geometric domain of case 1 was a

triangle mesh with 2282 modes, where the outside boundary condition (red boundary) follows $u = u_{\partial D}(x)$ and the three boundaries of the inside rectangle follows $u = 0$. The geometric domain of case 2 was a triangle mesh with 592 nodes. As shown in Fig. 4.9b, the two boundaries of the inside rectangle had different boundary conditions. The left and right sides had a close Euclidean distance but a relatively far geodesic distance, which could bring more challenges to neural operator learning. The boundary condition functions $u_{\partial D}(x)$ in Fig. 4.9c were generated from a gaussian process defined in Ref. [220].

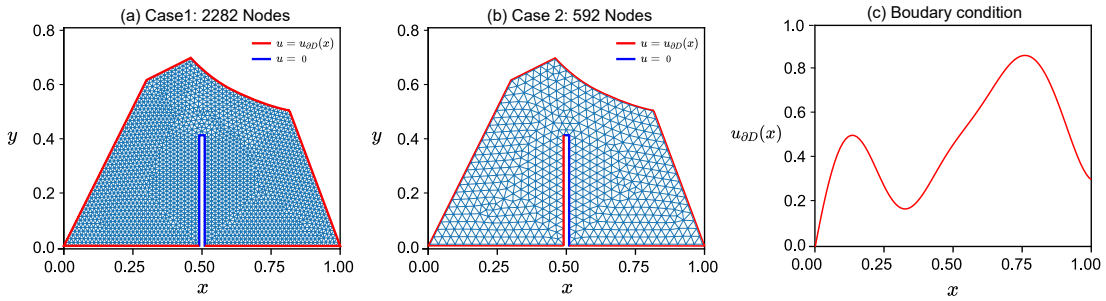


Figure 4.9: The geometric domain and boundary conditions of the Dary flow cases.

The learning target in the Darcy flow problem was the mapping from the diffusion coefficient field $a(x, y)$ to the pressure field $u(x, y)$:

$$\mathcal{G} : a(x, y) \mapsto u(x, y) \tag{4.24}$$

where the source term was set to 1, i.e. $f = 1$. The input diffusion coefficient field $a(x, y)$ was generated by the Gaussian random field with a piecewise function, namely $a(x, y) = t(\mu)$, where μ is a distribution defined by $\mu = \mathcal{N}(0, (-\Delta + 9I)^{-2})$ [219]. After sampling from this distribution, the diffusion field $a(x, y)$ could be generated from the following piecewise function:

$$t(\mu) = \begin{cases} 12, & \mu \geq 0 \\ 3, & \mu < 0 \end{cases} \tag{4.25}$$

The labelled data for training the neural operator model was the pair of $a(x, y)$ and $u(x, y)$. 1200 sets of input data $a(x, y)$ were randomly generated first. Then the corresponding $u(x, y)$ was solved by Matlab's SOLVEPDE toolbox [242]. 1000 of them were used as the training data, and the rest 200 samples were defined as the test data.

4.5.1.2 Experimental settings

Considering that the compared operator models have similar architectures, the selection of model parameters follows these principles: (1) The common parameters of all models are set to the same values, that is, the training learning rate is 0.001, the batch size is 100, and the iteration is 1000. (2) The parameters of each model are optimised based on the recommended values from the public source code.

LNO: The network structure of LNO is designed similarly to Darcy case studies of FNO [211], including 4 L-layers, $k_l = k_p = 32$, $d_v = 32$, the learning rate is 0.001, the batch size is 100, the number of iterations is 1000, and the optimizer is Adam. Note that the combined bases Φ and its pseudo-inverse $\Phi^\dagger$ are pre-calculated before training, and the forward training of the low-dimensional kernel integral follows Eq. 4.22.

FNO: Since the original FNO cannot deal with the complex geometric domain, the mesh interpolation solution from research published in the paper [220] is adopted to construct a 50*50 regular grid for FNO. The function values of the new mesh nodes are fitted by the original irregular nodes. The numbers of Fourier modes in both directions are $k_m = 20$, and the network width parameter is set to $d_v = 64$.

DeepOnet: The structure of the DeepOnet model follows the original settings of Darcy flow case in the paper [220], where the branch network is a fully connected network with hidden layers of 256*256*100, and the trunk net is the fully connected network with hidden layers of 128*128*128*100.

POD-DeepOnet: POD-DeepOnet is the latest variant of DeepOnet, in which the branch network can directly learn the weights of the POD basis. The case study was compared with POD-DeepOnet because the POD-based dimensional reduction technique is widely used and proven effective in PDE solving and surrogate modelling problems. The model structure followed the original setting in the paper [220], and the size of the POD basis is set to 64, equal to k_m.

The loss function for all methods in this experiment is the relative L_2 loss that is defined as below:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \frac{\left\| u^{(i)} - \mathcal{G}_\theta \left(a^{(i)} \right) \right\|_{L^2}}{\left\| u^{(i)} \right\|_{L^2}} \tag{4.26}$$

4.5.1.3 Experimental results

Fig. 4.10 shows the convergence of training loss and test loss of all methods in Case 1. All methods were trained at the same learning rate (0.001) for the same number of iterations (1000). DeepOnet and POD-DeepOnet can achieve a very small training loss of around 10^{-4} after 400 iterations, which is much smaller than the 10^{-2} of LNO and FNO. As to the test loss from Fig. 4.10(b), LNO can archive 10^{-2} in only 100 iterations, which convergences much faster than the other three methods.

Tab. 4.1 shows the final training results of each method in Case1, where A(B) refers to the mean and standard variation of 10 repeated trials. Both the training and test errors of FNO are more than 2%, which means that the reconstruction of the regular grid brings significant approximation error of the frequency information. DeepOnet and POD-DeepOnet can provide a test error of 1.8% with a training error close to 0. In comparison, LNO has a superior performance with a test error of only 1.38%.

Fig. 4.11 presents the comparative prediction results of one test sample in Case 1. Fig. 4.11a is the mesh grid of the domain. Fig. 4.11b is the input diffusion coefficient field

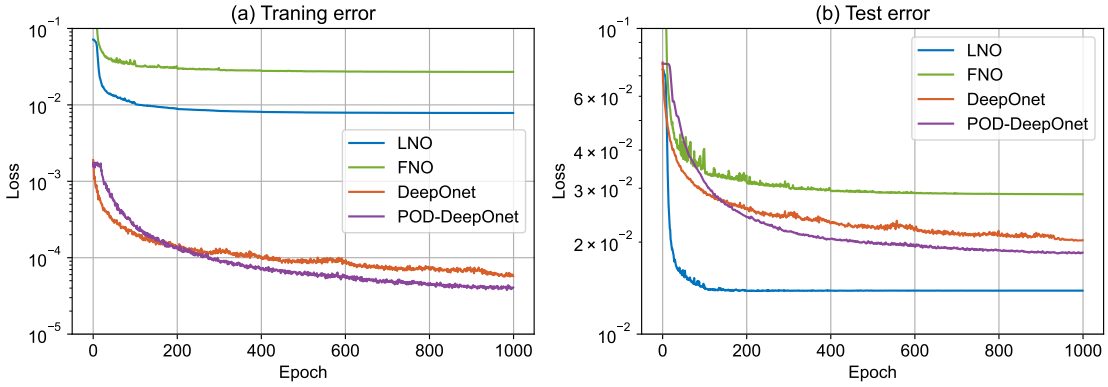


Figure 4.10: The convergence of loss functions for Case 1 of Darcy flow problem

Table 4.1: The performance comparison for Case 1 of Darcy flow problem.

Method	**Parameter**	**Train error(%)**	**Test error(%)**
FNO(Grid)	13,132,545	2.712(0.076)	2.780(0.081)
DeepOnet	722,248	0.004(0.000)	1.878(0.025)
POD-DeepOnet	3,722,200	0.004(0.000)	1.839(0.020)
LNO	532,961	0.820(0.195)	**1.384(0.021)**

Training error and test error are relative L_2 error.

$a(x, y)$ generated by Eq. 4.25. Fig. 4.11c is the reference value of the output pressure field $u(x, y)$ solved by Matlab. Fig. 4.11d is the pressure field predicted by the LNO. Fig. 4.11e-h are the prediction errors of FNO, DeepOnet, POD-DeepOnet and LNO, respectively. Δ_{mean} in each figure refers to the average absolute error over all nodes in the geometric domain, and Δ_{max} is the maximum absolute error on all nodes. Due to inaccurate grid interpolation, FNO has the most significant error (up to 0.16), especially in the boundary region. DeepOnet and POD-DeepOnet show significant errors on the right side of the rectangle where the output field has a large gradient. Meanwhile, LNO can reduce Δ_{mean} and Δ_{max} significantly compared with all other methods.

Tab. 4.2 shows the final training results of each method in Case2. The overall results are similar to Tab. 4.1, with LNO also achieving the smallest test error and DeepOnet and POD-DeepOnet still showing significant overfitting. Note that the parameterisation of FNO and LNO is independent of the grid resolution, so Case 1 and Case 2 can share the same network architecture with exactly the same number of parameters. By comparison, the number of parameters of DeepOnet and POD-DeepOnet increases with the grid size.

Fig. 4.12 shows the comparative prediction results of one test sample in Case 2 of Darcy flow problem. Due to the inconsistent left and right boundary conditions in the rectangular region, the output field shows an abrupt change near the upper boundary of the rectangle. All four methods have significant errors in this region, while LNO is still able to output more accurate predictions, with Δ_{mean} and Δ_{max} smaller than the other three methods.

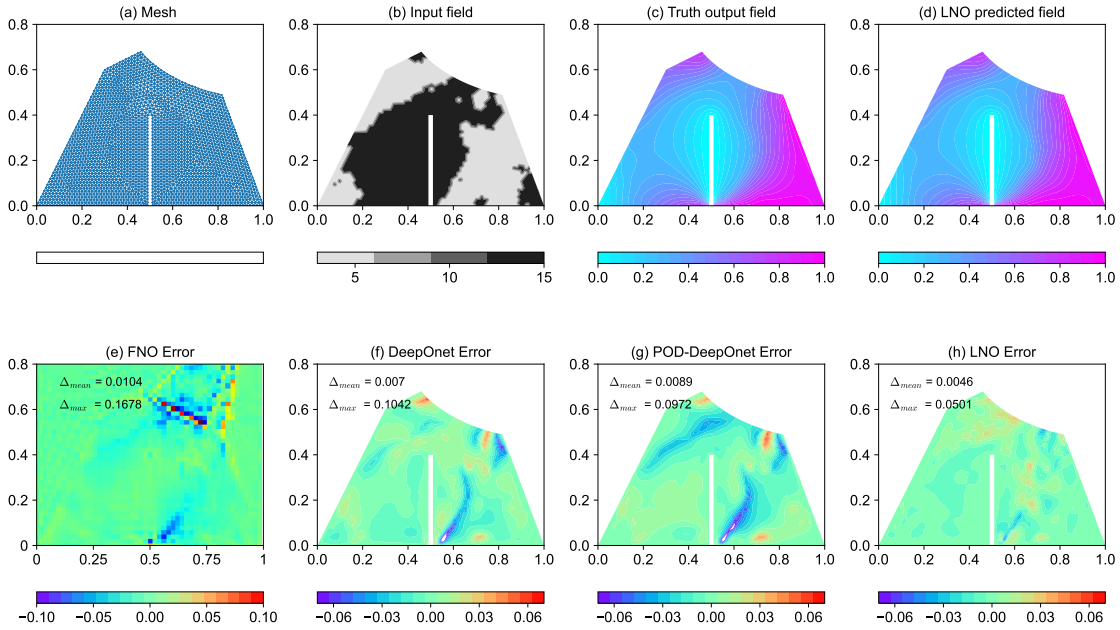


Figure 4.11: The comparison of prediction results for Case 1 of the Darcy flow problem.

Table 4.2: The performance comparison for Case 2 of Darcy flow problem.

Method	**Parameter**	**Train error(%)**	**Test error(%)**
FNO(Grid)	13,132,545	2.490(0.094)	2.670(0.087)
DeepOnet	289,608	0.004(0.000)	1.927(0.023)
POD-DeepOnet	1,018,200	0.004(0.000)	1.896(0.030)
LNO	532,961	1.001(0.051)	**1.681(0.014)**

Training error and test error are relative L_2 error.

4.5.2 Composite part deformation prediction

4.5.2.1 Problem definition

This Section will describe the investigation of the effectiveness of the proposed LNO method on a complex 3d irregular geometry, namely predicting the curing deformation of a Carbon Fiber Reinforced Polymer (CFRP) composite part based on a given temperature field. The large size, complex shape and high accuracy requirements of aerospace CFRP parts impose increased demands on deformation control during the curing process [9]. To reduce the distortion of curing, the part geometry was divided into several regions to apply different curing temperatures. Therefore, constructing the predictive model of the temperature field to deformation field on the geometry can provide essential support for further curing process optimising. As shown in Fig. 4.13, the learning problem of this case is defined as the mapping from the temperature field $a(x, y, z)$ to the deformation field $u(x, y, z)$ on the given composites part.

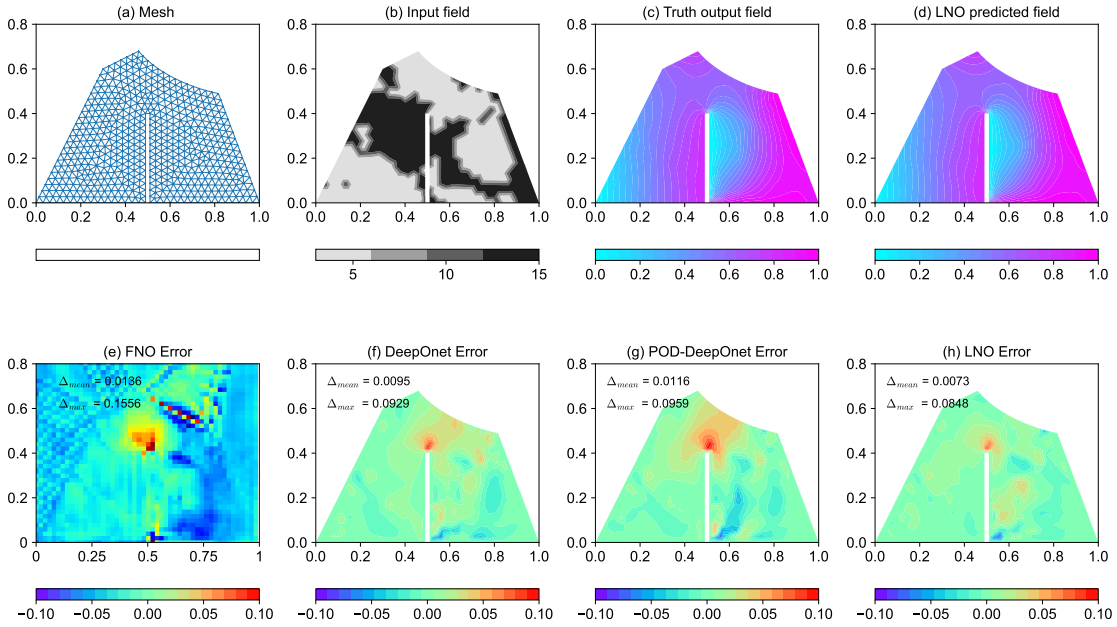


Figure 4.12: The comparison of prediction results for Case 2 of the Darcy flow problem.

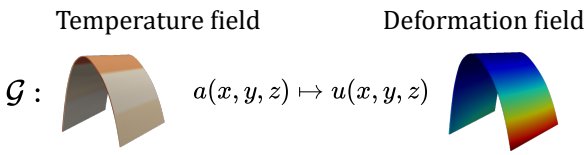


Figure 4.13: The problem definition of the composite part deformation prediction problem.

As shown in Fig. 4.14a, the part geometry is represented by a tetragonal mesh constructed in the Finite Element Method (FEM) simulation[243], comprising a total of 8576 nodes. The input temperature field is designed according to the actual requirements and constraints of the curing process. The internal and external surfaces of the composite part are divided into 20 separate curing zones, with the temperature of each zone generated randomly between the $370K \sim 400K$. Therefore, the input data of the neural operator learning problem is the temperature values on 8576 grid nodes. The deformation fields of the composite part were simulated by FEM considering heat transfer, curing reactions, viscoelastic mechanics and other processes [9]. A total of 300 data pairs of temperature-to-deformation fields were simulated, 200 of them are defined as training data and the rest 100 as test data.

4.5.2.2 Experimental settings

Considering the prohibitive computational burden of the spatial mesh interpolation of 3D parts, FNO is not implemented in this task, and this research only compares the proposed LNO with DeepOnet and POD-DeepOnet. The basis sizes of LNO are also

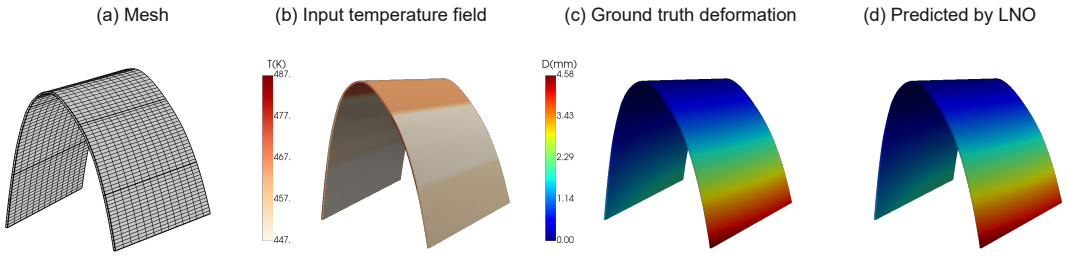


Figure 4.14: The composite part curing deformation prediction problem. (a) Mesh of the composite part. (b) The input temperature field of a test sample. (c) The ground truth deformation field of a test sample. (d) The predicted deformation field of a test sample.

$k_l = k_p = 32$. The other parameters are consistent with the Darcy flow case. Meanwhile, this section also compares the reduced-order regression method, which is commonly used for finite element simulation surrogate modelling problems in engineering [244]. High-dimensional input and output fields are usually translated into a small number of representative indicators, such as POD mode weights or other statistical indicators [245]. In this case, a Gaussian process reduced order model (marked as GP-POD) was developed to predict the POD eigenvalues of the deformation field. The input of GP-POD is a 10-dimensional vector representing the temperatures of the 10 curing regions, and the output of the model is the top 50 POD eigenvalues of the deformation field solved from the training data. And the final deformation field can be reconstructed based on the POD eigenvectors and the predicted POD eigenvalues.

4.5.2.3 Experimental results

The quantitative performance of different methods is shown in Tab. 4.3, the values in brackets refer to the standard deviation of ten repeated experiments. The test error of LNO is only 0.22%, much smaller than that of GP-POD and DeepOnet (1.21% and 0.51%), and the number of parameters of the LNO model is also smaller than DeepOnet and POD-DeepOnet. Relative $L2$ error can reflect the overall prediction performance on the entire data set, but as composite curing is a risk-sensitive problem, it is more meaningful to focus on the maximum error of the predicted deformation field predicted. E_{max} in Tab. 4.3 is defined as the maximum absolute value of the deformation prediction error over all nodes of all test samples. It can be seen that the E_{max} of LNO is only $0.015mm$, which is much smaller than the other comparison methods, while the standard deviation of the maximum error predicted by LNO is only 0.001mm.

Figures 4.14b-d depicts the LNO predicted result of a test sample, where b-d refer to the input temperature field, the ground truth deformation field and the predicted deformation field, respectively. The deformation field predicted by FNO is quite close to the reference value. Fig. 4.15 depicts the deformation field prediction errors of the four methods on the sample shown in Figures 4.14. GP-POD and DeepOnet have large and non-uniform errors on the right and bottom regions of the part, while POD-DeepOnet shows a more uniform global error on the upper side of the part. By comparison, the LNO is 'light green' over the entire part, with obviously far less error than the other

Table 4.3: The performance comparison of different methods on the composite case.

Method	**Parameters**	**Test error(%)**	E_{max}(mm)
GP-POD	-	1.21(0.06)	0.181(0.020)
DeepOnet	828,872	0.51(0.03)	0.028(0.003)
POD-DeepOnet	4,396,488	0.47(0.07)	0.027(0.004)
LNO	68,977	**0.22(0.03)**	**0.015(0.001)**

three methods. The maximum deformation prediction error over all nodes is marked separately for different methods in Fig. 4.15. It can be seen that the maximum errors of GP, DeepOnet and POD-DeepOnet are all higher than 0.1mm, while the maximum prediction error of LNO is only 0.004mm.

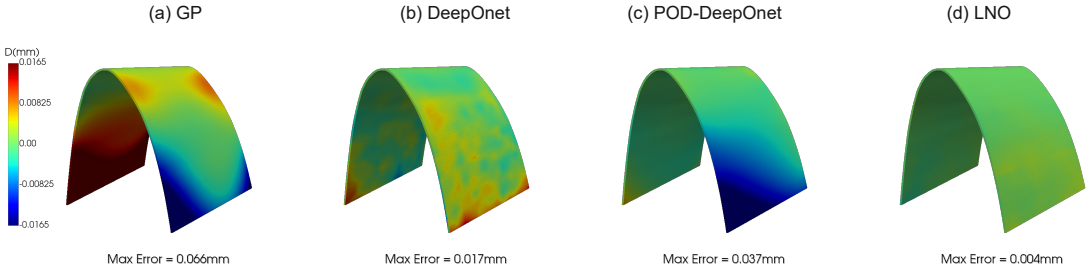


Figure 4.15: The comparison of deformation prediction error.

The training convergence of different methods is shown in Fig. 4.16. The training error of all three methods reaches 0.05% after 1000 iterations. However, there is a significant difference in the test error convergence of the three methods. LNO achieves an error of less than 0.5% in only about 100 iterations, which is significantly faster than other methods.

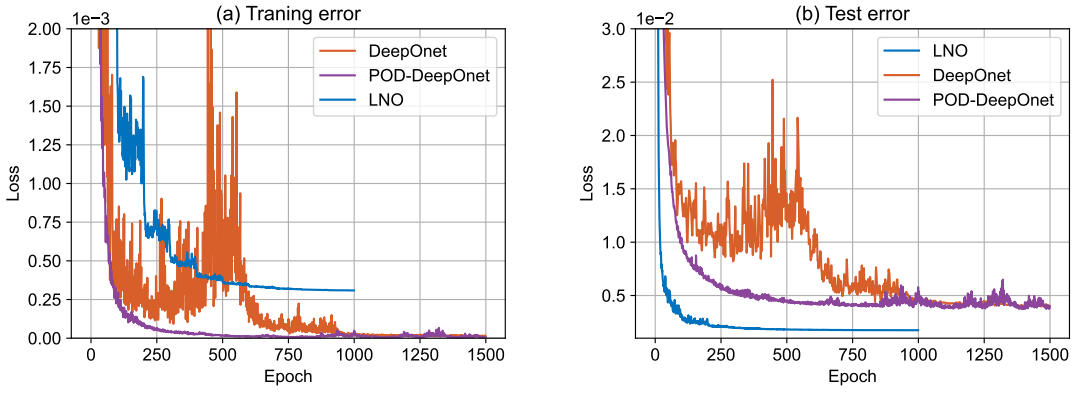


Figure 4.16: The convergence of loss functions for the composite case.

To verify the performance of different methods under the limited label data scenario. Fig. 4.17 presents the test error and maximum error (E_{max}) for each method with 100 training samples and 200 training samples. With a smaller sample size, The test error

and E_{max} for all methods increase significantly, with LNO still maintaining the lead. It can also be seen that the LNO achieves a test error of 0.045% and a maximum error of 0.027mm with 100 samples. By contrast, DeepOnet and POD-DeepOnet require 200 samples to achieve similar error performance.

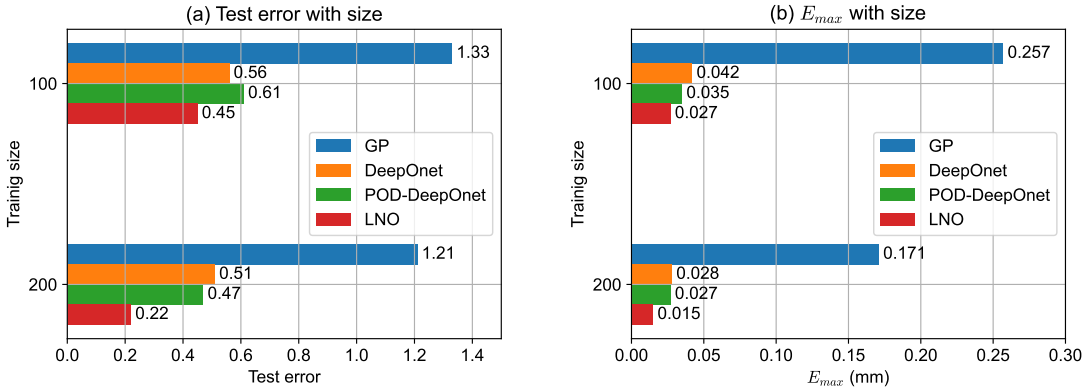


Figure 4.17: The convergence of loss functions for the composite case.

4.5.3 Analysis and discusions

(1) Node-level prediction errors

The above experiments only investigated the L_2 error and the maximum prediction error for each case, this Section will analyse the distribution of prediction error over all nodes of all test samples. Fig 4.18a-c show the statistical results of the three cases. It can be seen that the prediction errors of all nodes for all methods show Gaussian distributions with mean values approximating 0. The estimated standard deviations of different methods are marked in each figure. In Fig. 4.18a, Case 1 of the Darcy flow problem, the nodes' standard deviation for LNO is $\sigma = 0.0073$, which is slightly smaller than the remaining three methods. In Fig. 4.18c, the composite case, the standard deviation of the LNO is only $\sigma = 0.0164$mm, which is substantially lower compared to existing methods. In summary, LNO reduces the prediction error uniformly and comprehensively for most nodes.

(2) Laplacian and POD basis

LNO contains geometric-related Laplacian bases and attribute-related POD bases. The two groups of bases have different characteristics, and thus are able to enhance the feature extraction capability of the model. Fig. 4.19 shows the LNO for the composite case with different base combinations, with the same total number of bases of $k_m = 64$. 'TWO basis' means 32 Laplacian bases and 32 POD bases, 'POD basis' stands for only 64 POD bases, namely $k_l = 0, k_p = 64$, and 'LBO basis' means only 64 Laplacian bases i.e. $k_l = 64, k_p = 0$. As shown in Fig. 4.19a-b, in both the training and test steps, the combination bases, 'Two basis', can convergent to a very low loss quickly, much

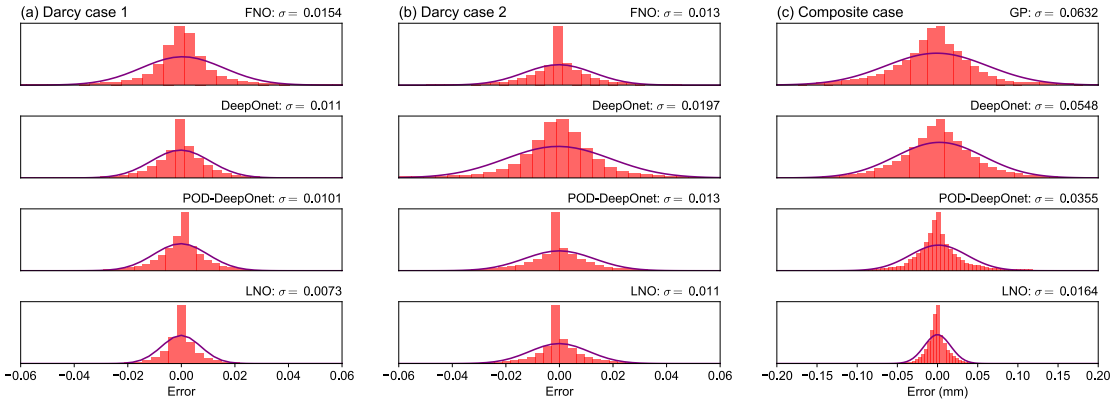


Figure 4.18: The distribution of deformation prediction error over all nodes. (a) Case 1 in the darcy flow problem. (b) Case 2 in the darcy flow problem. (c) Composite case.

better than the single bases results, which means the two groups of bases can provide complementary physics information of feature extraction in LNO.

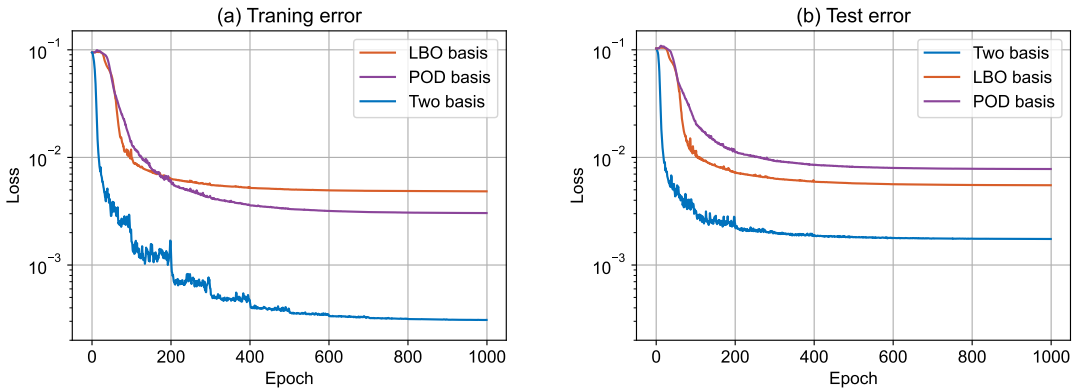


Figure 4.19: The training convergence of LNO under different bases.

4.6 Summary

Predicting high-dimensional property fields of parts can provide valuable support for quality control and process optimisation. However, the geometry representation and the feature extraction normally require complex parametric models and extensive labelled data. This research proposed a physics-guided low-dimensional neural operator, which transforms the issue of high-dimensional field mapping of part property field into a low-dimensional mapping issue in the basis domain. The physics-guided bases, including geometric-related and attribute-related bases, are embedded into the neural network structure to enhance the feature extraction ability.

The proposed the physics-guided low-dimensional neural operator can represent complex geometry with a limited number of basis, thus can significantly reduce the

parameter complexity of the data-driven models. The experiment on the 3D composite deformation field prediction case demonstrated that the proposed model could provide an accurate prediction result with fewer labelled data. Further analysis revealed that the two groups of bases can provide complementary information for feature extraction. The proposed LNO could potentially be applied in various part property field prediction problems, such as deformation or stress field prediction for curing, machining or assembling. A well-established LNO model could also support process optimisation by utilising the collected simulation data or experimental data. Chapter 5 will describe the combination of the LNO model with the sampling method developed in Chapter 2 and the transfer learning method developed in Chapter 3, and the validation of the effectiveness in a more complex manufacturing case.

Chapter 5

Validation of the developed data-driven curing deformation prediction system by case studies

All models are wrong, but some are useful.

– George E. P. Box

This Chapter describes the validation of the developed technologies in a composite part deformation prediction case. A data-driven deformation prediction system was developed based on the methods described in Chapter 2, 3, and 4. The effectiveness of the system was validated on a complex Carbon Fiber Reinforced Polymer part.

5.1 Data-driven curing deformation prediction system for composites manufacturing

5.1.1 Introduction of the case study

Carbon Fiber Reinforced Polymer (CFRP) composite materials, which are lightweight and high-strength, are preferred materials for weight reduction and performance enhancement in modern aerospace industries [246]. CFRP parts used in aerospace have large size and complex shapes, therefore imposing higher requirements on deformation control during the manufacturing process. As one of the key processes of composites manufacturing, curing refers to using high temperatures to stimulate the chemical reactions and physical changes of the resin, thereby forming CFRP parts with load-bearing properties. Non-uniform residual stresses generated during the curing process can cause curing deformations such as spring-back, warpage, and bending-twisting combination, which not only risks the CFRP parts being scrapped but also becomes an important reason for damages and failures during subsequent assemblies [247].

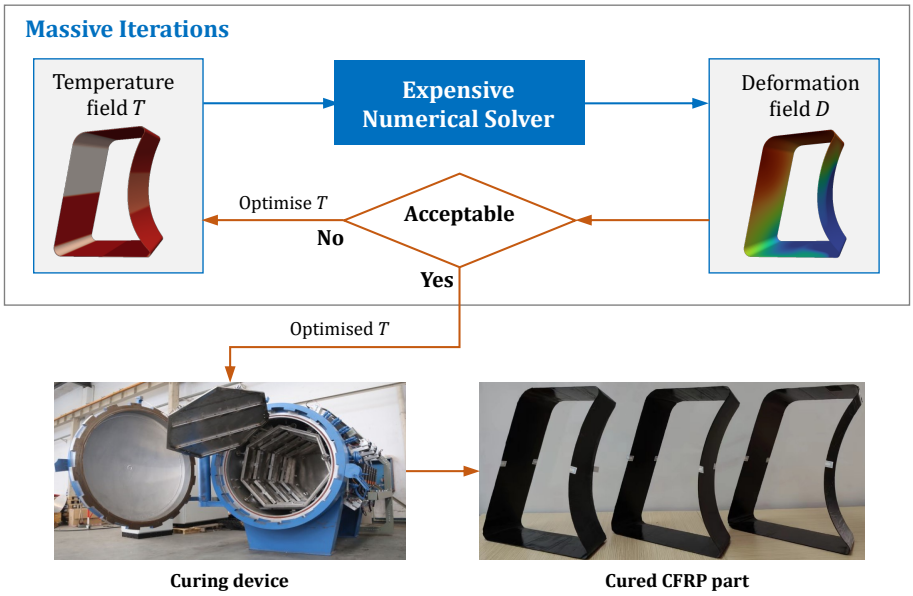


Figure 5.1: The composite curing problem.

Regulating the curing temperature distribution of a part is an effective means of controlling curing deformation. As shown in Fig. 5.1, optimising the curing temperature field usually requires a large number of iterations based on the prediction results of the curing deformation field. Therefore, establishing a fast prediction model from the

curing temperature field to the deformation field is of great significance for optimising and designing the temperature field of CFRP parts [11]. Numerical simulation methods, such as the finite element method, have become the most widely used curing process modelling methods. However, high-fidelity curing deformation simulation requires accurate modelling of complex physicochemical processes and fine meshing of the part calculation domain, resulting in highly expensive and time-consuming calculations. For example, it would take several hours to simulate the complete deformation field of the Boeing 787 wing skin part during the curing process, and hundreds of labelled training data would require several months to obtain. Therefore, the computational efficiency of the traditional numerical modelling methods is insufficient to meet the requirements for the temperature field optimisation of the CFRP parts. This Chapter will describe a developed data-driven curing deformation prediction system based on the proposed methods described in previous Chapters to achieve accurate and efficient prediction.

5.1.2 The developed data-driven curing deformation prediction system based on the proposed methods

Since traditional data-driven deformation predicting modelling rely on a large number of high-fidelity data samples, this research developed a data-driven curing deformation prediction system based on the proposed methods, aiming to reduce the number of high-fidelity data samples required while ensuring model prediction accuracy. The complex physicochemical processes and fine meshing in the curing simulation lead to the very time-consuming acquisition of high-fidelity simulation data, whilst a large amount of low-fidelity simulation data could be obtained quickly by simplifying the curing process or simplifying the meshing. Although the prediction accuracy of low-fidelity simulation data cannot meet the requirements of subsequent curing temperature field optimisation, it can provide auxiliary information for high-fidelity sample design and data-driven model training.

The main elements and interaction between the elements of the system are shown in Fig. 5.2. The system consists of three modules: the sampling module (described in Chapter 2), the LNO modelling module (described in Chapter 4) and the transfer learning module (described in Chapter 3). For the CFRP parts to be manufactured, a large amount of low-fidelity simulation data was first generated by the finite element simulation software. The sampling module used the low-fidelity simulation data to construct a value function, and then guided the generation of a small amount of high-fidelity data based on the proposed AV4Sam. At the same time, the LNO modelling module trained a neural operator model based on the low-fidelity simulation data, regarded as the source model. The transfer learning module used the generated small amount of high-fidelity simulation data to update the source model, thus constructing a highly accurate CFRP construction curing deformation prediction model.

The framework and core algorithms were developed based on the Python language [248]. The data pre-processing and statistical analysis was carried out based on the sklearn library[249]. The data-driven models were parameterised and optimised based on the automatic differentiation library Pytorch[250]. The geometric information pro-

cessing and extraction were carried out based on Pyvista[251]. The visualisation interface was developed on PyQt5[252]. The following sections will introduce the interface and functions of the three modules.

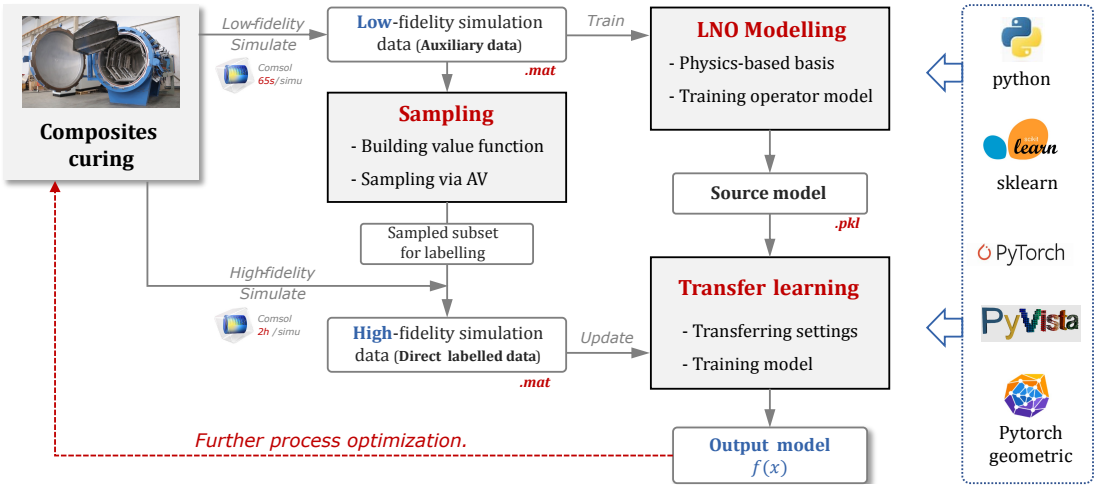


Figure 5.2: The framework of the developed data-driven deformation prediction system.

5.1.3 The sampling module

The sampling module aims at selecting a small set of samples from the part temperature field sample space that is of greater value for model training. This section uses the C-shaped part described in Chapter 4 as an example to illustrate the function and operation flow of the software module. First, 300 groups of temperature field data are randomly generated from the part curing temperature design space, and the corresponding deformation field data are obtained using a simplified finite element simulation. After that, 300 temperature-to-deformation data pairs constitute the low-fidelity data set. The purpose of this module is to use cheap low-fidelity simulation data to evaluate a small set of samples that are more valuable for data-driven model training, so as to guide the generation of high-fidelity simulation data. The software interface of this module is shown in Fig.5.3.

The sampling module first imports the simulated low-fidelity data set to evaluate the value function of samples. The basic learners for valuation include ANN from the Pytorch library and Gaussian process regression models from the GPytorch library. After setting the convergence index and parameter tolerance, the Shapley values of all samples can be calculated, and the performance of the model of adding/removing samples is also plotted in the software, which can provide a reference for the user to observe the effectiveness of the value function. After obtaining the value function, the optimal temperature field samples can be determined by clicking the button 'Start Sampling'. The default kernel function is the RBF function.

As the case set up in Fig.5.3, the sampled 20 temperature fields will be stored in .mat or .csv format as the output of this module. This means that these 20 samples are the most

valuable temperature samples for data-driven model training among the input 300 sets of samples. The sampled temperature field data will be imported to FEM software through a Python script to carry out the high-fidelity deformation simulation. The simulated 20 temperature-to-deformation data pairs constitute the high-fidelity data, which is the direct labelled data mentioned in Fig. 5.2.

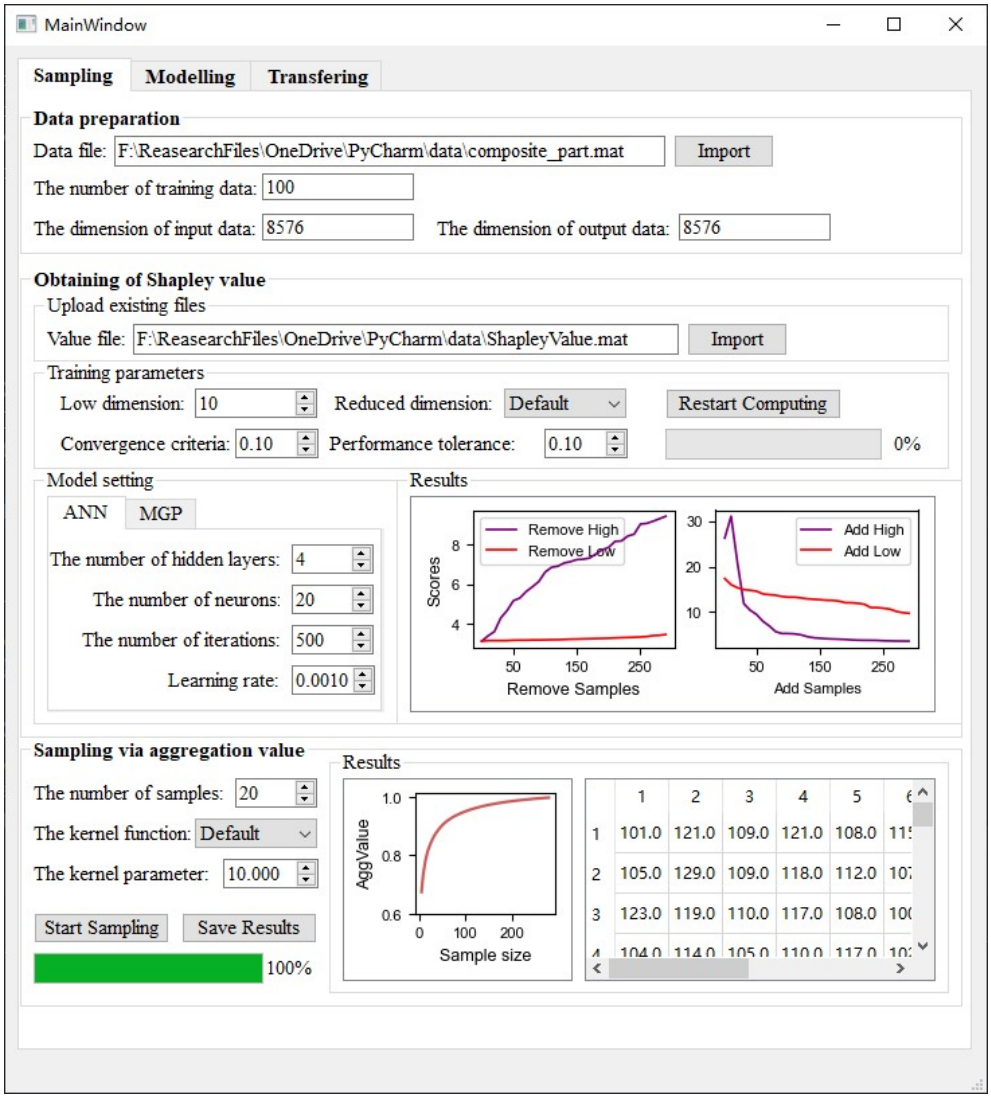


Figure 5.3: The sampling module of the developed system.

5.1.4 The neural operator training module

The neural operator training module is developed based on Python and the deep learning library Pytorch. Due to the scarcity of high-fidelity simulation data, this module will build a deformation field prediction model based on low-fidelity simulation data. The trained model is defined as the source model, which would be the input for the subsequent transfer learning. The software interface of this module is shown in Fig .5.4. Firstly, the low-fidelity simulation data and the mesh file (.stl or .vtk) of the part are im-

ported. This module would solve the physics-based basis, including the POD basis of the low-fidelity simulation data and the Laplacian basis of the geometry. After setting the LNO training parameters, clicking the button 'Start Training' would automatically call the two sets of basis and train the neural operator model. The final training performance of the model and the loss function convergence would be displayed in the software after training. Users can select any samples manually from the test data to display the prediction deformation field and the corresponding prediction errors.

After training the source model based on the low-fidelity simulation data, the model architecture and model parameters are stored in .pkl format, which would be used as input information for the subsequent transfer learning module.

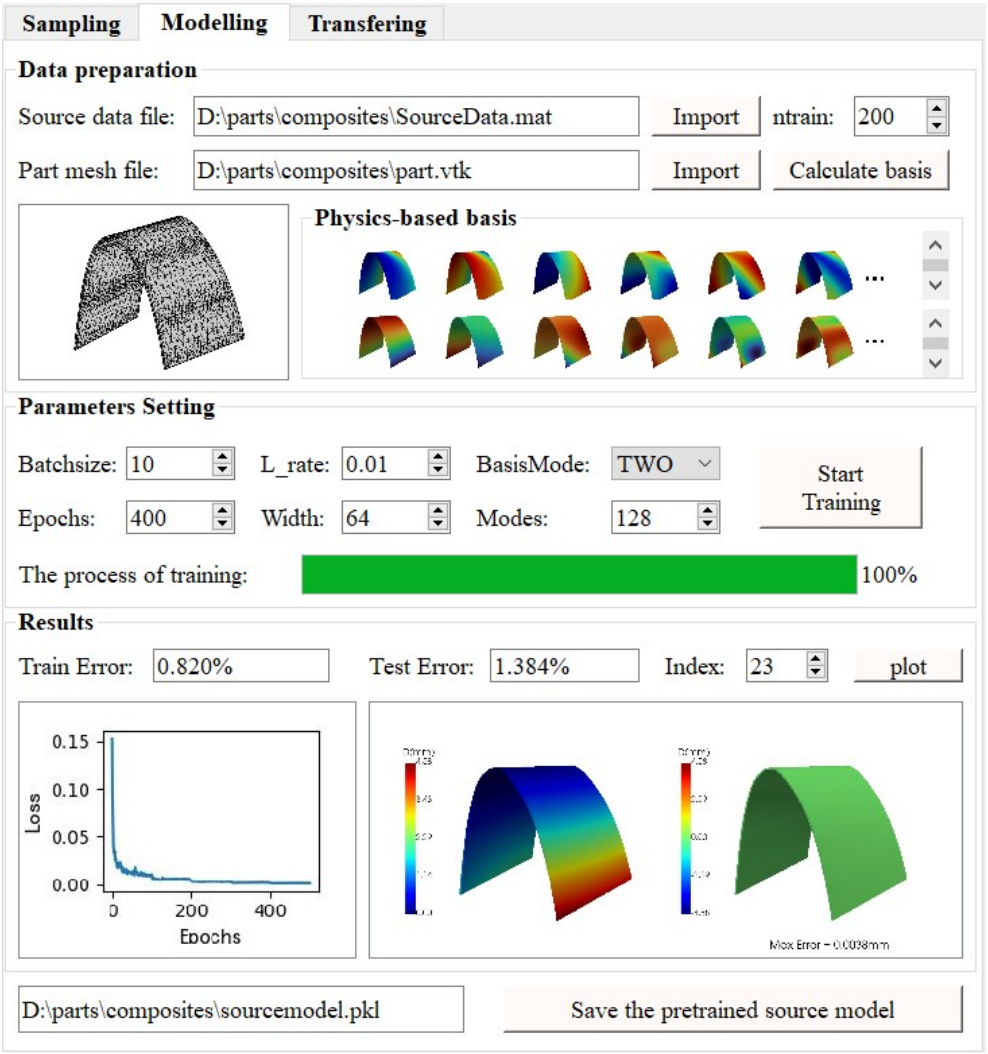


Figure 5.4: The LNO modelling module of the developed system.

5.1.5 The transfer learning module

The transfer learning module was developed based on Pytorch to transfer the parameters of the source model trained from low-fidelity simulation data to high-fidelity data.

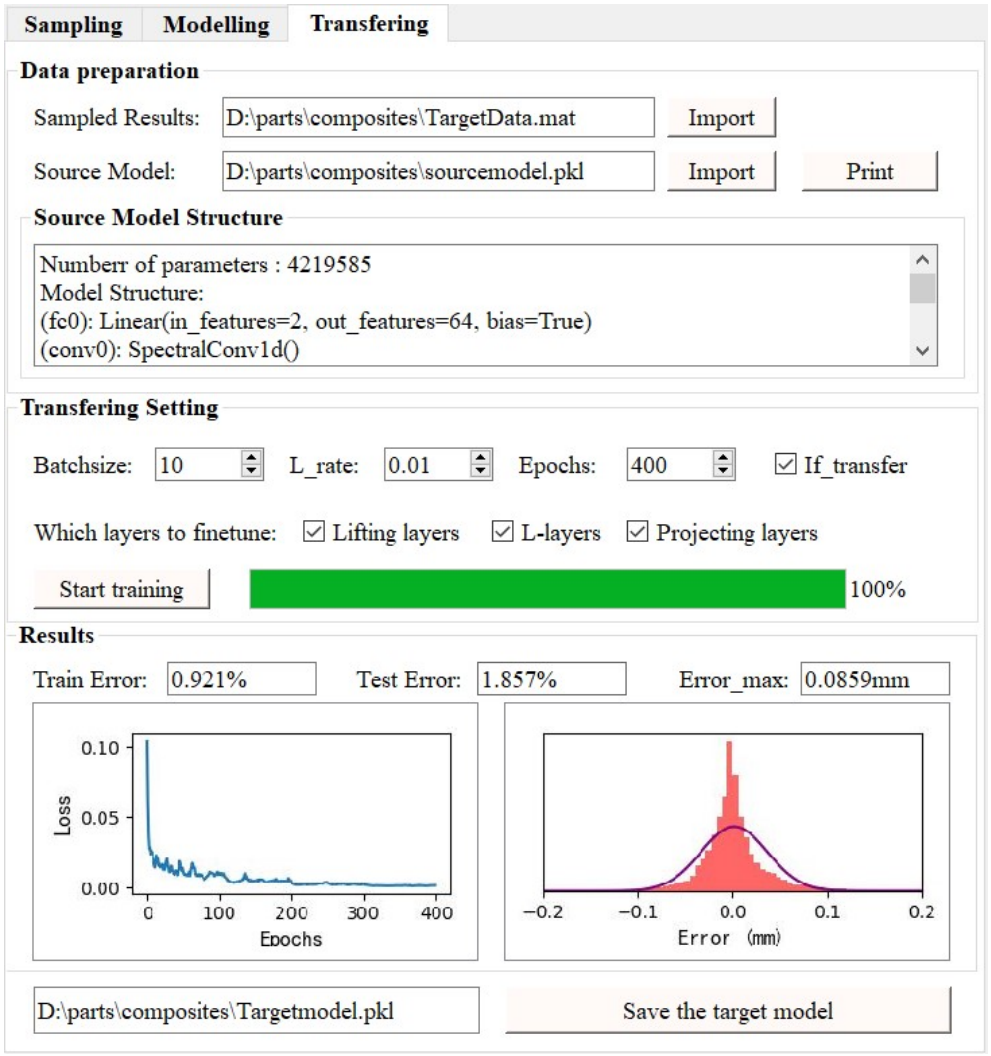


Figure 5.5: The transfer learning module of the developed system.

The software interface of the transfer learning module is shown in Fig.5.5. The proposed CEOD-based loss function was adopted to train the transfer learning model parameters. This module first imports the target data, i.e., the high-fidelity simulation data obtained from the sampling module, and imports the pre-trained source model file in .pkl format. The 'Source Model Structure' function could display the network structure and detailed dimensions of each layer in the source model. Then users could manually select which parts of the model to finetune. The default setting would finetune all the parameters. After clicking 'Start training', the transfer learning model would be trained based on the selected parameters and to-be-updated layers. The final trained model would be saved in .pkl format. The model obtained from this module could be used to predict the deformation field and optimise the temperature field for CFRP curing. For a given input temperature field data, the trained model could perform forward propagation to directly calculate the corresponding deformation field.

5.2 Validation of the developed data-driven curing deformation prediction system

This Section will describe the validation of the effectiveness of the developed framework on a complex composite part. The comparison with the existing data-driven modelling methods shows that the developed framework could predict the deformation accurately with much less high-fidelity simulation data.

5.2.1 The experimental CFRP part

The CFRP workpiece used for verification is shown in Fig. 5.6a. This workpiece is a complex closed revolving structure formed by multiple curved surfaces (Fig. 5.6b), which would deform significantly after high-temperature curing. The curing process is zoned self-resistance electric heating, where the internal and external surfaces of the workpiece are divided into 20 areas according to the radius of curvature for independent temperature control. The maximum deformation of the part with the un-optimised uniform temperature field is more than 4 mm, which cannot meet the actual requirement for further assembly. Optimisation of the temperature field iteratively based on deformation simulation results is an effective means for curing deformation control.

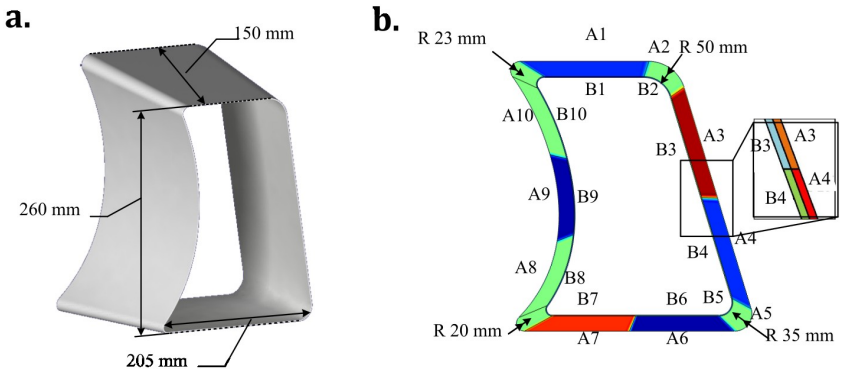


Figure 5.6: The CFRP part for system validation.

The widely used simulation model in the CFRP curing deformation modelling field is Path-dependent (PD) intrinsic structure model, which has high prediction accuracy but requires expensive simulation costs. It would take **7050s**, i.e., about 2 hours, to simulate the curing deformation for one temperature field. Classical GA optimisation requires up to 8000 iterations, which means more than 60 days for simulation. Establishing a data-driven model for curing deformation prediction can significantly reduce efforts of data simulation. As introduced in Chapter 4, the deformation prediction problem can be defined as the mapping from the temperature field $a(x, y, z)$ to the deformation field $u(x, y, z)$ on the 3D geometric part. According to the deformation prediction evaluation criteria provided by the engineers of a collaborating company, the maximum prediction error of the deformation field predicted by the data-driven model should be less than 5%

of the maximum deformation, which means $4mm * 5\% = 0.2mm$. Therefore, the following experiments focus on achieving a maximum prediction error of less than $0.2mm$ with as less high-fidelity simulation data as possible.

In this case, the multi-step stable Rapid Prediction Model for curing deformation simulation proposed by Liu et al. [9] was selected as the low-fidelity simulation model. The CFRP workpiece was divided into a geometric mesh composed of 8576 nodes in Comsol. The Rapid Prediction Model can simulate the curing deformation in only **65s**. Although the accuracy of the Rapid Prediction Model is not as good as the high-fidelity Path-dependent model, it can provide auxiliary information for subsequent high-fidelity data generation and model training. In this case, 300 groups of low-fidelity simulations are generated by Rapid Prediction Model first. A small set of high-value temperature field data are determined by the aggregated value sampling module, and the corresponding deformation fields are then simulated by the Path-dependent model to construct the high-fidelity simulation data.

5.2.2 The experimental settings

This research used the classical operator model DeepOnet as the reference to verify the deformation prediction performance of the developed system under the data scarcity scenario. The detailed settings of each comparison method are introduced as below:

DeepOnet: Training deep-operator-network directly trained with randomly selected high-fidelity simulation data as the reference. The network architecture was designed based on the basic structure proposed by Lulu et al. [220], which includes a fully connected backbone network of 250*250*100 and a fully connected branch network of 128*128 *128*100 with hidden layers. The model was trained with Adam optimiser with a learning rate of 0.001, a batch size of 100, and 10000 iterations. The training data includes randomly generated sets of 20/30/50/100/200 temperature fields and their corresponding high-fidelity simulated deformation fields.

LNO+Sampling: Training low-dimensional neural operator model with the sampled high-fidelity simulation data. The LNO model has 4 L-layers. Considering the complexity of the 3d deformation field, the number of input bases was set to 128 to improve the expressiveness of the model, i.e., $k_l = 128$, $d_v = 64$. The LNO model was trained with Adam optimiser with a learning rate of 0.001, a batch size of 10, and 10000 iterations. The training data are datasets with 20/30/50/100/200 high-fidelity simulation data sampled by the proposed sampling method.

LNO+Sampling+TL: Training the LNO model based on the low-fidelity simulation data and then transferring the model to the sampled high-fidelity simulation data. LNO+Sampling+TL is the complete scheme in the framework shown in Fig. 5.2. The transfer learning part utilises the proposed conditional kernel embedding loss function to update the parameters of the source model with 300 iterations.

5.2.3 Results and analysis

To verify the prediction performance of each method in the data scarcity scenario, each model is first trained with only 20 high-fidelity training data. Fig .5.7 shows one prediction result of LNO+Sampling+TL on the test set. Fig .5.7 a-c are the input temperature field, the ground-truth output deformation field and the predicted deformation field, respectively. It can be seen that the predicted deformation field is very close to the reference value.

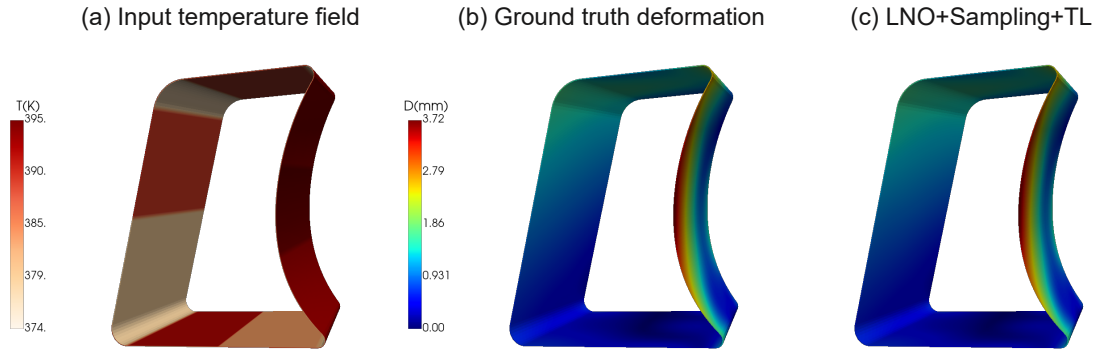


Figure 5.7: The deformation prediction result with 20 high-fidelity training data.

Fig .5.8 provides the deformation field prediction errors of the four methods for the above-mentioned sample. The four figures share a consistent colour scale. Fig .5.8 a is the direct prediction result of the source model trained with only low-fidelity data. The maximum error is as high as 0.976mm, which means that the accuracy of the low-fidelity simulation data cannot meet the demand of engineering applications. Fig .5.8 b shows the deformation field prediction error of DeepOnet, where the maximum error is 0.44mm. Fig .5.8 c is the prediction error of LNO+Sampling. The maximum error is reduced to 0.239mm, while the right area of the part still has a significant error. Fig .5.8 d shows the prediction error of LNO+Sampling+TL. It can be observed that the whole part is 'light green', and the maximum error is only 0.089mm.

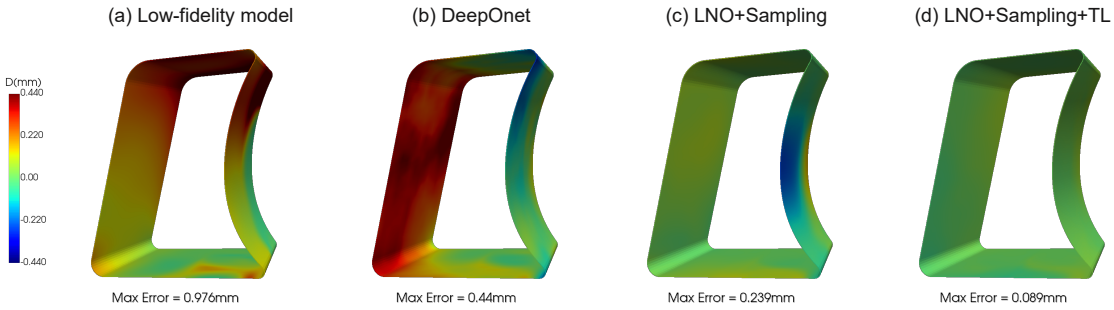


Figure 5.8: The deformation prediction errors of different methods with 20 high-fidelity training data.

Fig .5.9 shows the performance of the three methods with different sample sizes. The test set consists of 100 high-fidelity simulation data. The vertical axis of Fig .5.9 is E_{max}, the average value of the maximum error on the 100 test samples. The horizontal

axis is the number of high-fidelity samples, namely 20, 30, 50, 100 and 200. All methods achieve smaller E_{max} with more samples, while LNO+Sampling+TL has the lowest errors, followed by LNO+Sampling, and DeepOnet has the biggest errors for all sample sizes.

The difference between LNO+Sampling+TL and LNO+Sampling is larger in the case of the small number of samples, including 20 and 30. It means that transfer learning can significantly improve model performance in data scarcity situations. For the 200-sample scenario, the maximum errors of LNO+Sampling and LNO+Sampling+TL have similar performance, only 0.037mm and 0.036mm, which is **more than 80% error reduction** compared to $E_{max} = 0.197mm$ of DeepOnet.

The actual requirement of this workpiece provided by the collaborating company is that the maximum prediction error of the deformation field should be less than 0.2mm. Therefore, a more practical indicator is the number of high-fidelity samples required to achieve the given accuracy requirement. The performance requirement, $E_{max} = 0.2mm$, is marked with a pink line in Fig .5.9. DeepOnet has $E_{max} = 0.197mm$ with 200 high-fidelity samples, while LNO+Sampling can provide a E_{max} below 0.2mm with only 50 high-fidelity samples. In contrast, LNO+Sampling+TL can achieve $E_{max} = 0.186mm$ with only 20 high-fidelity samples, which means that **the prediction accuracy requirement is satisfied with one-tenth of the sample size of DeepOnet**.

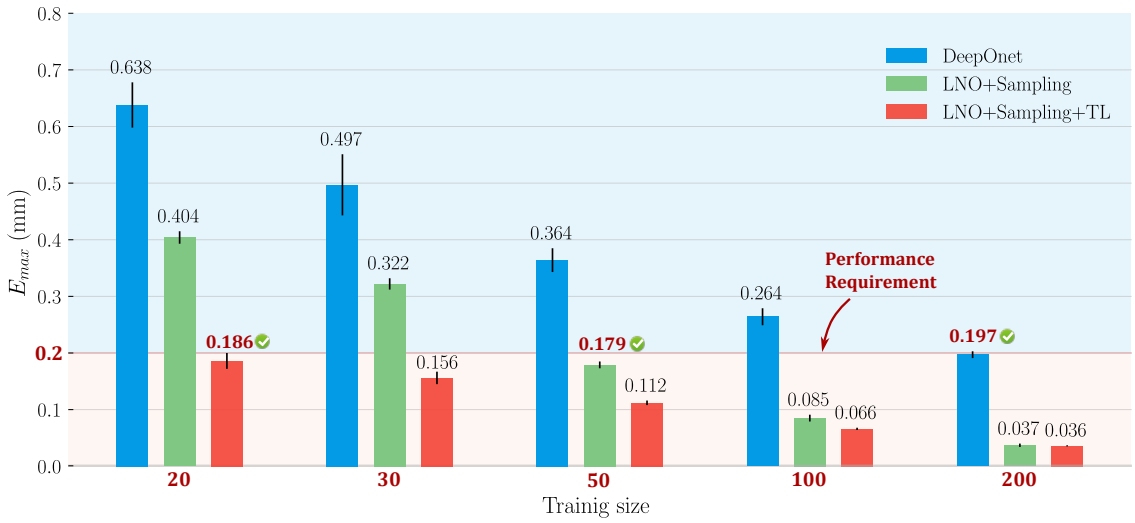


Figure 5.9: The performance comparison under data scarcity scenarios.

The above analysis shows that the system can significantly improve the accuracy of model prediction with same number of samples. Meanwhile, the system can significantly reduce the required number of samples for a given prediction accuracy requirement.

5.3 Summary

To validate the proposed methods described in the previous Chapters, this Chapter described the data-driven deformation prediction system for composite manufacturing and introduced the system framework and functionality. The developed system was validated on a complex CFRP workpiece. The experimental results show that the system can reduce the curing deformation prediction error by 80% compared with existing methods when the same number of training data was given. Meanwhile, for the given prediction accuracy requirements, the developed system reduced the required number of training data by 90% compared with the existing methods.

Chapter 6

Conclusions and further work

Après la pluie, le beau temps.

– Comtesse de Ségur

Data-driven smart manufacturing has demonstrated tremendous potential and drawn increasing research attention. Due to the expensive efforts of labelled data collection, establishing data-driven models with limited labelled data is an inevitable trend and a challenge for developing smart manufacturing. This thesis proposed a new framework for data-driven predictive manufacturing modelling under data-scarcity scenarios. This Chapter will summarise the main contributions of the reported research and provide future research perspectives.

6.1 Summary of contributions

The most important effort of this research was **exploiting manufacturing data generating process to improve the subsequent data modelling process**. For manufacturing predictive modelling problems under limited labelled data, the upper limit performance for data driven models was restricted by the information contained in the dataset. Previous researchers focused on how to train data-driven models based on given datasets, which overlooked the potential of actively leveraging the data-generating process. This research proposed a new framework and developed several methods for manufacturing predictive modelling under data scarcity scenarios. The main contributions are summarised below:

- A comprehensive literature review of data-driven manufacturing was carried out. The basic concepts related to smart manufacturing and recently developed frameworks about data-driven manufacturing have been thoroughly analysed. As a result, typical manufacturing processes and widely-used machine learning methods were categorised based on different characteristics. The literature review confirmed that data labelling of the manufacturing process was expensive and time-consuming both computationally and experimentally. Several advanced modelling techniques for data-scarcity scenarios were reviewed in depth and two research gaps were identified, i.e., **(1) passive data generation and collection**, and **(2) insufficient modelling information**, which not only supported the framework of this research but also provided direction for the development of data-driven manufacturing technologies in industry.
- For research gap 1 (passive data generation and collection), this research proposed a novel aggregation-value-based sampling method to actively select optimal labelled data for data-driven manufacturing applications. A new concept, aggregation value, was proposed to describe the contribution of samples to the performance of data-driven models. The value function was first established based on the auxiliary data or physics priors, then the optimal dataset could be obtained by greedily maximising the aggregation value. Experiments on several manufacturing cases demonstrated that the proposed method could provide optimised samples compared with existing representativeness based and value-based sampling methods, thus could potentially reduce the labelling efforts for manufacturing predictive modelling problems.
- For research gap 2 (insufficient modelling information), this research focused on transfer learning to leverage the transferable knowledge from similar manufacturing configurations so as to compensate for the insufficient modelling information of the direct

labelled data. A structured conditional distribution adaptation method was proposed to adapt the distribution difference between different tasks to extract the transferable knowledge. The validation of the proposed transfer learning method was conducted in various manufacturing problems including predicting tool top dynamics, multi-sensor measurement, and tool wear. The experimental results conclusively demonstrated that leveraging auxiliary data, when direct labelled data was insufficient, could improve the modelling performance.

- This research further proposed a physics-guided low-dimensional neural operator for predicting high-dimensional part property fields. By incorporating physics priors into the neural network structure, this approach could enhance the learning capabilities of neural network and reduces the need for labelled data. The high-dimensional mapping of the part property field could be transformed into a simpler low-dimensional mapping within the physics-based domain. This novel model efficiently represents complex geometries using a limited number of basis, resulting in a substantial reduction in parameter complexity for data-driven models. The experimental analysis, focusing on predicting 3D composite deformation fields, demonstrated that the proposed model could deliver accurate predictions even with a smaller amount of labelled data compared with existing deep learning methods.

- Finally, this research developed an integrated data-driven curing deformation prediction system based on the above proposed methods of sampling, transfer learning and physics-guided low-dimensional neural operator. The developed system was validated using a complex CFRP workpiece. The experimental results demonstrated that the system could achieve an 80% reduction in prediction error for curing deformation compared to existing techniques when provided with the same amount of training data. Additionally, when considering the desired prediction accuracy, the developed system significantly reduced the required training data by 90% compared to the existing method.

6.2 Future research perspectives

Data-driven manufacturing predictive modelling under data-scarcity scenarios is an important challenge for smart manufacturing. This thesis reported a proposed modelling framework that incorporated sampling, knowledge transfer, and data-physics combinations. However, this was only the beginning. There still exist a series of topics that deserve in-depth study:

- **Process optimisation**: The purpose of manufacturing predictive modelling is the subsequent process analysis and optimisation. Previous surrogate-model-based optimisation methods mainly considered prediction models established from data. When direct labelled data, auxiliary data, and physical priors are all available simultaneously, the optimisation problem may have more constraints and more solutions. Therefore, a worthwhile research topic is how to integrate multiple information in manufacturing predictive modelling to establish a process optimisation model.

- **The connection with digital twin**: This research focused on the establishment of predictive models where the data involved can be monitoring data, simulation data or experimental data. The digital twin is more concerned with the bidirectional interaction between the virtual model and the actual model, so it is an interesting direction to combine the data-driven prediction model of simulation data and the prediction model of experimental data to build a digital twin model.
- **Integration of multiple physics priors**: The method described in Chapter 4 utilised simple physics priors to design the network architecture, thereby enhancing the feature extraction capabilities of neural networks. In practical manufacturing processes, there are often multiple different physical priors, including strong priors from formulas and empirical weak priors. Therefore, how to integrate multiple priors to guide both network architecture and loss function design deserves further research.

Bibliography

[1] Tan Jianrong et al. "Research on Key Technical Approaches for the Transition from Digital Manufacturing to Intelligent Manufacturing". In: *Strategic Study of Chinese Academy of Engineering* 19.3 (2017), pp. 34–44.

[2] Meng Zhang et al. "Digital twin data: methods and key technologies". In: *Digital Twin* 1.2 (2022), p. 2.

[3] Ke Xu et al. "Advanced data collection and analysis in data-driven manufacturing process". In: *Chinese Journal of Mechanical Engineering* 33.1 (2020), pp. 1–21.

[4] Peter O'Donovan et al. "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities". In: *Journal of Big Data* 2.1 (2015), pp. 1–26.

[5] Fei Tao et al. "Digital twin towards smart manufacturing and industry 4.0". In: *Journal of manufacturing systems* 58 (2021), pp. 1–2.

[6] Andrew Kusiak. "Smart manufacturing". In: *International Journal of Production Research* 56.1-2 (2018), pp. 508–517.

[7] Chao Shang and Fengqi You. "Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era". In: *Engineering* 5.6 (2019), pp. 1010–1016.

[8] Chunquan Li, Yaqiong Chen, and Yuling Shang. "A review of industrial big data for decision making in intelligent manufacturing". In: *Engineering Science and Technology, an International Journal* 29 (2022), p. 101021.

[9] Yingxiang Shen et al. "Self-resistance electric heating of shaped CFRP laminates: temperature distribution optimization and validation". In: *The International Journal of Advanced Manufacturing Technology* 121.3-4 (2022), pp. 1755–1768.

[10] Pascal Hubert, G Fernlund, and A Poursartip. *Manufacturing techniques for polymer matrix composites (PMCs)*. Cambridge, MA: Woodhead Publishing, 2012.

[11] Giacomo Struzziero, Julie JE Teuwen, and Alexandros A Skordos. "Numerical optimisation of thermoset composites manufacturing processes: A review". In: *Composites Part A: Applied Science and Manufacturing* 124 (2019), p. 105499.

[12] Stefan Hiemer and Stefano Zapperi. "From mechanism-based to data-driven approaches in materials science". In: *Materials Theory* 5.1 (2021), pp. 1–9.

[13] Ye Ding et al. "A full-discretization method for prediction of milling stability". In: *International Journal of Machine Tools and Manufacture* 50.5 (2010), pp. 502–509.

[14] Sina Amini Niaki et al. "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture". In: *Comput Methods Appl Mech Eng* 384 (2021), p. 113959.

[15] Yinghao Cheng et al. “Mechanism-based Structured Deep Neural Network for Cutting Force Forecasting using CNC Inherent Monitoring Signals”. In: *IEEE/ASME Transactions on Mechatronics* (2021).

[16] Zhuo Wang et al. “Data-driven modeling of process, structure and property in additive manufacturing: A review and future directions”. In: *Journal of Manufacturing Processes* 77 (2022), pp. 13–31.

[17] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Berlin: Springer, 2006.

[18] Keith D Humfeld et al. “A machine learning framework for real-time inverse modeling and multi-objective process optimization of composites for active manufacturing control”. In: *Composites Part B: Engineering* 223 (2021), p. 109150.

[19] Martin Postel, Bircan Bugdayci, and Konrad Wegener. “Ensemble transfer learning for refining stability predictions in milling using experimental stability states”. In: *The International Journal of Advanced Manufacturing Technology* 107.9 (2020), pp. 4123–4139.

[20] Zuowei Zhu et al. “Machine learning in tolerancing for additive manufacturing”. In: *CIRP annals* 67.1 (2018), pp. 157–160.

[21] Ye Yuan et al. “A general end-to-end diagnosis framework for manufacturing systems”. In: *Natl Sci Rev* 7.2 (2020), pp. 418–429.

[22] Tianxiang Kong et al. “Data construction method for the applications of workshop digital twin system”. In: *Journal of Manufacturing Systems* 58 (2021), pp. 323–328.

[23] Charles E Harris, James H Starnes Jr, and Mark J Shuart. “Design and manufacturing of aerospace composite structures, state-of-the-art assessment”. In: *J Aircr* 39.4 (2002), pp. 545–560.

[24] Gengxiang Chen, Yingguang Li, and Xu Liu. “Pose-dependent tool tip dynamics prediction using transfer learning”. In: *International Journal of Machine Tools and Manufacture* 137 (2019), pp. 30–41.

[25] Toni Cvitanic, Vinh Nguyen, and Shreyes N Melkote. “Pose optimization in robotic machining using static and dynamic stiffness models”. In: *Robotics and Computer-Integrated Manufacturing* 66 (2020), p. 101992.

[26] Milad Ramezankhani et al. “Making costly manufacturing smart with transfer learning under limited data: A case study on composites autoclave processing”. In: *Journal of Manufacturing Systems* 59 (2021), pp. 345–354.

[27] Milad Ramezankhani et al. “A Data-driven Multi-fidelity Physics-informed Learning Framework for Smart Manufacturing: A Composites Processing Case Study”. In: *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE. 2022, pp. 01–07.

[28] David Gonzalez-Jimenez et al. “Data-driven fault diagnosis for electric drives: A review”. In: *Sensors* 21.12 (2021), p. 4024.

[29] Monika Klippert et al. “Industrie 4.0–An empirical and literature-based study how product development is influenced by the digital transformation”. In: *Procedia CIRP* 91 (2020), pp. 80–86.

[30] Fei Tao et al. “Data-driven smart manufacturing”. In: *Journal of Manufacturing Systems* 48 (2018), pp. 157–169.

[31] Yuqian Lu, Xun Xu, and Lihui Wang. “Smart manufacturing process and system automation–a critical review of the standards and envisioned scenarios”. In: *Journal of Manufacturing Systems* 56 (2020), pp. 312–325.

[32] E Wallace and F Riddick. “Panel on enabling smart manufacturing”. In: *State College, USA* (2013).

[33] Maija Breque, Lars De Nul, and Athanasios Petridis. “Industry 5.0: Towards more sustainable, resilient and human-centric industry”. In: *Res. Innov., Eur. Commission* (2021).

[34] Andrew Kusiak. “Fundamentals of smart manufacturing: A multi-thread perspective”. In: *Annual Reviews in Control* 47 (2019), pp. 214–220.

[35] Baicun Wang et al. “Toward human-centric smart manufacturing: A human-cyber-physical systems (HCPS) perspective”. In: *Journal of Manufacturing Systems* 63 (2022), pp. 471–490.

[36] Linn D Evjemo et al. “Trends in smart manufacturing: Role of humans and industrial robots in smart factories”. In: *Current Robotics Reports* 1 (2020), pp. 35–41.

[37] Ying Cheng et al. “Cyber-physical integration for moving digital factories forward towards smart manufacturing: a survey”. In: *The International Journal of Advanced Manufacturing Technology* 97 (2018), pp. 1209–1221.

[38] Yong Lin, Petros Ieromonachou, and Wenxian Sun. “Smart manufacturing and supply chain management”. In: *2016 International Conference on Logistics, Informatics and Service Sciences (LISS)*. IEEE. 2016, pp. 1–5.

[39] Nilufer Tuptuk and Stephen Hailes. “Security of smart manufacturing systems”. In: *Journal of manufacturing systems* 47 (2018), pp. 93–106.

[40] Jinjiang Wang et al. “Deep learning for smart manufacturing: Methods and applications”. In: *Journal of manufacturing systems* 48 (2018), pp. 144–156.

[41] Arfan Majeed et al. “A big data-driven framework for sustainable and smart additive manufacturing”. In: *Robotics and Computer-Integrated Manufacturing* 67 (2021), p. 102026.

[42] Junliang Wang et al. “Big data analytics for intelligent manufacturing systems: A review”. In: *Journal of Manufacturing Systems* (2021).

[43] Li Li et al. "Meta-learning based industrial intelligence of feature nearest algorithm selection framework for classification problems". In: *Journal of Manufacturing Systems* 62 (2022), pp. 767–776.

[44] Leilani H Gilpin et al. "Explaining explanations: An overview of interpretability of machine learning". In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.

[45] Benjamin Moseley. "Physics-informed machine learning: from concepts to real-world applications". PhD thesis. University of Oxford, 2022.

[46] Jinjiang Wang et al. "Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability". In: *Journal of Manufacturing Systems* 63 (2022), pp. 381–391.

[47] Mojtaba Mozaffar et al. "Mechanistic artificial intelligence (mechanistic-AI) for modeling, design, and control of advanced manufacturing processes: Current state and perspectives". In: *Journal of Materials Processing Technology* (2021), p. 117485.

[48] Noel P Greis et al. "Physics-guided machine learning for self-aware machining". In: *2020 AAAI Spring Symposium on AI and Manufacturing*. 2020.

[49] Sourav Saha et al. "Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113452.

[50] Feng Zhou et al. "Optimization of numerical control program and machining simulation based on VERICUT". In: *Journal of Shanghai Jiaotong University (Science)* 24 (2019), pp. 763–768.

[51] Yu-Yue Yu et al. "Online stability boundary drifting prediction in milling process: An incremental learning approach". In: *Mechanical Systems and Signal Processing* 173 (2022), p. 109062.

[52] Jinjiang Wang et al. "Physics guided neural network for machining tool wear prediction". In: *Journal of Manufacturing Systems* 57 (2020), pp. 298–310.

[53] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[54] Trevor Hastie et al. "Overview of supervised learning". In: *The elements of statistical learning: Data mining, inference, and prediction* (2009), pp. 9–41.

[55] Han Ding et al. "State of AI-based monitoring in smart manufacturing and introduction to focused section". In: *IEEE ASME Trans Mechatron* 25.5 (2020), pp. 2143–2154.

[56] Hongrui Cao, Xingwu Zhang, and Xuefeng Chen. "The concept and progress of intelligent spindles: a review". In: *International Journal of Machine Tools and Manufacture* 112 (2017), pp. 21–52.

[57] Ming Ge, Yangsheng Xu, and Ruxu Du. "An Intelligent Online Monitoring and Diagnostic System for Manufacturing Automation". In: *IEEE Transactions on Automation Science and Engineering* 5.1 (2008), pp. 127–139. DOI: 10.1109/TASE.2006.886833.

[58] Chunsheng Zhao et al. "Development and application prospects of piezoelectric precision driving technology". In: *Frontiers of Mechanical Engineering in China* 3 (2008), pp. 119–132.

[59] Muhammad Rizal et al. "An embedded multi-sensor system on the rotating dynamometer for real-time condition monitoring in milling". In: *The International Journal of Advanced Manufacturing Technology* 95 (2018), pp. 811–823.

[60] Dragan Aleksendrić, Pierpaolo Carlone, and Velimir Ćirović. "Optimization of the temperature-time curve for the curing process of thermoset matrix composites". In: *Applied Composite Materials* 23 (2016), pp. 1047–1063.

[61] Ce Zhang et al. "Review of curing deformation control methods for carbon fiber reinforced resin composites". In: *Polymer Composites* 43.6 (2022), pp. 3350–3370.

[62] K Kerrigan et al. "An integrated telemetric thermocouple sensor for process monitoring of CFRP milling operations". In: *Procedia CIRP* 1 (2012), pp. 449–454.

[63] Ramzyzan Ramly, Wahyu Kuntjoro, and Mohd Kamil Abd Rahman. "Using embedded fiber Bragg grating (FBG) sensors in smart aircraft structure materials". In: *Procedia Engineering* 41 (2012), pp. 600–606.

[64] Jiaqi Hua et al. "A zero-shot prediction method based on causal inference under non-stationary manufacturing environments for complex manufacturing systems". In: *Robotics and Computer-Integrated Manufacturing* 77 (2022), p. 102356.

[65] Sarah K Everton et al. "Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing". In: *Materials & Design* 95 (2016), pp. 431–445.

[66] Li Cai and Yangyong Zhu. "The challenges of data quality and data quality assessment in the big data era". In: *Data science journal* 14 (2015).

[67] Orkun Özşahin, Erhan Budak, and H Nevzat Özgüven. "In-process tool point FRF identification under operational conditions using inverse stability solution". In: *International Journal of Machine Tools and Manufacture* 89 (2015), pp. 64–73.

[68] Sif Eddine Sadaoui, Charyar Mehdi-Souzani, and Claire Lartigue. "Multisensor data processing in dimensional metrology for collaborative measurement of a laser plane sensor combined to a touch probe". In: *Measurement* 188 (2022), p. 110395.

[69] Na Cai et al. "Freeform machining feature recognition with manufacturability analysis". In: *Procedia CIRP* 72 (2018), pp. 1475–1480.

[70] Ivo Utke et al. "Mechanical properties of 3D nanostructures obtained by focused electron/ion beam-induced deposition: A review". In: *Micromachines* 11.4 (2020), p. 397.

[71] Peng Guo et al. “Autonomous Profile Tracking for Multiaxis Ultrasonic Measurement of Deformed Surface in Mirror Milling”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–13.

[72] SP Leo Kumar. “Measurement and uncertainty analysis of surface roughness and material removal rate in micro turning operation and process parameters optimization”. In: *Measurement* 140 (2019), pp. 538–547.

[73] Marc-Antoine de Pastre, Yann Quinsat, and Claire Lartigue. “Shape defect analysis from volumetric data-Application to lattice struts in additive manufacturing”. In: *Precision Engineering* 76 (2022), pp. 12–28.

[74] G Chen et al. “A comparative study of Ti-6Al-4V powders for additive manufacturing by gas atomization, plasma rotating electrode process and plasma atomization”. In: *Powder technology* 333 (2018), pp. 38–46.

[75] Xuan Bien Duong et al. “Optimize the feed rate and determine the joints torque for industrial welding robot TA 1400 based on kinematics and dynamics modeling”. In: *International Journal of Mechanical Engineering and Robotics Research* 9.9 (2020), pp. 1335–1340.

[76] Fan Chen et al. “Contact force control and vibration suppression in robotic polishing with a smart end effector”. In: *Robotics and Computer-Integrated Manufacturing* 57 (2019), pp. 391–403.

[77] Felix Finkeldey et al. “Learning-based prediction of pose-dependent dynamics”. In: *Journal of Manufacturing and Materials Processing* 4.3 (2020), p. 85.

[78] LL Alhadeff et al. “Protocol for tool wear measurement in micro-milling”. In: *Wear* 420 (2019), pp. 54–67.

[79] Majda Paweł and Powałka Bartosz. “Rapid method to determine accuracy and repeatability of positioning of numerically controlled axes”. In: *International Journal of Machine Tools and Manufacture* 137 (2019), pp. 1–12.

[80] Yingguang Li et al. “A novel method for accurately monitoring and predicting tool wear under varying cutting conditions based on meta-learning”. In: *CIRP Annals* 68.1 (2019), pp. 487–490.

[81] J Munoa et al. “Chatter suppression techniques in metal cutting”. In: *CIRP annals* 65.2 (2016), pp. 785–808.

[82] Mohit Law, A Srikantha Phani, and Yusuf Altintas. “Position-dependent multibody dynamic modeling of machine tools based on improved reduced order models”. In: *Journal of manufacturing science and engineering* 135.2 (2013).

[83] Binxun Li et al. “Simulated and experimental analysis on serrated chip formation for hard milling process”. In: *Journal of Manufacturing Processes* 44 (2019), pp. 337–348.

[84] Fabian Neugebauer et al. “Multi scale FEM simulation for distortion calculation in additive manufacturing of hardening stainless steel”. In: *international workshop on thermal forming and welding distortion, Bremen, Germany*. 2014.

[85] Navid Zobeiry and Anoush Poursartip. “Theory-guided machine learning for process simulation of advanced composites”. In: *arXiv:2103.16010* ().

[86] Julius Pfrommer et al. “Optimisation of manufacturing process parameters using deep neural networks as surrogate models”. In: *Procedia CiRP* 72 (2018), pp. 426–431.

[87] Andrés Sio Sever et al. “Use of the phenomenon of acoustic emission for real-time monitoring of milling processes”. In: *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*. Vol. 259. 8. Institute of Noise Control Engineering. 2019, pp. 1913–1921.

[88] Wanqun Chen et al. “Finite element simulation and experimental investigation on cutting mechanism in vibration-assisted micro-milling”. In: *The International Journal of Advanced Manufacturing Technology* 105.11 (2019), pp. 4539–4549.

[89] Ryosuke Matsuzaki, Tadahiro Kobara, and Ryota Yokoyama. “Efficient estimation of thermal conductivity distribution during curing of thermoset composites”. In: *Advanced Composite Materials* 30.sup2 (2021), pp. 34–49.

[90] Zhiwei Zhao et al. “A subsequent-machining-deformation prediction method based on the latent field estimation using deformation force”. In: *Journal of Manufacturing Systems* 63 (2022), pp. 224–237.

[91] M Alauddin, MA El Baradie, and MSJ Hashmi. “Computer-aided analysis of a surface-roughness model for end milling”. In: *Journal of materials processing technology* 55.2 (1995), pp. 123–127.

[92] George EP Box. *An Introduction to Response Surface Methodology*. Tech. rep. WISCONSIN UNIV MADISON, 1964.

[93] M Alauddin, MA El Baradie, and MSJ Hashmi. “Prediction of tool life in end milling by response surface methodology”. In: *Journal of Materials Processing Technology* 71.3 (1997), pp. 456–465.

[94] Michael Rausch and William H Sanders. “Evaluating the effectiveness of metamodeling in emulating quantitative models”. In: *Quantitative Evaluation of Systems: 18th International Conference, QEST 2021, Paris, France, August 23–27, 2021, Proceedings 18*. Springer. 2021, pp. 127–145.

[95] Nestor V Queipo et al. “Surrogate-based analysis and optimization”. In: *Progress in aerospace sciences* 41.1 (2005), pp. 1–28.

[96] Gustavo Tapia et al. “Gaussian process-based surrogate modeling framework for process planning in laser powder-bed fusion additive manufacturing of 316L stainless steel”. In: *The International Journal of Advanced Manufacturing Technology* 94.9 (2018), pp. 3591–3603.

[97] Oludare Isaac Abiodun et al. “State-of-the-art in artificial neural network applications: A survey”. In: *Heliyon* 4.11 (2018), e00938.

[98] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. “Artificial neural networks: A tutorial”. In: *Computer* 29.3 (1996), pp. 31–44.

[99] Martin Postel et al. “Neural network supported inverse parameter identification for stability predictions in milling”. In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 71–87.

[100] Mark Ebden. “Gaussian processes: A quick introduction”. In: *arXiv preprint arXiv:1505.02965* (2015).

[101] Gengxiang Chen et al. “A kernel transfer learning based multi-sensor surface reconstruction framework for reverse engineering”. In: *Procedia CIRP* 109 (2022), pp. 672–677.

[102] Ming Ge, Yangsheng Xu, and Ruxu Du. “An intelligent online monitoring and diagnostic system for manufacturing automation”. In: *IEEE Transactions on Automation Science and Engineering* 5.1 (2008), pp. 127–139.

[103] Mohsen Soori and Behrooz Arezoo. “Cutting tool wear prediction in machining operations, a review”. In: *Journal of New Technology and Materials* (2022).

[104] Xingjian Li et al. “Delta: Deep learning transfer using feature map with attention for convolutional networks”. In: *arXiv preprint arXiv:1901.09229* (2019).

[105] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. “A review on the long short-term memory model”. In: *Artificial Intelligence Review* 53 (2020), pp. 5929–5955.

[106] Chuang Sun et al. “Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing”. In: *IEEE transactions on industrial informatics* 15.4 (2018), pp. 2416–2425.

[107] M Hossein Rahimi, Hoai Nam Huynh, and Yusuf Altintas. “On-line chatter detection in milling with hybrid machine learning and physics-based model”. In: *CIRP Journal of Manufacturing Science and Technology* 35 (2021), pp. 25–40.

[108] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. “Coresets for data-efficient training of machine learning models”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vienna: ACM, 2020, pp. 6950–6960.

[109] Ehsan Elhamifar, Guillermo Sapiro, and S Shankar Sastry. “Dissimilarity-based sparse subset selection”. In: *IEEE Trans Pattern Anal Mach Intell* 38.11 (2015), pp. 2182–2197.

[110] L Eriksson et al. “Design of experiments”. In: *Principles and Applications, Learn ways AB, Stockholm* (2000).

[111] G Gary Wang and Songqing Shan. “Review of metamodeling techniques in support of engineering design optimization”. In: (2007).

[112] Punit Kumar and Atul Gupta. "Active learning query strategies for classification, regression, and clustering: a survey". In: *Journal of Computer Science and Technology* 35.4 (2020), pp. 913–945.

[113] Andreas Krause and Carlos Guestrin. "Beyond convexity: Submodularity in machine learning". In: *ICML Tutorials* (2008).

[114] Jieji Ren et al. "Generative Model-Driven Sampling Strategy for the High-Efficiency Measurement of Complex Surfaces on Coordinate Measuring Machines". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11.

[115] Sariel Har-Peled and Soham Mazumdar. "On coresets for k-means and k-median clustering". In: *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. 2004, pp. 291–300.

[116] Haitao Liu, Yew-Soon Ong, and Jianfei Cai. "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design". In: *Structural and Multidisciplinary Optimization* 57.1 (2018), pp. 393–416.

[117] Luca Pagani and Paul J Scott. "Curvature based sampling of curves and surfaces". In: *Computer Aided Geometric Design* 59 (2018), pp. 32–48.

[118] Pengzhen Ren et al. "A survey of deep active learning". In: *ACM Computing Surveys (CSUR)* 54.9 (2021), pp. 1–40.

[119] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. "Active learning by querying informative and representative examples". In: *Advances in neural information processing systems* 23 (2010).

[120] Pinar Donmez, Jaime G Carbonell, and Paul N Bennett. "Dual strategy active learning". In: *European Conference on Machine Learning*. Springer. 2007, pp. 116–127.

[121] Yoav Freund et al. "Selective sampling using the query by committee algorithm". In: *Machine learning* 28.2 (1997), pp. 133–168.

[122] Mateo Leco, Thomas McLeay, and Visakan Kadirkamanathan. "A two-step machining and active learning approach for right-first-time robotic countersinking through in-process error compensation and prediction of depth of cuts". In: *Robotics and Computer-Integrated Manufacturing* 77 (2022), p. 102345.

[123] AJ Hughes et al. "On risk-based active learning for structural health monitoring". In: *Mechanical Systems and Signal Processing* 167 (2022), p. 108569.

[124] Giovanna Martinez Arellano and Svetan Ratchev. "Towards an active learning approach to tool condition monitoring with bayesian deep learning". In: (2019).

[125] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[126] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[127] Hanrui Wu, Qingyao Wu, and Michael K Ng. "Knowledge preserving and distribution alignment for heterogeneous domain adaptation". In: *ACM Transactions on Information Systems (TOIS)* 40.1 (2021), pp. 1–29.

[128] Sheng Zhang et al. "Combining cross-modal knowledge transfer and semi-supervised learning for speech emotion recognition". In: *Knowledge-Based Systems* 229 (2021), p. 107340.

[129] Fuzhen Zhuang et al. "A comprehensive survey on transfer learning". In: *Proceedings of the IEEE* 109.1 (2020), pp. 43–76.

[130] Siyu Shao et al. "Highly accurate machine fault diagnosis using deep transfer learning". In: *IEEE Transactions on Industrial Informatics* 15.4 (2018), pp. 2446–2455.

[131] Mohamed Marei, Shirine El Zaatari, and Weidong Li. "Transfer learning enabled convolutional neural networks for estimating health state of cutting tools". In: *Robotics and Computer-Integrated Manufacturing* 71 (2021), p. 102145.

[132] Matthew Staib and Stefanie Jegelka. "Distributionally robust optimization and generalization in kernel methods". In: *Advances in Neural Information Processing Systems* 32 (2019).

[133] Kai Li et al. "A novel adversarial domain adaptation transfer learning method for tool wear state prediction". In: *Knowledge-Based Systems* 254 (2022), p. 109537.

[134] Yan Xu et al. "A digital-twin-assisted fault diagnosis using deep transfer learning". In: *Ieee Access* 7 (2019), pp. 19990–19999.

[135] Eric Tzeng et al. "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014).

[136] Mingsheng Long et al. "Deep transfer learning with joint adaptation networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2208–2217.

[137] Hanrui Wu et al. "Geometric knowledge embedding for unsupervised domain adaptation". In: *Knowledge-Based Systems* 191 (2020), p. 105155.

[138] Hanrui Wu et al. "Heterogeneous domain adaptation by information capturing and distribution matching". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 6364–6376.

[139] Hanrui Wu et al. "Iterative refinement for multi-source visual domain adaptation". In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[140] Chuanqi Tan et al. "A survey on deep transfer learning". In: *International conference on artificial neural networks*. Springer. 2018, pp. 270–279.

[141] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems* 27 (2014).

[142] Emmanuel Moulay, Vincent Léchappé, and Franck Plestan. "Properties of the sign gradient descent algorithms". In: *Information Sciences* 492 (2019), pp. 29–39.

[143] Russell Stewart and Stefano Ermon. "Label-free supervision of neural networks with physics and domain knowledge". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[144] Yucheng Wang et al. "Digital twin enhanced fault prediction for the autoclave with insufficient data". In: *Journal of Manufacturing Systems* 60 (2021), pp. 350–359.

[145] George Em Karniadakis et al. "Physics-informed machine learning". In: *Nat Rev Phys* 3.6 (2021), pp. 422–440.

[146] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. "Hamiltonian neural networks". In: *Advances in Neural Information Processing Systems* 32 (2019).

[147] Michael Lutter, Christian Ritter, and Jan Peters. "Deep lagrangian networks: Using physics as model prior for deep learning". In: *arXiv preprint arXiv:1907.04490* (2019).

[148] Navid Zobeiry and Keith D Humfeld. "A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications". In: *Engineering Applications of Artificial Intelligence* 101 (2021), p. 104232.

[149] Meng Zhang et al. "A physical model and data-driven hybrid prediction method towards quality assurance for composite components". In: *CIRP Annals* 70.1 (2021), pp. 115–118.

[150] Krishnateja Killamsetty et al. "Glister: Generalization based data subset selection for efficient and robust learning". In: *arXiv:2012.10630* ().

[151] Akshay L Chandra et al. "On initial pools for deep active learning". In: *Proceedings of the 35th Advances in Neural Information Processing Systems (NIPS)*. New York, NY: MIT Press, 2021, pp. 14–32.

[152] Krithika Manohar et al. "Predicting shim gaps in aircraft assembly with machine learning and sparse sensing". In: *J Manuf Syst* 48 (2018), pp. 87–95.

[153] Amirata Ghorbani and James Zou. "Data shapley: Equitable valuation of data for machine learning". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. New York, NY: ACM, 2019, pp. 2242–2251.

[154] Pang Wei Koh and Percy Liang. "Understanding black-box predictions via influence functions". In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney: ACM, 2017, pp. 1885–1894.

[155] Amirata Ghorbani, Michael Kim, and James Zou. "A distributional framework for data valuation". In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Vienna: ACM, 2020, pp. 3535–3544.

[156] Yongchan Kwon, Manuel A Rivas, and James Zou. "Efficient computation and analysis of distributional Shapley values". In: *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*. New York, NY: IEEE Information Theory Society, 2021, pp. 793–801.

[157] S Durga et al. "Training Data Subset Selection for Regression with Controlled Generalization Error". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. New York, NY: ACM, 2021, pp. 9202–9212.

[158] Maya Gupta et al. "Diminishing returns shape constraints for interpretability and regularization". In: *Proceedings of the 32th Advances in Neural Information Processing Systems (NIPS)*. Montréal: MIT Press, 2018.

[159] Soumi Das et al. "Finding High-Value Training Data Subset Through Differentiable Convex Programming". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Bilbao: Springer, 2021, pp. 666–681.

[160] Gao Feng et al. "An adaptive sampling method for accurate measurement of aero-engine blades". In: *Measurement* 173 (2021), p. 108531.

[161] N Rama Suri and Yadati Narahari. "Determining the top-k nodes in social networks using the shapley value". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*. 2008, pp. 1509–1512.

[162] Andreas Krause and Daniel Golovin. "Submodular function maximization." In: *Tractability* 3 (2014), pp. 71–104.

[163] Lloyd S Shapley et al. "A value for n-person games". In: (1953).

[164] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: *International conference on machine learning*. PMLR. 2020, pp. 9269–9278.

[165] Alexandre Duval and Fragkiskos D Malliaros. "Graphsvx: Shapley value explanations for graph neural networks". In: *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*. Springer. 2021, pp. 302–318.

[166] Yuichi Motai. "Kernel association for classification and prediction: A survey". In: *IEEE Trans Neural Netw Learn Syst* 26.2 (2014), pp. 208–223.

[167] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31th Advances in Neural Information Processing Systems (NIPS)*. Vol. 30. New York, NY: MIT Press, 2017.

[168] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. "An analysis of approximations for maximizing submodular set functions—I". In: *Math Program* 14.1 (1978), pp. 265–294.

[169] Alex Kriz. *The Cifar 10 Dataset*. https://www.cs.toronto.edu/~kriz/cifar.html (10 September 2022, date last accessed). 2022.

[170] Bearing Data Center. *Case Western Reserve University Seeded Fault Test.* https://engineering.case.edu/bearingdatacenter (10 September 2022, date last accessed).

[171] PHM society. *A PHM society conference data challenge, Tool wear dataset.* https://www.phmsociety.org/competition/phm/10 (10 September 2022, date last accessed).

[172] Iain Ainsworth, Mihailo Ristic, and Djordje Brujic. "CAD-based measurement path planning for free-form shapes using contact probes". In: *Int J Adv Manuf Technol* 16.1 (2000), pp. 23–31.

[173] Gengxiang Chen, Yingguang Li, and Xu Liu. "Transfer Learning Under Conditional Shift Based on Fuzzy Residual". In: *IEEE Trans Cybern* 52.2 (2022), pp. 960–970. DOI: 10.1109/TCYB.2020.2988277.

[174] Jindong Wang et al. "Visual domain adaptation with manifold embedded distribution alignment". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 402–410.

[175] Jingjing Li et al. "Transfer independently together: A generalized framework for domain adaptation". In: *IEEE transactions on cybernetics* 49.6 (2018), pp. 2144–2155.

[176] Dirk Tasche. "Fisher consistency for prior probability shift". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3338–3369.

[177] Joaquin Quinonero-Candela et al. *Dataset shift in machine learning*. Mit Press, 2008.

[178] Jiayuan Huang et al. "Correcting sample selection bias by unlabeled data". In: *Advances in neural information processing systems* 19 (2006).

[179] S Trevis Certo et al. "Sample selection bias and Heckman models in strategic management research". In: *Strategic Management Journal* 37.13 (2016), pp. 2639–2657.

[180] Kun Zhang et al. "Domain adaptation under target and conditional shift". In: *International conference on machine learning*. PMLR. 2013, pp. 819–827.

[181] Mingsheng Long et al. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[182] Xinyang Chen et al. "Representation Subspace Distance for Domain Adaptation Regression." In: *ICML*. 2021, pp. 1749–1759.

[183] Stéphane Lathuilière et al. "A comprehensive analysis of deep regression". In: *IEEE transactions on pattern analysis and machine intelligence* 42.9 (2019), pp. 2065–2081.

[184] Simon S Du et al. "Hypothesis transfer learning via transformation functions". In: *Advances in neural information processing systems* 30 (2017).

[185] Xuezhi Wang, Tzu-Kuo Huang, and Jeff Schneider. "Active transfer learning under model shift". In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1305–1313.

[186] Hua Zuo et al. "Fuzzy regression transfer learning in Takagi–Sugeno fuzzy models". In: *IEEE Transactions on Fuzzy Systems* 25.6 (2016), pp. 1795–1807.

[187] Jochen Garcke and Thomas Vanck. "Importance weighted inductive transfer learning for regression". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*. Springer. 2014, pp. 466–481.

[188] David Pardoe and Peter Stone. "Boosting for regression transfer". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. 2010, pp. 863–870.

[189] Jie Lu et al. "Transfer learning using computational intelligence: A survey". In: *Knowledge-Based Systems* 80 (2015), pp. 14–23.

[190] Mohamed Laid Hadjili and Vincent Wertz. "Takagi-Sugeno fuzzy modeling incorporating input variables selection". In: *IEEE Transactions on fuzzy systems* 10.6 (2002), pp. 728–742.

[191] Tomohiro Takagi and Michio Sugeno. "Fuzzy identification of systems and its applications to modeling and control". In: *IEEE transactions on systems, man, and cybernetics* 1 (1985), pp. 116–132.

[192] Soumi Ghosh and Sanjay Kumar Dubey. "Comparative analysis of k-means and fuzzy c-means algorithms". In: *International Journal of Advanced Computer Science and Applications* 4.4 (2013).

[193] Ciro Castiello et al. "Interpretable fuzzy partitioning of classified data with variable granularity". In: *Applied Soft Computing* 74 (2019), pp. 567–582.

[194] Alexander Fabisch. "gmr: Gaussian mixture regression". In: *Journal of Open Source Software* 6.62 (2021), p. 3054.

[195] Sinno Jialin Pan et al. "Domain adaptation via transfer component analysis". In: *IEEE transactions on neural networks* 22.2 (2010), pp. 199–210.

[196] James M Joyce. "Kullback-leibler divergence". In: *International encyclopedia of statistical science*. Springer, 2011, pp. 720–722.

[197] Masashi Sugiyama et al. "Direct importance estimation for covariate shift adaptation". In: *Annals of the Institute of Statistical Mathematics* 60.4 (2008), pp. 699–746.

[198] Le Song et al. "Hilbert space embeddings of conditional distributions with applications to dynamical systems". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 961–968.

[199] Krikamol Muandet et al. “Kernel mean embedding of distributions: A review and beyond”. In: *Foundations and Trends® in Machine Learning* 10.1-2 (2017), pp. 1–141.

[200] Le Song, Kenji Fukumizu, and Arthur Gretton. “Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models”. In: *IEEE Signal Processing Magazine* 30.4 (2013), pp. 98–111.

[201] Jian Wang et al. “Study of weighted fusion methods for the measurement of surface geometry”. In: *Precision Engineering* 47 (2017), pp. 111–121.

[202] Richard Leach. *Fundamental principles of engineering nanometrology*. Elsevier, 2014.

[203] Zuowei Zhu et al. “Data fusion-based method for the assessment of minimum zone for aspheric optics”. In: *Computer-Aided Design and Applications* 18.2 (2020), pp. 309–327.

[204] Ji Ding et al. “A multisensor data fusion method based on gaussian process model for precision measurement of complex surfaces”. In: *Sensors* 20.1 (2020), p. 278.

[205] Zhongyi Michael Zhang et al. “Applications of data fusion in optical coordinate metrology: a review”. In: *The International Journal of Advanced Manufacturing Technology* (2022), pp. 1–16.

[206] Jian Wang, Richard K Leach, and Xiang Jiang. “Review of the mathematical foundations of data fusion techniques in surface metrology”. In: *Surface topography: metrology and properties* 3.2 (2015), p. 023001.

[207] Kaidi Zhang et al. “A three-dimensional surface measurement system implemented with Gaussian process based adaptive sampling”. In: *Precision Engineering* 72 (2021), pp. 595–603.

[208] Xingyu Yan and Alex Ballu. “Tolerance analysis using skin model shapes and linear complementarity conditions”. In: *Journal of Manufacturing Systems* 48 (2018), pp. 140–156.

[209] Xun Gong and Hsi-Yung Feng. “Cutter-workpiece engagement determination for general milling using triangle mesh modeling”. In: *Journal of Computational Design and Engineering* 3.2 (2016), pp. 151–160.

[210] Julius Schoop, Md Mehedi Hasan, and Hamzah Zannoun. “Physics-Informed and Data-Driven Prediction of Residual Stress in Three-Dimensional Machining”. In: *Experimental Mechanics* 62.8 (2022), pp. 1461–1474.

[211] Zongyi Li et al. “Fourier neural operator for parametric partial differential equations”. In: *arXiv preprint arXiv:2010.08895* (2020).

[212] Xiaomeng Li et al. “H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes”. In: *IEEE transactions on medical imaging* 37.12 (2018), pp. 2663–2674.

[213] Nastaran Enshaei, Safwan Ahmad, and Farnoosh Naderkhani. "Automated detection of textured-surface defects using UNet-based semantic segmentation network". In: *2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE. 2020, pp. 1–5.

[214] Mayank Raj et al. "Estimation of local strain fields in two-phase elastic composite materials using UNet-based deep learning". In: *Integrating Materials and Manufacturing Innovation* 10 (2021), pp. 444–460.

[215] Osman Semih Kayhan and Jan C van Gemert. "On translation invariance in cnns: Convolutional layers can exploit absolute spatial location". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14274–14285.

[216] Haoqi Dong et al. "Stability analysis of thin-walled parts end milling considering cutting depth regeneration effect". In: *The International Journal of Advanced Manufacturing Technology* 113.11 (2021), pp. 3319–3328.

[217] Peter J Baddoo et al. "Physics-informed dynamic mode decomposition (piDMD)". In: *arXiv preprint arXiv:2112.04307* (2021).

[218] Zhiwei Zhao et al. "A New Method for Inferencing and Representing a Workpiece Residual Stress Field Using Monitored Deformation Force Data". In: *Engineering* (2022).

[219] Nikola Kovachki et al. "Neural operator: Learning maps between function spaces". In: *arXiv preprint arXiv:2108.08481* (2021).

[220] Lu Lu et al. "A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data". In: *Computer Methods in Applied Mechanics and Engineering* 393 (2022), p. 114778.

[221] Lu Lu, Pengzhan Jin, and George Em Karniadakis. "Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators". In: *arXiv preprint arXiv:1910.03193* (2019).

[222] Zongyi Li et al. "Neural operator: Graph kernel network for partial differential equations". In: *arXiv preprint arXiv:2003.03485* (2020).

[223] Huaiqian You et al. "Nonlocal kernel network (NKN): a stable and resolution-independent deep neural network". In: *arXiv preprint arXiv:2201.02217* (2022).

[224] Han Gao, Luning Sun, and Jian-Xun Wang. "PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain". In: *Journal of Computational Physics* 428 (2021), p. 110079.

[225] Jürgen Seiler et al. "Resampling images to a regular grid from a non-regular subset of pixel positions using frequency selective reconstruction". In: *IEEE Transactions on Image Processing* 24.11 (2015), pp. 4540–4555.

[226] Junfeng Chen, Elie Hachem, and Jonathan Viquerat. "Graph neural networks for laminar flow prediction around random two-dimensional shapes". In: *Physics of Fluids* 33.12 (2021), p. 123607.

[227] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. "Laplace–Beltrami spectra as 'Shape-DNA'of surfaces and solids". In: *Computer-Aided Design* 38.4 (2006), pp. 342–366.

[228] Wolfgang Arendt et al. "Weyl's Law: Spectral properties of the Laplacian in mathematics and physics". In: *Mathematical analysis of evolution, information, and complexity* (2009), pp. 1–71.

[229] Michael A Epton and Benjamin Dembart. "Multipole translation theory for the three-dimensional Laplace and Helmholtz equations". In: *SIAM Journal on Scientific Computing* 16.4 (1995), pp. 865–897.

[230] Terence Tao. *Fourier transform.* https://www.math.ucla.edu/~tao/preprints/fourier.pdf. 2016.

[231] Guoliang Xu. "Discrete Laplace–Beltrami operators and their convergence". In: *Computer aided geometric design* 21.8 (2004), pp. 767–784.

[232] Chu Wang, Babak Samari, and Kaleem Siddiqi. "Local spectral graph convolution for point set feature learning". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 52–66.

[233] Yifan Qie and Nabil Anwer. "Geometric deviation modeling for single surface tolerancing using Laplace-Beltrami Operator". In: *Procedia CIRP* 114 (2022), pp. 19–24.

[234] Marc Alexa et al. "Properties of laplace operators for tetrahedral meshes". In: *Computer Graphics Forum*. Vol. 39. 5. Wiley Online Library. 2020, pp. 55–68.

[235] Keenan Crane. "Discrete differential geometry: An applied introduction". In: *Notices of the AMS, Communication* (2018), pp. 1153–1159.

[236] Bruno Vallet and Bruno Lévy. "Spectral geometry processing with manifold harmonics". In: *Computer Graphics Forum*. Vol. 27. 2. Wiley Online Library. 2008, pp. 251–260.

[237] Thomas Neumann et al. "Compressed manifold modes for mesh processing". In: *Computer Graphics Forum*. Vol. 33. 5. Wiley Online Library. 2014, pp. 35–44.

[238] Ricardo Vinuesa and Steven L Brunton. "Enhancing computational fluid dynamics with machine learning". In: *Nature Computational Science* 2.6 (2022), pp. 358–366.

[239] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218.

[240] Jacob H Seidman et al. "Nomad: Nonlinear manifold decoders for operator learning". In: *arXiv preprint arXiv:2206.03551* (2022).

[241] Shujian Li, Lihua Zhan, and Tengfei Chang. "Numerical simulation and experimental studies of mandrel effect on flow-compaction behavior of CFRP hat-shaped structure during curing process". In: *Archives of Civil and Mechanical Engineering* 18 (2018), pp. 1386–1400.

[242] P Howard. "Partial differential equations in Matlab 7.0". In: *University of Maryland, College Park (MD)* (2005).

[243] COMSOL Multiphysics. "Introduction to comsol multiphysics®". In: *COMSOL Multiphysics, Burlington, MA, accessed Feb* 9.2018 (1998), p. 32.

[244] Mengwu Guo and Jan S Hesthaven. "Reduced order modeling for nonlinear structural analysis using Gaussian process regression". In: *Computer methods in applied mechanics and engineering* 341 (2018), pp. 807–826.

[245] Marc P Mignolet et al. "A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures". In: *Journal of Sound and Vibration* 332.10 (2013), pp. 2437–2460.

[246] Shuting Liu et al. "A Multi-Zoned Self-Resistance Electric Heating Method for Curing Irregular Fiber Reinforced Composite Parts". In: *Advances in transdisciplinary engineering*. IOS Press, 2021.

[247] Anxin Ding et al. "A three-dimensional thermo-viscoelastic analysis of process-induced residual stress in composite laminates". In: *Composite Structures* 129 (2015), pp. 60–69.

[248] Damien Rolon-Mérette et al. "Introduction to Anaconda and Python: Installation and setup". In: *Quant. Methods Psychol* 16.5 (2016), S3–S11.

[249] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[250] Adam Paszke et al. "Automatic differentiation in pytorch". In: (2017).

[251] C Sullivan and Alexander Kaszynski. "PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)". In: *Journal of Open Source Software* 4.37 (2019), p. 1450.

[252] Joshua Willman and Joshua Willman. "Overview of pyqt5". In: *Modern PyQt: Create GUI Applications for Project Management, Computer Vision, and Data Analysis* (2021), pp. 1–42.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Nouvelle démarche de développement de modèles prédictifs de pilotage par les données en Smart-Manufacturing dans les cas de pénurie de données par l'utilisation de techniques de machine-Learning et l'IA.

Mots clés : Smart manufacturing, Machine learning, Data-driven, Predictive modelling

Résumé :

Le smart-manufacturing basé sur le pilotage par les données a démontré un potentiel très significatif à travers l'ensemble du processus de fabrication. Dans ce sens la littérature propose diverses méthodes d'apprentissage automatique (machine-learning), qui ont été appliquées avec succès pour résoudre différents problèmes de modélisation prédictive en production (MPM). Cependant, la réussite et surtout le niveau de performance de ces méthodes de modélisation prédictives sont directement liées à la qualité et la structuration des données utilisées comme point de départ. Malheureusement, l'analyse et la qualification de ces données issues de la fabrication que ça soit d'un point de vu expérimental ou numérique restent un travail fastidieux, chronophage et coûteux.

Afin de résoudre ce problème de pénurie de données et de qualité de données pour la modélisation prédictive en fabrication (MPM) cette thèse propose une nouvelle méthodologie systématique basée sur l'échantillonnage actif de données directement labellisées et le transfert de connaissances à partir de données auxiliaires et de données physiques combinées.

Les méthodes proposées ont été validées à travers divers cas d'applications, notamment pour la production des composites, la mesure multi-capteurs et la prédiction de l'usure des outils. Les résultats expérimentaux montrent que la démarche développée peut fournir des résultats de prédiction de déformation précis tout en réduisant de manière significative les données d'apprentissage requises de 90% par rapport aux méthodes existantes.

Title : New data-driven predictive modelling methods for data scarcity scenarios in smart manufacturing

Keywords : Smart manufacturing, Machine learning, Data-driven, Predictive modelling

Abstract :

Data-driven smart manufacturing has demonstrated tremendous potential across the entire manufacturing lifecycle. Various machine-learning methods have been developed and successfully applied to address different manufacturing predictive modelling (MPM) problems. However, the superior performance and effectiveness of data-driven predictive modelling methods heavily depend on the availability of a substantial amount of labelled data. Unfortunately, labelling manufacturing data, both computationally and experimentally, is often a costly and time-consuming task.

To deal with the data scarcity of MPM problems, this thesis proposed a systematic framework including the active sampling of direct labelled data, knowledge transfer from the auxiliary data and data-physics combination. An aggregation-value-based sampling was proposed based on the Game theory for sampling the most promising labelled data. A new transfer learning approach was proposed to facilitate the knowledge transfer from auxiliary data to the target task, thus improving the performance of the target model under data scarcity situations. To further leverage the prior knowledge of the manufacturing process, a physics-guided low-dimensional neural operator was developed, which incorporates physics priors into the neural network structure to enhance the learning capabilities.

The proposed methods were validated in various manufacturing cases including composite curing, multi-sensor measurement and tool wear prediction. Experimental results show that the developed system could provide accurate deformation prediction results while significantly reducing the required training data by 90% compared to the existing method.