



**HAL**  
open science

# Improving diagnostics and prognostics in sparse data and scarce knowledge conditions by physics-informed and self-supervised machine learning

Weikun Deng

► **To cite this version:**

Weikun Deng. Improving diagnostics and prognostics in sparse data and scarce knowledge conditions by physics-informed and self-supervised machine learning. Civil Engineering. Université de Toulouse, 2024. English. NNT : 2024TLSEP107 . tel-04845497

**HAL Id: tel-04845497**

**<https://theses.hal.science/tel-04845497v1>**

Submitted on 18 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Doctorat de l'Université de Toulouse

préparé à Toulouse INP

---

Amélioration du diagnostic et du pronostic dans des conditions  
de données rares et de connaissances limitées par  
l'apprentissage automatique informé par la physique et  
auto-supervisé

---

Thèse présentée et soutenue, le 7 novembre 2024 par

**Weikun DENG**

**École doctorale**

SYSTEMES

**Spécialité**

Génie Industriel

**Unité de recherche**

LGP - Laboratoire Génie de Production

**Thèse dirigée par**

Kamal MEDJAHER et Thi Phuong Khanh NGUYEN

**Composition du jury**

M. Nouredine ZERHOUNI, Président, SUPMICROTECH-ENSMM

Mme Mitra FOULADIRAD, Rapporteuse, Centrale Méditerranée

M. Phuc DO, Rapporteur, Université de Lorraine

M. Enrico ZIO, Examineur, Politecnico di Milano

M. Kamal MEDJAHER, Directeur de thèse, UTTOP

Mme Thi Phuong Khanh NGUYEN, Co-directrice de thèse, UTTOP

**Membres invités**

M. Christian GOGU, ISAE-SUPAERO

M. Jérôme MORIO, ISAE-SUPAERO

# Contents

<b>List of Notations</b>	xiii
<b>1 General introduction</b>	6
1.1 Context	6
1.2 Problem statement	8
1.3 Research directions and main contributions	10
1.3.1 Research directions	10
1.3.2 Main contributions	11
1.4 Thesis overview	14
1.5 List of publications and awards	15
1.5.1 Publications	15
1.5.2 Awards	17
<b>2 Literature review</b>	18
2.1 Introduction	18
2.2 Bibliometric literature analysis and quantitative review	19
2.2.1 Research trend analysis	20
2.2.1.1 Evolution of hybrid approaches	20
2.2.1.2 Evolution of unlabelled data learning	21
2.2.2 Statistical analysis of PIML and SSL in PHM	22
2.3 Qualitative literature review	24
2.3.1 An overview of PIML in PHM	24
2.3.1.1 Knowledge informed methods	24
2.3.1.2 Informed knowledge types	36
2.3.2 An overview of SSL in PHM	38
2.3.3 Focus of existing doctoral theses	44
2.4 Summary	46
<b>3 Constructing a generic PINN framework for PHM</b>	49
3.1 Introduction	50
3.2 PIML model based on mimetic theory	52

3.2.1	Investigating physics integration approaches: A simulated bearing degradation case study . . . . .	52
3.2.1.1	Different PIML models embedding the same physics knowledge . . . . .	53
3.2.1.2	Investigation of the PI-TCN models' performance. . . . .	55
3.2.2	New PIML model based on mimetic theory: Rotor finite element mimetic neural network . . . . .	58
3.2.2.1	Mimetic theory . . . . .	59
3.2.2.2	Rotor finite element mimetic neural network . . . . .	60
3.2.2.3	Validation of the proposed RFEMNN . . . . .	65
3.3	Constraint projection for PIML in few-shot faults diagnostics . . . . .	67
3.3.1	Constraint projection theory . . . . .	67
3.3.2	Enhancing RFEMNN's performance in few-shot learning with "Constraint projection" theory . . . . .	69
3.3.3	Application of constrain theory: Few-shot learning for rotor fault diagnostics . . . . .	70
3.3.3.1	Physics consistency validation results . . . . .	71
3.3.3.2	Rotor few-shot diagnostic results . . . . .	72
3.4	A generic PIML framework . . . . .	73
3.4.1	A two-branch parallel PIML framework . . . . .	74
3.4.2	Pretrained-physics alignment multistep training . . . . .	75
3.4.3	Case study: Battery RUL prediction using small early-stage lifecycle data . . . . .	76
3.4.3.1	Comparison results with SOTA models . . . . .	78
3.4.3.2	Comparison results of prediction performance on a new dataset without re-learning . . . . .	79
3.5	Summary . . . . .	80
<b>4</b>	<b>Improved SSL for PHM</b> . . . . .	<b>83</b>
4.1	Introduction . . . . .	84
4.2	Novel contrastive SSL in sequential mining for RUL prediction . . . . .	84
4.2.1	Existing contrastive SSL paradigm problem statement . . . . .	85
4.2.2	Novel contrastive SSL . . . . .	87
4.2.2.1	Contrastive pairs design . . . . .	88
4.2.2.2	Proposed contrastive SSL strategy . . . . .	88
4.2.3	Application of the proposed contrastive SSL for accurate bearing RUL prediction . . . . .	90
4.2.3.1	Proposed model . . . . .	90
4.2.3.2	Result discussion . . . . .	93
4.3	Considering downstream information in SSL pretext task design . . . . .	95
4.3.1	Analysis of the proposed SSL drawbacks . . . . .	95

4.3.2	Deep enhancement of the contrastive SSL . . . . .	97
4.3.3	Validation of the improved contrastive SSL model on bearing prognostics . . . . .	100
4.3.3.1	Benchmark results . . . . .	101
4.3.3.2	Validation results of the proposed method . . . . .	103
4.4	Summary . . . . .	107
<b>5</b>	<b>Improving existing PIML and SSL paradigms to build a common PHM model across scenarios</b> . . . . .	<b>109</b>
5.1	Introduction . . . . .	110
5.2	Extend PIML to active mining of knowledge . . . . .	111
5.2.1	Gated neuron theory for liquid NN . . . . .	112
5.2.2	Discovering physics on unlabeled data . . . . .	114
5.2.3	Validation of the proposed methodology for dynamic identification in robotic arms . . . . .	115
5.2.3.1	Test E2NN performance on real data . . . . .	116
5.2.3.2	Investigation on the E2NN's robustness . . . . .	117
5.3	E2E data-driven model for cross-scenario time series processing . . . . .	118
5.3.1	Gated selective state spaces mechanism . . . . .	119
5.3.2	End-to-End (E2E) CNN-SSM model . . . . .	121
5.3.3	Performance evaluation of E2E CNN-SSM model on different datasets . . . . .	123
5.3.3.1	Bearing prediction results . . . . .	123
5.3.3.2	Battery prediction results . . . . .	125
5.4	A generic model incorporating the advanced PIML and SSL . . . . .	126
5.4.1	Liquid physics-informed CNN-SSM model . . . . .	128
5.4.2	Hybrid learning strategy . . . . .	129
5.4.2.1	Supervised information generation . . . . .	129
5.4.2.2	Simultaneous supervised and self-supervised learning . . . . .	130
5.4.3	Investigating the performance of liquid PI CNN-SSM model . . . . .	131
5.4.4	Interpretable outcomes of knowledge representation after Hybrid learning . . . . .	139
5.5	Conclusion . . . . .	143
<b>6</b>	<b>Conclusion and future work</b> . . . . .	<b>145</b>
6.1	Summary of our contributions . . . . .	145
6.2	Discussion of our key findings . . . . .	147
6.2.1	Insights and challenges in PIML and SSL for PHM applications . . . . .	148
6.2.2	Building generic PHM model by thinking outside the box . . . . .	149

6.2.2.1	A generic architectural approach to PIML modeling . . . . .	149
6.2.2.2	SSL pretext task design considering downstream PHM tasks and information . . . . .	149
6.2.2.3	Breaking the old PIML and SSL paradigms to solve the problem of sparse data and scarce knowledge . . . . .	150
6.3	Limitations and Future work . . . . .	151
6.3.1	Limitation of the existing research . . . . .	151
6.3.2	Future perspectives . . . . .	152
<b>A</b>	<b>Appendix of Introduction</b> . . . . .	<b>181</b>
A.1	“Ill-posed” problematic nature . . . . .	181
<b>B</b>	<b>Appendix: Litterature review</b> . . . . .	<b>183</b>
B.1	Supplementary notes on PIML and SSL in PHM . . . . .	184
B.1.1	Statistical analysis of PIML in PHM . . . . .	184
B.1.2	Statistical analysis of SSL in PHM . . . . .	186
B.2	Review articles on PIML . . . . .	190
B.3	Focus of existing doctoral theses . . . . .	193
<b>C</b>	<b>Appendix: Physics-informed machine learning in PHM</b> . . . . .	<b>206</b>
C.1	Detailed description of the data generation process for the validation case in Section 3.2.1 . . . . .	206
C.2	Deep discussion on the performance of the PIML models proposed in Section 3.2.1 . . . . .	208
C.3	Rotor unbalance and shaft crack experiments . . . . .	212
C.4	RFEMNN structure details and the related physics introduction . . . . .	215
C.5	Multi-task supervised training . . . . .	216
C.6	RFEMNN’s evaluation metrics . . . . .	217
C.7	Supplementary comparison of RFEMNN and other benchmark models . . . . .	218
C.7.1	Recognition results of the proposed RFEMNN and other SOTA models . . . . .	218
C.7.2	Fault location results . . . . .	222
C.8	SEI-DCN model structure . . . . .	223
C.9	Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models . . . . .	227
<b>D</b>	<b>Appendix: Improved Self Supervised Learning strategy for Prognostics and Health Management</b> . . . . .	<b>237</b>

D.1	Novel contrastive metrics for SSL . . . . .	237
D.2	Prognostia bearing dataset and the data processing method for the proposed SSL approach validation . . . . .	238
D.2.1	PRONOSTIA dataset introduction . . . . .	238
D.2.2	Data preprocessing setting . . . . .	238
D.3	CNN-LSTM benchmark model configuration . . . . .	238
D.4	Qualitative prediction of the degradation trajectory . . . . .	239
D.5	Deep enhancement of the contrastive SSL . . . . .	239
D.6	Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses . . . . .	244
D.6.1	Qualitative results . . . . .	244
D.6.2	Uncertainty in predictions . . . . .	246
D.6.3	Computation cost comparison . . . . .	246
D.6.4	Impact of labeled data availability on the proposed SSL model . . . . .	248
D.6.5	Impact analysis of model architecture . . . . .	249
D.6.6	Generalization test of the proposed SSL strategy . . . . .	253
<b>E</b>	<b>Appendix: Breaking the old PIML and SSL paradigms to build a common PHM model across scenarios</b> . . . . .	<b>255</b>
E.1	Detailed description of benchmarking models based on deep residual contraction networks . . . . .	255
E.2	Proposed metric for evaluating inversion results of robotic arm dynamics . . . . .	256
E.3	Supplementary validation results and discussion of the proposed E2NN performance . . . . .	257
E.4	Computational cost of the Transformer model . . . . .	260
E.5	A detailed derivation of the gating selective state space mechanism	261
E.6	Performance evaluation of End-to-End CNN-SSM model on different datasets . . . . .	264
<b>F</b>	<b>Abstract</b> . . . . .	<b>269</b>

## List of Figures

1.1	PHM roles in the machine lifecycle. . . . .	6
-----	---	---

1.2	Dual challenges on the physics side and the data side. . . . .	8
1.3	“Ill-posed” problem in PHM modelling. . . . .	10
1.4	Structure and main contributions of our thesis. . . . .	14
2.1	Schematic flow of the bibliometric analysis. . . . .	19
2.2	A technological roadmap for hybrid model architectures. . . . .	20
2.3	A technological roadmap for unlabelled data learning. . . . .	21
2.4	Main application areas and data sources of PIML in PHM. . . . .	22
2.5	Main application areas and data sources of SSL in PHM. . . . .	23
2.6	Taxonomy of existing PIML methods in PHM. . . . .	25
2.7	Two ways for embedding physics knowledge as a ML operator. . . . .	33
2.8	Two ways to construct physics-constrained learning. . . . .	36
2.9	Different knowledge forms of PIML in PHM. . . . .	37
2.10	The main steps in self-supervised learning. . . . .	39
2.11	Taxonomy of self-supervised learning in PHM. . . . .	39
2.12	Scientific positioning of our thesis. . . . .	47
3.1	Research framework for constructing a generic PIML in PHM. . . . .	51
3.2	Lightweight TCN architecture diagram. . . . .	53
3.3	Updated model configuration with physics-informed health indicator. . . . .	54
3.4	Embedding physics equations into a neural network layer. . . . .	54
3.5	Constructing Physics-Informed Loss based on model conflicts. . . . .	55
3.6	Prediction results of different models. . . . .	56
3.7	Model evaluation results. . . . .	57
3.8	Loss landscapes of three PIML models. . . . .	57
3.9	Illustration of mimetic theory in the context of neural networks. . . . .	59
3.10	Mimetic theory for deriving logical and operator content alignment. . . . .	60
3.11	FEM-based rotor fault diagnostic. . . . .	61
3.12	Schematic diagram of the proposed RFEMNN model. . . . .	62
3.13	Structure of a FEM elementary neuron in NN layer. . . . .	63
3.14	Building customized layers according to FEM assembly rules. . . . .	63
3.15	Vibration behaviour-based physics consistency supervises the mimetic process. . . . .	64
3.16	Structure of CNN-LSTM layer to mimic the FEM solution process. . . . .	65
3.17	Average results of fault localization with 10 fold-cross validation on diagnostics fault location. . . . .	66
3.18	Learning strategies with different solution ideas. . . . .	68
3.19	Workflow for enhancing RFEMNN performance using constraint projection driven by reinforcement learning. . . . .	69



3.20	Applying RL-driven constraint theory in RFEMNN model's few-shot learning. . . . .	71
3.21	Vibration reconstruction (left) and physics consistency (right) investigation after RL fine-tuning. . . . .	72
3.22	Embedding physical knowledge into neural network models using parallel architectures. . . . .	74
3.23	Novel PIML learning strategy. . . . .	75
3.24	SEI informed DCNN model diagram. . . . .	77
3.25	Predicted trajectory results for the new test battery packs's remaining discharging cycles. . . . .	80
4.1	Physiologically degraded monitoring processes. . . . .	87
4.2	Construction of contrastive sample pairs. . . . .	88
4.3	Multi-hierarchy Siamese models with nested structural reuse. . . . .	89
4.4	Benchmark CNN-LSTM network architecture. . . . .	91
4.5	Siamese CNN-LSTM implementation diagram. . . . .	92
4.6	Prediction errors of different models. . . . .	93
4.7	Performance of SSL and CNN-LSTM (Benchmark) models at different percentages of the labeled degradation trajectories in the training set. . . . .	94
4.8	Negative impacts of converting normalized labels to true labels on downstream task predictions. . . . .	96
4.9	Self-supervised learning considering downstream information. . . . .	96
4.10	Contrastive SSL architecture considering downstream information. . . . .	97
4.11	Inversion of the input temporal feature dimension to generate the contrastive tensor sample. . . . .	98
4.12	Basic structure of feature extractor. . . . .	98
4.13	Basic structure of failure time predictor. . . . .	99
4.14	The structure of RUL calculator in the proposed neural network. . . . .	99
4.15	CNN-LSTM network architecture. . . . .	100
4.16	Prediction results of Benchmark model. . . . .	102
4.17	Box plots of quantitative statistics of Benchmark model prediction errors. . . . .	102
4.18	Prediction results of SSL model. . . . .	104
4.19	Box plots of quantitative statistics of SSL model prediction errors. . . . .	104
4.20	Box plots of the impact of different amounts of labelled data on the prediction results of the model. . . . .	106
5.1	Research framework for building the generic PHM model across scenarios based on the combination of improved PIML and SSL paradigms. . . . .	110

5.2	From knowledge embedded representation to active discovery. . .	112
5.3	Differences between attention mechanism and liquid NN dynamics.	113
5.4	Liquid neuron design. . . . .	114
5.5	E2NN: Deep Residual Shrinkage Network with embedded equations. . . . .	115
5.6	Prediction results on real robot manipulators. . . . .	117
5.7	Selectivity mechanism to construct input-dependent state matrices.	120
5.8	Parallel scanning algorithm for accelerated computation to update hidden state [181]. . . . .	121
5.9	End-to-End CNN-SSM model architecture for processing variable input length sequences. . . . .	122
5.10	Violin diagram for prediction error analysis of bearing RUL. . . .	124
5.11	Violin diagram for prediction error analysis of battery RUL. . . .	125
5.12	Generic end-to-end cross-scenario time series processing model for prognostic tasks. . . . .	126
5.13	A multi-step hybrid learning strategy for generic model training. .	127
5.14	Auto-discovered knowledge informs CNN-SSM in liquid form. . .	128
5.15	Generating labels for unlabelled data for supervised learning. . . .	130
5.16	Prediction results of the same CNN-SSM with Physics-informed CNN-SSM on multiple datasets. . . . .	132
5.17	Performance difference in bearing RUL predictions before and after physics learning. . . . .	135
5.18	Performance difference in battery RUL predictions before and after physics learning. Polar axis range [-1, 0.5]. . . . .	136
5.19	Performance difference in tool wear predictions before and after physics learning. . . . .	137
5.20	Performance difference in CFRP tubes remaining loading cycle predictions before and after physics learning. . . . .	138
5.21	Operator weight changes in the actively discovered knowledge about the horizontal vibration during 1st bearing's degradation. .	140
5.22	Operator weight changes in the actively discovered knowledge about the vertical vibration during 1st bearing's degradation. . . .	141
5.23	Operator weight changes when actively discovering knowledge from horizontal vibration signals during 2nd bearing's degradation.	142
C.1	Simulation process of contact stiffness degradation. . . . .	207
C.2	Defective contact and roller failure schematic. . . . .	207
C.3	Heat map of the PI-input space model. . . . .	208
C.4	Heat-map of PI-layer model. . . . .	209
C.5	Heat map of the PI-loss model "Add" Layer. . . . .	209
C.6	Comparison of loss landscapes under different conditions. . . . .	211

C.7	Schematic diagram of the experimental platform. . . . .	212
C.8	Different structure layouts of the test bench. . . . .	214
C.9	Optimized models selected based on fault identification, unbalance localization, and crack localization. . . . .	216
C.10	Average results of fault localization with 10 fold-cross validation on diagnostics fault location. . . . .	218
C.11	Average results of ANN-based fault localization with 10 fold-cross validation[364]. . . . .	222
C.12	Average results of LSTM-based fault localization with 10 fold-cross validation[365]. . . . .	222
C.13	Average results of ELM-based fault localization with 10 fold-cross validation[363]. . . . .	223
C.14	Average results of NO_FEM_NN based fault localization with 10 fold-cross validation. . . . .	223
C.15	Truncate the original sequence in a fixed-length time window to construct an end-to-end input. . . . .	224
C.16	Schematic diagram of D-CNN with a data flow. . . . .	225
C.17	Demonstration of predicted trajectory results for remaining discharging cycles of test battery packs. . . . .	231
C.18	Box plots the prediction errors of remaining discharging cycles (RDCs) for DCNN and SEI-DCN (our approach). . . . .	232
C.19	Schematic depiction of the modulation in channel weights after integrating features across various branches during the multi-step learning phases. . . . .	234
C.20	SEI layer thickness growth model informed DCNN functional module integration diagram. . . . .	235
D.1	The result of the discrete wavelet transformation. . . . .	239
D.2	RUL predictions on 10 different degradation trajectories of bearings. . . . .	241
D.3	Innovation points of the proposed new SSL learning paradigm. . . . .	242
D.4	Novel SSL workflow for implementation of the proposed solution in pretext task's framework. . . . .	243
D.5	Inversion of the input temporal feature dimension to generate the contrastive tensor sample. . . . .	244
D.6	Basic structure of feature extractor. . . . .	244
D.7	Basic structure of failure time predictor. . . . .	244
D.8	Horizontal direction vibration. . . . .	245
D.9	Vertical direction vibration. . . . .	245
D.10	Prediction uncertainty result of the proposed method in 50 times Monte Carlo tests. . . . .	247

D.11	Box plots of absolute prediction errors under the impact of different amounts of labeled data. . . . .	249
D.12	Comparison of prediction results of failure times under different paths. . . . .	250
D.13	Prediction error when residual blocks of different layers are connected to the predictor. . . . .	251
D.14	Recognition ability of different models for sequential data under 100% labeling degradation process. . . . .	252
D.15	Recognition ability of different models for sequential data under 50% labelling degradation process. . . . .	252
D.16	Comparison of tool volume wear degree prediction results. . . . .	254
E.1	Framework of the deep residual shrinkage network (DRSN). . . . .	255
E.2	Comparison of the performance of the DRSN (above) and E2NN (below). . . . .	258
E.3	Reconstructed torque by using Matlab. . . . .	260
E.4	Qualitative representation of bearing normalised RUL prediction results. . . . .	265
E.5	Qualitative representation of battery normalized RUL prediction results. . . . .	265
E.6	Operator weights change when actively discovering knowledge from <b>current</b> signals during the 1st battery degradation. . . . .	267
E.7	Operator weights change when actively discovering knowledge from <b>temperature</b> signal during the 1st battery degradation. . . . .	268

## List of Tables

1.1	“4Cs” requirements. . . . .	13
2.1	Summary of physics-informed input space studies in PHM. . . . .	25
2.2	Summary of physics-embedded algorithm structure in PHM. . . . .	28
2.3	Summary of Physics-constraint learning in PHM. . . . .	34
2.4	Summary of SSL in PHM. . . . .	40
2.5	Unresolved challenges in PIML. . . . .	45
2.6	Unresolved challenges in SSL. . . . .	45

3.1	Comparison of the proposed RFEMNN model with the SOTA models. . . . .	66
3.2	Diagnostic results across various limited-labelled-data scenarios. . . . .	73
3.3	Performance of different models. . . . .	78
4.1	Comparison of the prediction MAE of different models. . . . .	93
4.2	Comparison of different models on the normalized RUL predictions. . . . .	105
5.1	Validation on real-world data. . . . .	116
5.2	Comparison of the robustness of different methods across various metrics when applied to new data. . . . .	118
5.3	Cross-scenarios dataset introduction. . . . .	133
5.4	Updated prediction accuracy of different models. . . . .	133
B.1	Summary of the training metrics, testing metrics, and monitoring signals for PIML according to PHM tasks. . . . .	185
B.2	Summary of applications, models, and signal types for SSL in PHM. . . . .	186
B.3	Existing review papers that mention PIML in PHM . . . . .	191
B.4	Summary of recent PhD theses on physics-informed machine learning (PIML). . . . .	193
C.1	Overview of the experimental setup. . . . .	213
C.2	Illustration of the data labeling. . . . .	215
C.3	Physics parameters in RFEMNN. . . . .	215
C.4	Confusion matrix for multiple fault diagnostics results. . . . .	217
C.5	Confusion matrices of RFEMNN faults recognition. . . . .	219
C.6	Confusion matrices of CNN (raw data as input) [359]. . . . .	220
C.7	Confusion matrices of CWSCNN [359]. . . . .	220
C.8	Confusion matrices of DRSNs (wavelet spectrum as input) [360]. . . . .	220
C.9	Confusion matrices of DRSNs (raw data as input) [360]. . . . .	221
C.10	faults recognition confusion matrices of CNN-SVM[361]. . . . .	221
C.11	faults recognition confusion matrices of STFNN[362]. . . . .	221
C.12	faults recognition confusion matrices of STFNN. . . . .	221
C.13	Data processing and hyperparameters for benchmark model. . . . .	228
C.14	Hyperparameters in physics informed branch. . . . .	229
C.15	Schematic representation of the differences in SEI-DCN compared to the original DCNN model. . . . .	230
C.16	Performance metrics of different PIML models. . . . .	235
D.1	Common Contrast Loss Functions. . . . .	237
D.2	Data processing and learning hyperparameters for benchmark model. . . . .	240

D.3	Comparison of the computation costs of different methods. . . . .	248
E.1	Comparison of different methods on various metrics. . . . .	258
E.2	Results of $\beta$ estimation for the robotic manipulators torque model.	259

# List of Notations

## Abbreviations

AE	Auto-Encoder
ANN	Artificial Neural Network
CFRP	Carbon Fiber Reinforced Polymer
CNN	Convolution Neural Network
D-CNN	Dilated Convolution Neural Network
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network
DRSN	Deep Residual Shrinkage Network
E2E	End-to-End
E2NN	Equation Embedded Neural Network
FCN	Fully Convolution Network
FEM	Finite Element Method
FLOPs	Floating Point Operations
GMP	Global Max Pooling
GRU	Gated Recurrent Unit
HBM	High Bandwidth Memory
HCP	Hard Constraint Projections
HI	Health Indicator
IQR	Interquartile Range
KL	Kullback-Leibler
LSTM	Long-Short Term Memory

MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
PBM	Physics-Based Model
PIML	Physics-Informed Machine Learning
PINN	Physics-Informed Neural Network
ReLU	Rectified Linear Unit
RFEMNN	Rotor Finite Element Mimetic Neural Network
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SEI	Solid Electrolyte Interphase
SEI-DCN	SEI-informed Dilated CNN
SRAM	Static Random-Access Memory
SSIM	Structural Similarity Index
SSL	Self-Supervised Learning
SSM	Selective State Space Model
TCN	Temporal Convolution Network

### **Symbols**

$\alpha_t, \alpha_m, \alpha_s, \alpha_v$	Weights for loss function components
$\bar{A}, \bar{B}, \bar{C}, \bar{D}$	State space matrices
$\ddot{q}$	Acceleration of the rotor
$\Delta$	Time-varying parameter in state space models
$\dot{q}$	Velocity of the rotor



$\epsilon_{inv}(t)$	Compound noise and model uncertainties
$\gamma$	Contact angle
$\mathcal{A}^*$	Pre-trained mapping function used in SSL for feature extraction
$\mathcal{A}_{inv}$	Operator representing the inverse PHM model
$\mathcal{B}$	Input Matrix in SSM
$\mathcal{C}$	Output Matrix in SSM
$\mathcal{D}$	Feedforward Matrix in SSM
$\mathcal{G}_{inv}$	Observed data
$\mathcal{L}_2$	Conservation loss function used to ensure consistent failure time predictions
$\mathcal{L}_{contrast}$	Contrastive loss function used to train the SSL model
$\mathcal{T}$	Transformation function used in the contrastive self-supervised learning (SSL) model
$\bar{p}$	Average localization accuracy
$Ac$	Accuracy
$C$	Empirical factor in SEI growth model
$D$	SEI degradation parameter
$E_a$	Effective activation energy for SEI growth
$F(q, t)$	Excitation force of the rotor
$Fa$	False alarm rate
$FF_1, FF_2$	Feedforward layers in the gated neuron
$g$	Gate value in the gated neuron theory
$h(t)$	Hidden state at time $t$
$k_B$	Boltzmann's constant
$k_p$	Hertzian elastic contact stiffness

$L_C$	Mean absolute error of unbalanced defects and shaft cracks
$L_m, L_s, L_t, L_v$	Components of the composite loss function
$L_U$	Mean absolute error of unbalanced defects and shaft cracks
$Len_i$	Length of shaft element $i$
$M$	Mass matrix of the rotor
$m_i$	Equivalent elementary mass
$Mis$	Missing rate
$n_s$	Number of contact surfaces
$n_t$	Negative sequence features representing the inverse degradation trend
$p_t$	Positive sequence features representing the natural progression of degradation
$p_{inv}$	System state or parameter to be estimated
$q$	Vibration displacement
$R$	Roller radius
$R_d$	Shaft section outer diameter
$r_d$	Shaft section inner diameter
$RT$	Operation time, used to calculate RUL
$S$	Stiffness matrix of the rotor
$s_i$	Equivalent elementary stiffness
$s_{ji}$	Support stiffness
$T$	Battery monitoring temperature
$Tre$	Failure time predicted by the failure time predictor in the SSL model
$U_{deg}$	Model uncertainty for degradation
$Vib_p$	Peak value of the vibration signal

# Acknowledgment

The past three years in Tarbes have been an extraordinary journey filled with challenges, growth, and countless unforgettable moments. Moving to a new country, adapting to a different culture, and diving into the depths of academic research have been transformative experiences. Tarbes, with its serene beauty and warm community, has been more than just a place to live—it has been a home that supported me through every step of this journey. As I look back, I am deeply grateful for the opportunities, the people, and the memories that this city has given me.

First and foremost, I would like to express my deepest gratitude to my PhD director, Prof. Kamal MEDJAHHER, for his invaluable guidance and encouragement throughout this journey. He taught me not only how to analyze problems from diverse perspectives—like the flow of Tai Chi—but also how to ask meaningful scientific questions, a skill he considers essential for any PhD researcher. At the same time, he told me that the study was to be conducted for the purpose and description of a tree in the plains, not like exploration in the rainforest, which is what I need. Prof. Kamal’s mentorship has shaped my ability to prioritize tasks and swiftly develop optimal strategies for advancing projects. Beyond academia, he has always shown genuine care for international students like me, often reminding us to eat well, rest well, and enjoy life in France. His energetic, youthful spirit and unwavering positivity have been a constant source of inspiration. I am profoundly grateful for his insightful scientific philosophy and for his vision, his student-first vision of empowering every student to contribute to their field and build a meaningful life.

My heartfelt thanks also go to my co-director, Prof. Khanh Thi Phuong NGUYEN, who has been a pillar of strength and support throughout my research. Her wisdom, elegance, and kindness are unparalleled. From helping me navigate the software and integrate seamlessly into the team, to providing backup plans when challenges arose, she has always been there with a calm and encouraging presence. Prof. Khanh taught me the art of parallel and efficient work, rescuing me from countless chaotic and naïve ideas, and helping me identify the most brilliant contributions. Beyond academia, she has been like an elder sister, extending warmth through memorable moments like the barbecue at her home and heartfelt Christmas cards. She has understood the solitude and challenges of studying abroad and consistently encouraged me to persevere toward a fruitful future. For her mentorship and compassion, I am truly grateful.

To Prof. Christian GOGU, I owe sincere thanks for his insightful guidance and visionary thinking. He is like an eagle with a far-reaching view, elevating

my methodologies and ideas through our discussions. I regard him as a "Michelin chef" in research, capable of adding a touch of brilliance to any work. I will never forget the awe I felt when I first encountered his insights on integrating upstream and downstream tasks in SSL or his thoughts on multi-step alignment in PIML—it was as if a voice inside me shouted, "Yes, this is the path I want to do" I am also deeply thankful for the exchange opportunity he provided, making my three months at UCF a short yet memorable and enriching experience.

To Prof. Jérôme Morio, my journey began with you. It was your email response to my PhD application that set everything into motion, bringing me to this exceptional team. Over the past three years, you have been a constant source of encouragement and support. Whether it was constructing metrics, crafting arguments for papers, or discussing the broader research narrative, your selfless assistance has been invaluable. That one long letter after another from you is full of positive energy. Thank you for your unwavering help and for being a mentor in both the finer and larger aspects of research.

I would like to extend my heartfelt gratitude to the esteemed members of the jury for their invaluable support, insightful feedback, and encouragement during my thesis defense. Prof. Zerhouni, your thoughtful questions, long-standing contributions to PHM, and inspiring discussions have greatly enriched my academic journey since my master's studies, and I feel truly honored to have had the opportunity to engage with you. Prof. Mitra, your detailed and encouraging feedback, as well as your insightful comments on methodology and future possibilities, have been deeply motivating, and I am grateful for the chance to discuss these topics further with you. Prof. Enrico, your profound insights from both engineering and academic perspectives, as well as your encouraging affirmation, have provided immense inspiration, and I deeply appreciate your willingness to explore practical applicability and potential collaborations. Prof. Phuc Do, your thorough review, constructive insights, and thoughtful discussions on methodology and future directions have been invaluable, and I am truly excited about the prospect of collaborating with you and Prof. Khanh on future endeavors. Thank you all for your time, expertise, and unwavering support, which have made this milestone a truly memorable and meaningful experience.

I extend my heartfelt gratitude to the University of Technology Tarbes Occitanie Pyrénées (UTTPO), a newly established yet vibrant institution. As the fourth University of Technology in France, UTTPO unites the National School of Engineers of Tarbes (ENIT) and the University Institute of Technology of Tarbes (IUT). I believe this collaboration fosters a dynamic environment for education, research, and innovation. I sincerely wish UTTPO continued success and growth in its academic and education endeavors.

I am sincerely grateful for the fruitful collaboration with Prof. Wu Dazhong at University of central Florida, thanks the insightful discussions we've shared,

which have significantly advanced my research. Moreover, his unwavering support during my award applications has been invaluable, and I deeply appreciate your mentorship and guidance throughout this process.

Besides these, in the journey of life, there are some people whose presence shapes our experience and whose friendship becomes a cornerstone of our happiness. To my dear brother and companion, Sagar JOSE, I owe endless gratitude for the memories we created together. From our culinary experiments in the kitchen to lounging on your sofa watching movies, from strolling the streets of Tarbes and climbing with foods, while sharing our thoughts to chatting endlessly about our dreams for research, love, careers, and the future, every moment with you has been a blend of wisdom and laughter. Even during those months when our exchange programs kept us apart, I always felt the deep connection of our minds. You promised to be the godfather of my child and spoke passionately about waiting for AGI and a better world—may every one of your dreams come true, and may life bring you nothing but happiness and success.

To Yan YAN, who has been my companion through life's many twists and turns, we used to travel through France, from south to north with our bike. We have seen the Atlantic sunsets and foggy mornings in Paris together, your optimism and support are admirable. Whether we were venting about life, sharing worries about the future, or simply exchanging ideas, your wisdom and positivity were infectious. You are a chef not only in the kitchen but also in life, and your grace and intelligence inspire me. I wish you a life as beautiful and fulfilling as yours.

To Zhengyu LIU, whose polite knock on the door became a comforting routine in the lab, thank you for your kindness, breadth of knowledge, and humility. Your youthful energy and profound insights made our conversations deeply memorable, and your culinary talents are simply unforgettable. I wish you success in your PhD and a life filled with happiness and fulfillment as you chase your dreams.

To Duc An NGUYEN, whose radiant smile and diligent spirit have left a lasting impact, I am grateful for our friendship that feels like destiny. Your rapid growth in cooking and research is nothing short of inspiring, and your reflections on life resonate deeply with me. May your journey ahead be smooth, your efforts rewarded, and your laughter never cease.

To Raza, my adventurous brother, thank you for driving me through storms and sharing your grand ideas and deep perspectives. Your courage and thoughtful approach to life are truly admirable. I wish you all the best as you continue your journey with clarity and purpose.

To Rogers, whose very mention brings a smile to my face, thank you for filling my life with joy and light-heartedness. Though some of our plans, like hitting the gym, remain unfulfilled, your dedication to work, research, and relationships

is inspiring. May you always meet life's challenges with your signature humor and strength.

To Fabio and Eric, my chess coaches, collaborators, and another cherished brothers, thank you for being the queen in the chessboard of life—versatile, capable, and admired by all. Your kindness and leadership have earned you respect, and your friendship is a treasure I deeply value. Martin, with his quiet brilliance and calming presence, has also been a source of support and inspiration. Meeting all of you on my first day in the lab set the tone for my wonderful years in Tarbes. Martin also shared with me his wifi account and the weekend we met at Lola's house for drinks, which started my three wonderful years at Tarbes, and later on I met Delphine, Ajdin, and I'm thankful that they created such a good atmosphere in the lab, helping each other, and that the PhD students are like family.

To Delphine, Ajdin, SylvainLe hung and the rest of our lab family, thank you for fostering an atmosphere of warmth and mutual support, where we worked together like a family. You all have made these years truly unforgettable, and I am grateful beyond words.

I want to express my heartfelt gratitude to the Tarbes Badminton Club (BAT) and to all Mathieu, Cami, Yanick, Fabrice, Jeremmy, Régis, Michael, and so many others. Together, we fought fiercely, celebrated victories, and faced countless defeats in the group stages, but through it all, the spirit of "the shuttle never drops, and we never give up" shone brightly. Your genuine passion for the sport and unwavering determination have been truly inspiring. I am deeply grateful to have met you all, to have shared the court, and to have been part of this incredible journey together.

Lastly, my deepest thanks go to my family and my wife, Geyang CAO, and my family, for their unwavering love and support. Your belief in me has been my foundation and strength throughout this PhD journey. Also I am deeply grateful to my country for the unwavering support and recognition I have received through its scholarships, which have been a testament to its belief in my potential and aspirations. These opportunities have not only fueled my academic journey but also strengthened my determination to give back and contribute to its future. I wish my beloved homeland eternal prosperity, strength, and progress, as it continues to inspire and nurture countless dreams.

At the moment the defense was completed, I saw the wisps of layered clouds blowing up from the banks of the river in my hometown across the rolling deserts of Central Asia, the shimmering waves of the Mediterranean Sea, say hi to the Atlantic Ocean, and then scatter among the mountains and forests of the Pyrenees and melt into the snow that covered the ground. I saw my friends, all of us on the shores of a lake in the mountains, facing the full light of the sky, talking about all those things about the future and life, and downing the wine with the

sound of the wind in the most unimpeded way. The future belongs to us, we will change the world and bring new colors and flavors to everything, and this is to, where I have stayed and walked for the past three years, and I love it all dearly.

# General introduction

## Contents

<b>1.1</b>	<b>Context</b> . . . . .	<b>6</b>
<b>1.2</b>	<b>Problem statement</b> . . . . .	<b>8</b>
<b>1.3</b>	<b>Research directions and main contributions</b> . . . . .	<b>10</b>
1.3.1	Research directions . . . . .	10
1.3.2	Main contributions . . . . .	11
<b>1.4</b>	<b>Thesis overview</b> . . . . .	<b>14</b>
<b>1.5</b>	<b>List of publications and awards</b> . . . . .	<b>15</b>
1.5.1	Publications . . . . .	15
1.5.2	Awards . . . . .	17

## 1.1 Context

According to the World Bank, global manufacturing, valued at €44.3 trillion in 2023, is expected to grow by 2.7% in 2024. In this vast sector, even a 1% efficiency improvement can significantly restructure the entire industrial landscape, which is precisely where Prognostics and Health Management (PHM) can offer significant enhancement.

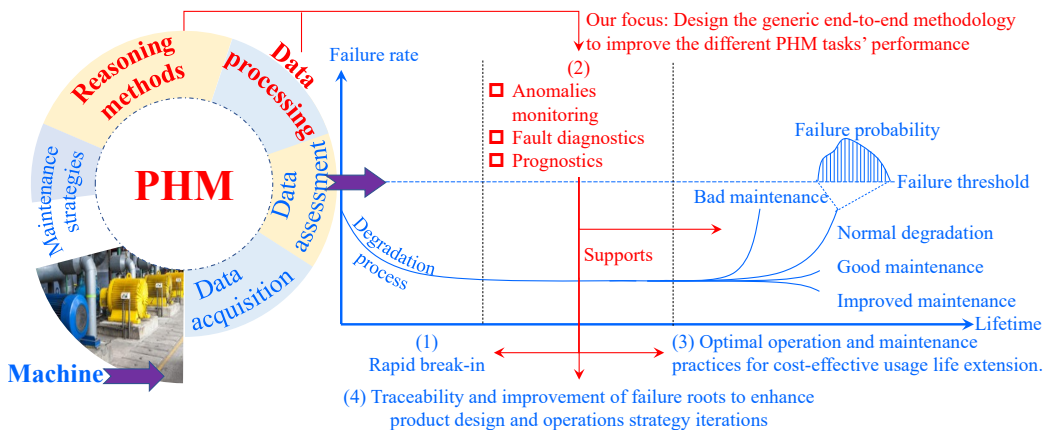


Figure 1.1: PHM roles in the machine lifecycle.



PHM is interdisciplinary engineering. It utilizes current and historical data for real-time monitoring, diagnostics, and prognostics of machines, aiming to predict failures and provide cost-effective maintenance support for extending the usage life [1]. As shown in Fig. 1.1, the PHM pipeline involves acquiring data through sensors and pre-processing it to extract health state insights for abnormal detection, fault diagnostics, and remaining useful life (RUL) predictions. Within these results, throughout the product degradation cycle, PHM technology can be employed to expedite break-in during the early phase, prolong stable operation in the mid-phase, and detect anomalies for timely diagnosis, and enables the proactive formulation of an optimized maintenance strategy for the rapid degradation phase at the later stage. Ultimately, PHM enhances manufacturing efficiency by reducing downtime, which aligns with different national strategies such as the United States “Re-industrialization Strategy,” the EU’s “Horizon Europe Strategic Plan 2025-2027”, and Chinese “New Productivity Force.”

Applying PHM, which brought visible benefits, has driven the market, valued at \$6.5 billion in 2023, to an expected 17% compound annual growth rate through 2028. For example, in the operations and maintenance of F-35 air fighters, PHM reduces non-reproducible faults by 82%, cuts maintenance manpower by 40%, lowers logistics costs by over 50%, and extends the aircraft’s lifespan to 8000 flight hours. Similar in civilian applications, 95% of companies report improved KPIs and a positive return on investment from PHM [2], with 60% noting a 9% increase in machine uptime and 27% achieving amortization within a year [3, 4]. Therefore, PHM services prompt machinery’s role in generating future cash flows by cost-effective continuous production, beyond merely being fixed assets in economic life.

Despite widespread recognition, challenges persist in the implementation of PHM solutions. According to PHM market reports, the sustainable development of PHM services, particularly those integrating machine learning (ML) and deep learning (DL), faces critical obstacles due to limitations in data quality, quantity, and the consistency of physics knowledge [2, 5]. Numerous successes often exist only in exceptional pilot projects and are difficult to replicate, leading to a crisis of trust and deployment of advanced PHM technologies.

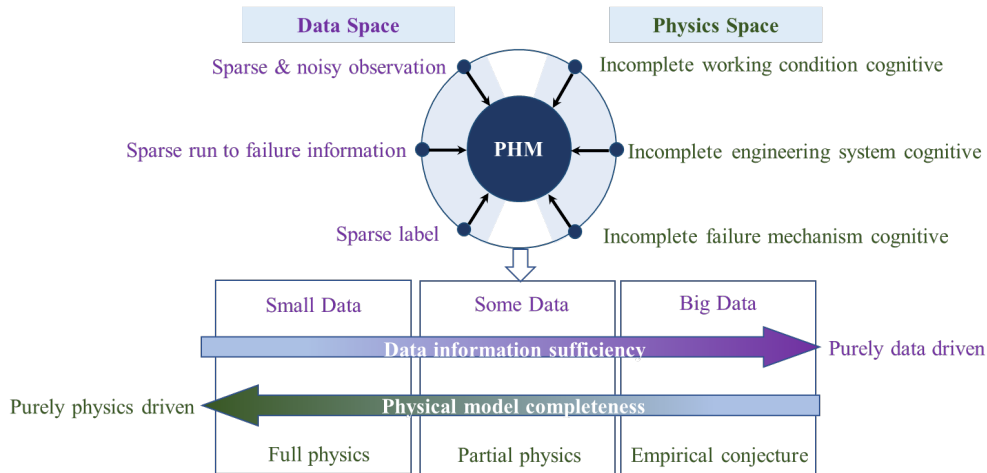
In this context, *our thesis aims to advance the theoretical foundations and practical deployment solutions of PHM, ultimately proposing a generic PHM model capable of effective replication and adaptation across various scenarios and evolving conditions.*

In the following 4 sections, we provide a comprehensive overview of the underlying dilemma we encountered and outline the innovative research program proposed in this chapter. In Section 1.2, the problem statement offers a detailed overview of the key challenges and motivations that underpin this research. Section 1.3 addresses the research issues that arise from the problem statement, fol-

lowed by a comprehensive discussion of the main contributions of our thesis. Finally, Section 1.4 outlines the structure of this thesis, offering a clear roadmap for the reader. It summarizes the content and organization of each subsequent chapter, guiding the reader from the identification of the research problem to the proposed solutions and their comprehensive analysis throughout the thesis. Section 1.5 lists the publications and awards related to this thesis.

## 1.2 Problem statement

Fig. 1.2 highlights the dual challenges inherent in traditional PHM paradigms, with particular emphasis on the complexities associated with both data-driven and physics-based approaches.



**Figure 1.2:** Dual challenges on the physics side and the data side.

Data-driven models leverage statistical and ML techniques to analyze monitoring data, facilitating the identification of patterns and the prediction of future system behaviors [6]. These physics-agnostic models excel when information-rich data are available, enabling the capture of intricate relationships in degradation. In contrast, physics-based models (PBMs) are grounded in the fundamental principles governing system behaviour, such as physics, chemistry, and engineering mechanics [5, 7, 8], offering insights into failure mechanisms and allowing for interpretable diagnostics and prognostics. These huge-data insensitive models are particularly effective in scenarios with substantial mechanistic research and robust first-principles modelling, providing precise analytical representations of degradation processes.

However, **we posit that the PHM domain occupies a nuanced position**

**within the continuum of modelling approaches, residing in an “intermediate” region characterized by “Some data and Partial physics.”** This information insufficient region lies beyond the effective reach of purely PBMs and data-driven approaches.

In data space, although we are in the age of big data, the mere presence of a large volume of data does not equate to an equivalent abundance of information. PHM models often face insufficient information challenges from **“sparse and noise data” problems**. These issues arise from the difficulties in complete degradation process data acquisition, the limitations of measurement systems, and the high cost of data processing, cleaning, and labelling [9]. They result in the problems of the imbalance in health-failure data, the scarcity of run-to-failure records, the lack of data labels, and data missing [10].

In the physics space, we identify insufficient information issues as **“scarce knowledge”**. This scarcity emerges from barriers in interdisciplinary knowledge, unknown failure mechanisms, and the computational complexities of physical models [11]. These factors limit our understanding of the degradation process and obscure the coupling effects within and between different systems, making it particularly difficult to establish analytical first-principle physics [12, 13]. Consequently, despite extensive analysis, modelling, and experiments of various failure cases, the development of specialized models has not led to a unified model. Instead, these models remain separate and isolated from each other, further highlighting the challenges posed by scarce knowledge in the physics space.

Inspired by the neurogenic computational model developed for studying hippocampal learning mechanisms [14], our thesis conceptualizes the PHM modelling challenges in the context of “sparse and noisy data” as well as “scarce knowledge,” as an “ill-posed” problem. The detailed mathematical modeling process of this problem is provided in Appendix A.1. The “ill-posed” problem arises when its key Hadamard’s well-posedness criteria of existence, uniqueness, and stability—collectively are not satisfied. This implies methodological shortcomings in mapping observed information to system states as shown in Fig. 1.3.

**Existence** may be compromised when data sparsity or noise results in observations falling outside the expected range, leading to inaccurate mapping. Scarce knowledge further complicates this by making it difficult to accurately define the mapping function. **Uniqueness** issues arise when multiple system states produce the same observations, causing ambiguity. This is common for sparse observations, as the same characterization is often coupled behind different faults. Stability is challenged when small changes in input cause significant fluctuations in the estimated solutions. In contrast, a **“well-posed”** mapping has a **more compact and consistent solution space with clear boundaries**, making it **robust to different scenarios**. In PHM applications, well-posed mapping represents fewer **false alarms, more accurate detections, more reliable**

### 1.3. Research directions and main contributions

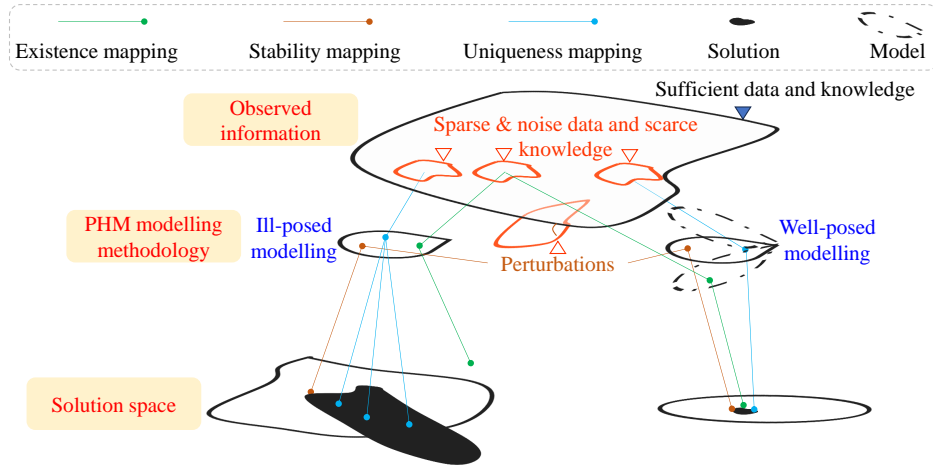


Figure 1.3: “Ill-posed” problem in PHM modelling.

#### predictions, and stronger generalization performance.

We argue that there are three critical research questions (RQ) arise in transforming an “ill-posed” mapping model into a “well-posed” one:

- **RQ1-How to refine mapping paths:** This involves exploring how to restructure the operations within ML or NN models to enhance their accuracy and strengthen their ability in prognostics and diagnostics by using the established physics as the mapping constraints.
- **RQ2-How to expand available observation states:** This explores how to expand the range of observable states, particularly by extracting valuable information from unlabelled data and transforming previously inaccessible observations into actionable insights.
- **RQ3-How to build generalized mapping family:** The focus is on how to standardize input data across these scenarios and link the model’s internal dynamics to the specific characteristics of each PHM scenario. This approach aims to develop a model that can accurately represent and respond to a wide range of industrial conditions, ensuring it remains both practical and adaptable in diverse real-world applications.

## 1.3 Research directions and main contributions

### 1.3.1 Research directions

The research questions outlined in Section 1.2 form the foundation for this thesis. To develop robust, adaptable, and accurate PHM models, ensuring practical

applicability, we explore the following research directions:

- **Constructing physics-informed machine learning (PIML) based hybrid models using PBMs and data-driven approaches to refine the “mapping path”**. This approach starts by analyzing the system’s physical principles to identify key elements that can be turned into mathematical constraints. These constraints are then incorporated into the model’s optimization process during training. By aligning data processing with these physical laws, the model becomes better at accurately reflecting real-world system behaviours. Restructuring the model’s operations ensures that machine learning or neural networks are not solely data-driven but are also guided by established physics, resulting in a model that can more effectively predict and diagnose system conditions across various scenarios.
- **Expanding the range of observable states with self-supervised learning (SSL) techniques**. SSL allows the model to extract useful information from unlabelled data, transforming previously inaccessible observations into actionable insights. This path involves designing learning tasks that convert unlabeled data into formats that the PHM model can effectively use. By expanding the available observation states, the model can more accurately map the relationship between different system states and observed data. This allows the model to utilize vast amounts of unlabelled data, thereby increasing its robustness and applicability across diverse conditions, even in scenarios where labelled data is scarce.
- **Developing a generic and flexible PHM framework based on physics-informed SSL**. The proposed framework optimizes the synergy between different components, ensuring they work together effectively to achieve desired outcomes. By integrating physical constraints with relevant information extracted from unlabeled data, this approach enhances the precision and robustness of the model’s solutions. Furthermore, a key aspect of this research path is the development of a robust pretraining and local fine-tuning procedure. This helps the model stay stable and perform well across different scenario in real-world situations.

#### 1.3.2 Main contributions

Building upon the research directions outlined in Section 1.3.1, this thesis makes several key contributions that advance the field of PHM. It addresses both scientific and practical challenges through the development of innovative modelling approaches, validated across diverse case studies.

- 1. Developing the innovative PIML models for PHM.** One of the most significant scientific contributions of this thesis is the introduction of new PIML models specifically tailored for PHM applications. These models integrate physical principles with machine learning techniques, enhancing the model's ability to predict and diagnose system conditions with higher accuracy and reliability. For instance, the novel PIML model inspired by the "Mimetic Theory" offers a unique approach to diagnostics, effectively validated in the rotor compound faults case study. Additionally, the development of a new PIML training strategy based on the "Constraint Projection" theory addresses the challenge of few-shot diagnostics, further reinforcing the scientific novelty of this work. These innovations highlight the potential of PIML to bridge the gap between theoretical physics and data-driven modelling, contributing new knowledge to the field.
- 2. Advancements in SSL for PHM.** The thesis also contributes to the scientific community by advancing SSL techniques tailored for PHM. The introduction of contrastive SSL strategies for prognostics represents a novel approach to handling unlabelled data, a common challenge in PHM applications. By improving the model's ability to utilize unlabelled data effectively, these contributions broaden the range of observable states and enhance the robustness of PHM models. The innovative application of SSL strategies, particularly those that consider downstream information, demonstrates a significant leap forward in the field, offering new methodologies for data-driven prognostics.
- 3. Development of a generic and flexible PHM framework.** A key scientific achievement of this thesis is the creation of a generic PHM framework that synergizes PIML with SSL. This framework is designed to be highly adaptable, and capable of discovering knowledge from both labeled and unlabeled data. By extending the PIML concept to actively uncover insights from unlabeled data and enhancing SSL to simultaneously exploit labeled and unlabeled data, this work sets a new standard for flexible and generalizable PHM models. The framework's validation across various prognostic scenarios, including bearings, batteries, tool wear, and CFRP fatigue, underscores its scientific originality and potential for broad application.
- 4. Demonstrating the scalability and adaptability of the proposed PHM framework across diverse industrial applications.** The generic PHM framework developed in this thesis represents a major practical advancement, offering a scalable and adaptive solution for a wide range of industrial applications. The framework's ability to dynamically adjust to different scenarios without the need for extensive parameter tuning makes it an

attractive option for real-world deployment. By reducing computational complexity and enhancing model efficiency, this framework addresses key challenges in PHM, such as resource constraints and the need for fast, reliable predictions.

**Table 1.1:** “4Cs” requirements.

4Cs: Cross-machine & devices, Cross-monitoring timescales, Cross-physical measurements, and Cross-prediction targets












<b>Cross-scenario</b>	<b>Description</b>	<b>Requirements</b>
Machine	Unified model for different machine types	Adapt knowledge across machine types, and implement flexible input architectures for varying sensor configurations.
Monitoring timescales	Handling various degradation modes and faults across time scales	Integrating short-term and long-term memory mechanisms, and implementing adaptive strategies for different time scales.
Physical measurements	Combining diverse data types from industrial systems	Developing robust sensor fusion techniques, and implementing self-adaptive feature extraction methods.
Prediction targets	Accommodating various PHM targets	Handle diverse physical meanings of targets, adapt to different trend patterns and accommodate varying scale scopes of prediction targets.

From an engineering perspective, the proposed generic PHM framework is designed to effectively adapt to the “4Cs” requirements, presented in Table 1.1. These “4Cs” requirements represent the critical ability of the framework to handle diverse cross-scenarios, including different machine types, various monitoring timescales, a range of physical measurements, and multiple prediction targets. Specifically, the framework must be capable of adapting knowledge across different machine types, implementing flexible input architectures for varying sensor configurations, and integrating short-term and long-term memory mechanisms to handle degradation modes and faults across different time scales. Additionally, it requires robust sensor fusion techniques and self-adaptive feature extraction methods to combine diverse data types from industrial systems. Finally, the framework must be versatile enough to accommodate various PHM

targets, adapting to different trend patterns and prediction scopes. This adaptability ensures that the framework can effectively address the complexities and variabilities inherent in real-world industrial applications.

## 1.4 Thesis overview

Fig. 1.4 presents an overview of the thesis structure. It is organized to progressively build upon the foundational concepts of PIML and SSL, leading to the development of a comprehensive and adaptable PHM framework.

Contributions	Validations	Publications
 <b>Chapter II</b> <b>Comprehensive bibliometric analysis and review on:</b> <ul style="list-style-type: none"> <li>• Physic-informed machine learning in PHM (PIML)</li> <li>• Self-supervised learning in PHM (SSL)</li> </ul>	 <b>V1:</b> Bearing RUL prediction	<b>J1, J5</b>
 <b>Chapter III</b> <b>Development of effective PIML approaches in PHM:</b> <ul style="list-style-type: none"> <li>• New PIML model based on the “Mimetic theory” for diagnostics (V2)</li> <li>• New PIML training strategy based on the “Constraint projection” theory for few-shot diagnostics (V2)</li> <li>• Novel generic PIML framework for prognostics (V3)</li> </ul>	 <b>V2:</b> Rotor compound faults diagnostics	<b>J2, J4, C1, C4, C5</b>
 <b>Chapter IV</b> <b>Development of effective SSL approaches in PHM:</b> <ul style="list-style-type: none"> <li>• New contrastive SSL strategies for prognostics (V1)</li> <li>• Improved SSL strategies considering downstream information for prognostics (V1)</li> </ul>	 <b>V3:</b> Battery RUL prediction	<b>J5, C3</b>
 <b>Chapter V</b> <b>Development of generic PHM framework based on physics-informed SSL:</b> <ul style="list-style-type: none"> <li>• Extending PIML concept to actively discover knowledge on unlabeled data (V6)</li> <li>• Enhancing SSL concept to simultaneously exploit label and unlabeled data</li> <li>• Constructing generic end-to-end prognostics model across different PHM scenarios by combing PIML and SSL (V1, V3, V4, V5)</li> </ul>	 <b>V4:</b> Tool wear prediction  <b>V5:</b> CFRP fatigue prediction	<b>J3, C2</b>
 <b>Chapter VI</b> <b>Conclusion and future work:</b> <ul style="list-style-type: none"> <li>• Summary of our contributions</li> <li>• Discussion of thesis perspectives</li> </ul>	 <b>V6:</b> Monitoring and inverse dynamics identification	<b>—</b>

**Figure 1.4:** Structure and main contributions of our thesis.

Chapter 2 presents a comprehensive bibliometric analysis and review conducted on two emerging areas in PHM: PIML and SSL. This review synthesizes existing knowledge, identifying key trends, challenges, and opportunities in these fields. The insights gained from this analysis serve as a basis for the subsequent development of novel PHM methodologies, guiding the integration of these techniques into effective PHM models.

Building on the insights from the literature review, Chapter 3 introduces new PIML approaches tailored for PHM applications. These include:

- A novel PIML model inspired by the “Mimetic Theory” for diagnostics, which has been validated through the rotor compound faults diagnostics case study (V2).



- A new PIML training strategy based on the “Constraint Projection” theory, specifically designed for few-shot diagnostics, also validated on the rotor compound faults diagnostics (V2).
- A generic PIML framework for prognostics, demonstrated through its application to battery RUL prediction (V3).

In Chapter 4, the thesis delves into the application of the “Global pretraining-local finetuning” procedure within SSL to expand the range of observable states, a crucial aspect of PHM modelling. This chapter introduces improved SSL strategies, particularly through new contrastive learning approaches, providing both theoretical advancements and practical demonstrations. The efficacy of these strategies is showcased through a case study focused on bearing RUL predictions, highlighting the successful implementation of directional representation learning.

Building on the methodologies introduced in previous chapters, Chapter 5 extends the conventional concepts of PIML and SSL to propose a comprehensive physics-informed SSL methodology. This chapter presents a unified, generic prognostics model designed to be applicable across a variety of degradation scenarios. The model is specifically tailored to complete degradation predictions within complex “4Cs” scenarios, achieving these outcomes with a lightweight and computationally efficient model. This chapter thus represents a significant step towards a more versatile and generalizable PHM framework.

Chapter 6 offers a comprehensive summary and forward-looking perspective on the entire thesis. It begins by synthesizing the key contributions made to the field of PHM through the integration of advanced PIML and SSL techniques. The chapter then highlights the successful application of the proposed models across various PHM scenarios, demonstrating their practical impact and versatility. Finally, it discusses future perspectives, outlining the potential of these approaches to meet the critical and evolving requirements of PHM systems, and paving the way for further advancements in the field.

## 1.5 List of publications and awards

### 1.5.1 Publications

- J1 **Weikun DENG**, Khanh T.P. NGUYEN, Kamal MEDJACHER, Christian GOGU, Jérôme MORIO (2023). Physics-informed machine learning in prognostics

- and health management: State of the art and challenges. *Applied Mathematical Modelling*, 124: 325-352.
- J2 **Weikun DENG**, Khanh T.P. NGUYEN, Kamal MEDJACHER, Christian Gogu, Jérôme MORIO (2023). Rotor dynamics informed deep learning for detection, identification, and localization of shaft crack and unbalance defects. *Advanced Engineering Informatics*.
- J3 **Weikun DENG**, Ardiani F, Khanh T.P. NGUYEN, Benoussaad M, Kamal MEDJACHER (2024). Physics informed machine learning model for inverse dynamics in robotic manipulators. *Applied Soft Computing*, 111877.
- J4 **Weikun DENG**, Hung LE, Christian GOGU, Khanh T.P. NGUYEN, Kamal MEDJACHER, Jérôme MORIO, Dazhong WU(2024). Generic Physics-Informed Machine Learning Framework for Battery Remaining Useful Life Prediction Using Small Early-Stage Lifecycle Data. *Available at SSRN 4770354* (Submitted to *Applied Energy*, Finished the revision, waiting for the Editor's decision).
- J5 **Weikun DENG**, Khanh T.P. NGUYEN, Kamal MEDJACHER, Christian GOGU, Jérôme MORIO (2024). Enhancing Prognostics for Sparse Labeled Data Using Advanced Contrastive Self-Supervised Learning with Downstream Integration. (Submitted to *Engineering Applications of Artificial Intelligence*, Finished the revision, Accepted).
- C1 **Weikun DENG**, Khanh T.P. NGUYEN, Christian GOGU, Jérôme MORIO, Kamal MEDJACHER (2022). Physics-informed lightweight temporal convolution networks for fault prognostics associated to bearing stiffness degradation. *PHM Society European Conference*, 7(1): 118-125.
- C2 **Weikun DENG**, Khanh T.P. NGUYEN, Kamal MEDJACHER (2022). Physics Informed Self Supervised Learning For Fault Diagnostics and Prognostics in the Context of Sparse and Noisy Data. *PHM Society European Conference*, 7(1): 574-576.
- C3 **Weikun DENG**, Khanh T.P. NGUYEN, Kamal MEDJACHER, Christian GOGU, Jérôme MORIO (2023). Bearings RUL prediction based on contrastive self-supervised learning. *IFAC-SectionsOnLine*, 56(2): 11906-11911.
- C4 **Weikun DENG**, Khanh T.P. NGUYEN, Christian GOGU, Jérôme MORIO, Kamal MEDJACHER (2023). A Few-Shot Learning Framework for Rotor Unbalance and Shaft Crack Fault Diagnostic Based on Physics-Informed Neural Network. *Structural Health Monitoring 2023*.

C5 **Weikun DENG**, Khanh T.P. NGUYEN, Christian GOGU, Jérôme MORIO, Kamal MEDJACHER, Hung LE, Dazhong WU (2024). A Novel PIML Architecture with Innovative Learning Paradigm Applied in Battery Prognostics. *CODIT 2024*.

### 1.5.2 Awards

- A1 Second best thesis prize at the doctoral symposium in the European Conference of the Prognostics and Health Management Society 2022 (PHMe22), Turin, Italy.
- A2 Best application Section final list on the 22nd World Congress of the International Federation of Automatic Control (IFAC 2023), Yokohama, JAPAN.
- A3 Chinese government award for outstanding self-financed students abroad: The award is open to all second-year and above doctoral students and post-doctoral Chinese researchers worldwide who are not funded by the CSC, and 650 recipients receive the award each year.

# Literature review

---

## Contents

<b>2.1</b>	<b>Introduction</b>	<b>18</b>
<b>2.2</b>	<b>Bibliometric literature analysis and quantitative review</b>	<b>19</b>
2.2.1	Research trend analysis	20
2.2.1.1	Evolution of hybrid approaches	20
2.2.1.2	Evolution of unlabelled data learning	21
2.2.2	Statistical analysis of PIML and SSL in PHM	22
<b>2.3</b>	<b>Qualitative literature review</b>	<b>24</b>
2.3.1	An overview of PIML in PHM	24
2.3.1.1	Knowledge informed methods	24
2.3.1.2	Informed knowledge types	36
2.3.2	An overview of SSL in PHM	38
2.3.3	Focus of existing doctoral theses	44
<b>2.4</b>	<b>Summary</b>	<b>46</b>

---

This chapter is based on the following publications:

- J1 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2023). Physics-informed machine learning in prognostics and health management: State of the art and challenges. *Applied Mathematical Modelling*, 124: 325-352.
- J5 **Weikun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2024). Enhancing Prognostics for Sparse Labeled Data Using Advanced Contrastive Self-Supervised Learning with Downstream Integration. (Submitted to *Engineering Applications of Artificial Intelligence*, Accepted).

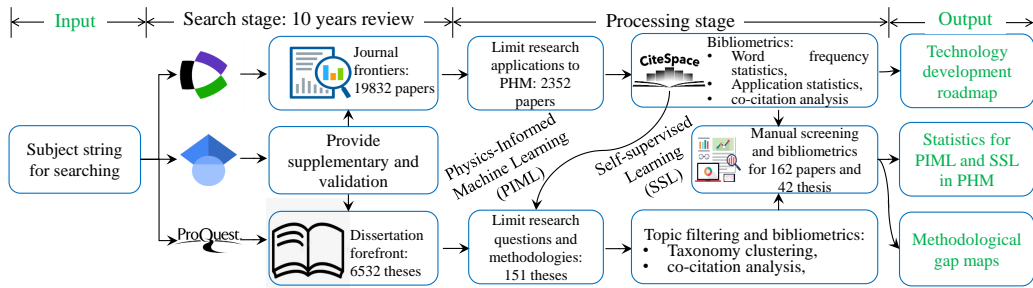
## 2.1 Introduction

This chapter has 4 sections. They provide a critical analysis of the state-of-the-art (SOTA) for constructing hybrid models and learning from unlabelled data. The discussion is framed from both qualitative and quantitative perspectives, focusing on research directions aimed at addressing the “ill-posed” problem within the context of “Sparse, noise data and scarce knowledge” introduced in Chapter 1.

Section 2.2 begins with a bibliometric analysis with a research roadmap survey detailed in Section 2.2.1. The analysis of the technological roadmap indicates that physics-informed machine learning (PIML) and self-supervised learning (SSL) represent the leading edge and foundational solutions in hybrid modeling and unlabelled data learning, respectively. Section 2.2.2 presents quantitative statistical analyses of PHM applications involving PIML and SSL, while Section 2.3 offers a systematic qualitative review of relevant PHM cases and theoretical developments. Finally, Section 2.4 provides a scientific positioning map that situates our study within the broader research landscape.

## 2.2 Bibliometric literature analysis and quantitative review

The entire bibliographic analysis procedure is presented in Fig.2.1.



**Figure 2.1:** Schematic flow of the bibliometric analysis.

The dataset analyzed covers the period from January 2014 to June 2024, obtained from Web of Science (WoS) and ProQuest, including technical papers, reports, and dissertations. To ensure completeness, Google Scholar was utilized to verify, complement, and de-duplicate the search results from WoS and ProQuest. Based on the search results, a subsequent co-citation analysis is performed using the CiteSpace tool [15]. This analysis reveals the frequency of prominent keywords, as discussed in Section 2.2.1, and helps identify prevailing research trends, mapping the technological evolution of hybrid models and unlabelled data learning. In Section 2.2.2, the thesis further undertakes a quantitative analysis of PIML and SSL, providing a systematic investigation into existing gaps in their application to PHM.

### 2.2.1 Research trend analysis

Technology evolution trends are identified through the clustering of keywords across yearly slices, offering insights into how specific research areas have developed over time. The results of this clustering are visually depicted in Fig. 2.2 and Fig. 2.3, illustrating the progression of technology trends concerning PIML and SSL.

#### 2.2.1.1 Evolution of hybrid approaches

Fig. 2.2 illustrates the evolving focus of keywords in research of hybrid approaches. Major focuses include anomaly detection, RUL estimation, and process optimization, all crucial for effective PHM applications. The proposal of “Physics In-

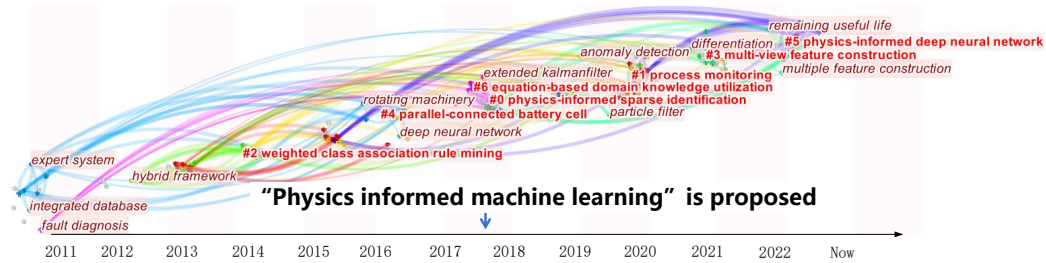


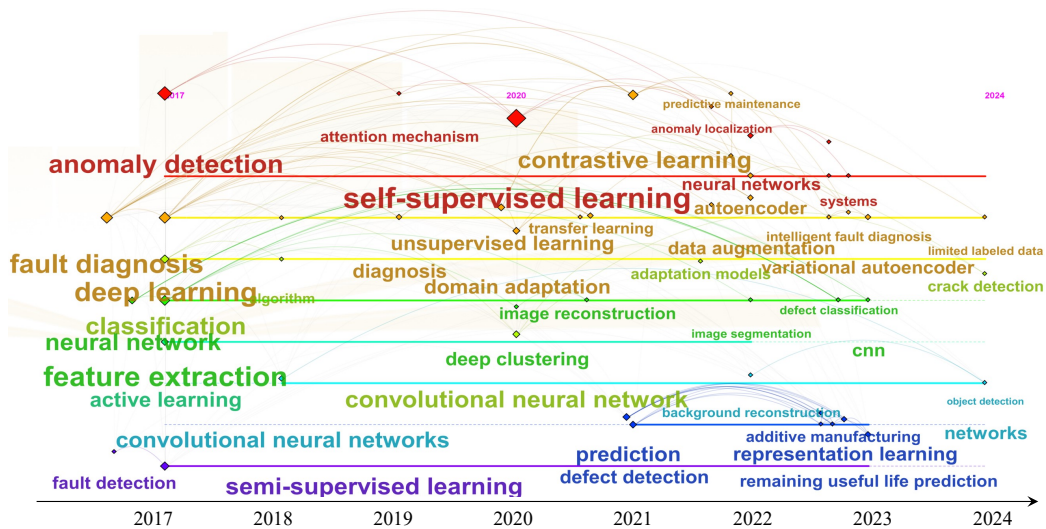
Figure 2.2: A technological roadmap for hybrid model architectures.

formed Machine Learning” in 2018 marked a pivotal methodologies advancement [16], catalyzing the development of physics-informed sparse identification [17], and physics-informed deep neural networks [18]. This transition was initially focused on expert systems and progressed to weighted class association rule mining by 2014. The shift towards Physics-Informed Neural Networks (PINN) [19] highlights the necessity of integrating physical principles directly into the ML pipeline rather than merely combining physics and ML in the input or output space. In particular, “Deep neural networks” are frequently leveraged to establish sophisticated PIML frameworks. This integration is driven by enhancing complex systems’ model accuracy, reliability, and interpretability [20]. Integrating equation-based domain knowledge underscores the critical role of leveraging domain-specific physical laws in ML models, ensuring predictions are data-driven and aligned with established scientific principles. There is a notable increase in research incorporating differential equations modeling lifetime degradation processes into machine learning models, as shown by the clustered hot words in the figure.

*This analysis motivates our research to focus on PINN and to address the questions “What form of physics to embed” and “How to embed physics knowledge.”*

### 2.2.1.2 Evolution of unlabelled data learning

As depicted in Fig. 2.3, initially, unsupervised learning and deep clustering techniques enhanced the identification of patterns and anomalies without labeled data. Subsequently, the focus shifted towards SSL, especially the contrastive learning-based paradigm, leveraging data's hidden pattern and structure for advanced anomaly detection and domain adaptation. Integrating neural networks, particularly autoencoders and variational autoencoders, facilitated complex tasks such as image-like monitoring data reconstruction and data augmentation, which are crucial for predictive maintenance and intelligent fault diagnostics. Recent developments have seen the emergence of representation learning, which improves model robustness and accuracy using limited labeled data.



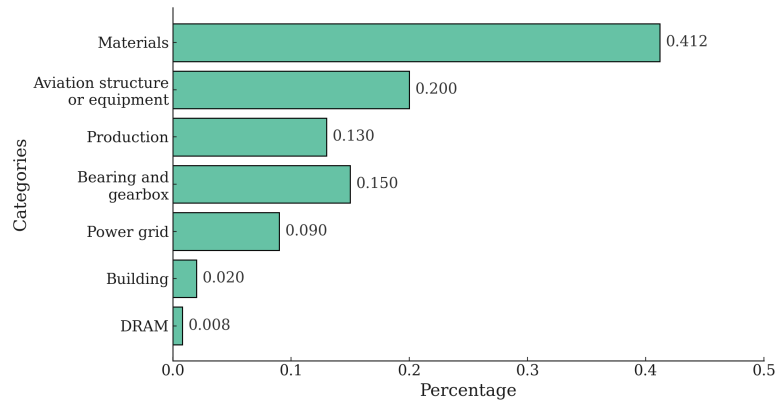
**Figure 2.3:** A technological roadmap for unlabelled data learning.

It is important to highlight that semi-supervised learning was another prominent technique in the early stages of development. In our view, SSL is more suitable for PHM applications as it can effectively learn from large volumes of unlabeled sensor data without the need for costly manual annotation during the feature learning phase. SSL leverages unlabeled data to create supervised signals intrinsically, resulting in more generic learned features. These advancements are applied in specific PHM areas such as crack detection, defect classification, and RUL prediction.

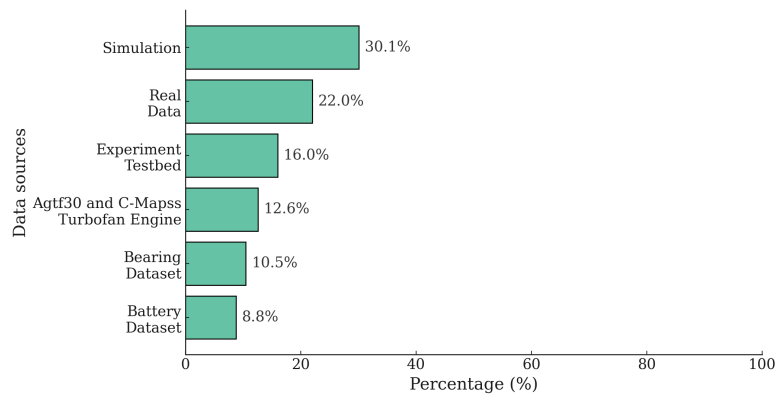
*Consequently, our thesis is driven by the need to develop robust SSL model frameworks for PHM. It aims to explore novel applications with a focus on data structures and knowledge mining. Additionally, it seeks to address the underdeveloped theoretical foundations of SSL.*

### 2.2.2 Statistical analysis of PIML and SSL in PHM

This section provides a comprehensive analysis of key statistics for PIML and SSL in PHM, examining three principal aspects: application domains, PHM tasks, and monitoring signals.



(a) Application cases.



(b) Data sources.

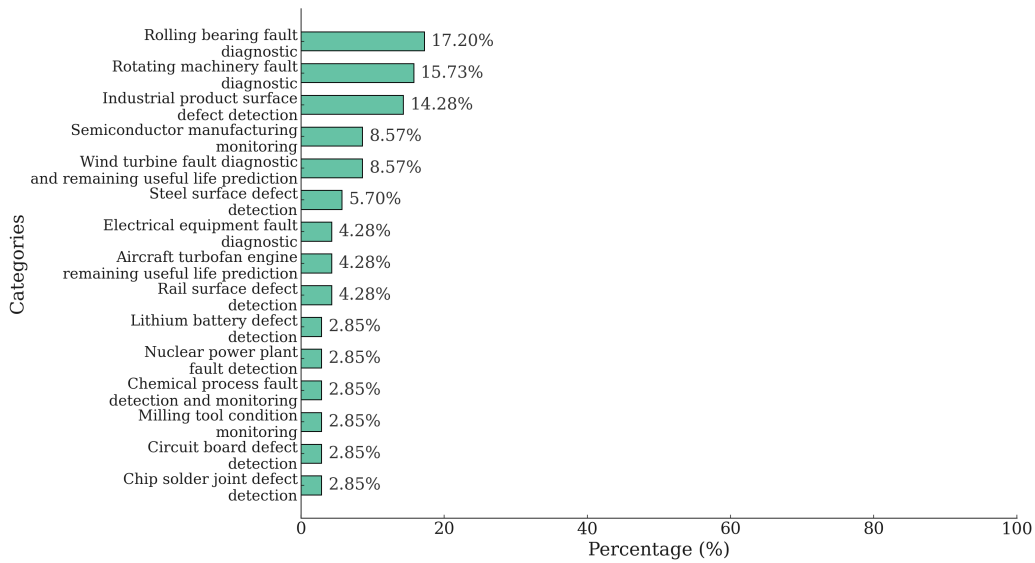
**Figure 2.4:** Main application areas and data sources of PIML in PHM.

Fig. 2.4 presents the distribution of application areas and data sources of the studies on PIML in PHM. From Fig. 2.4 (a), one can see that most of the current PIML studies in PHM focus on material damage (41.2%) because there already exists in this area numerous studies in mathematical and physical modeling of material dynamic behaviors. These studies provide a solid foundation for the rapid development of PIML models. Other applications such as aviation structure and equipment (20.0%), production equipment (13.0%), bearing and gearbox (15.0%), and power grid (9%) have also attracted more attention from the research community in recent years. Besides, considering data sources Fig. 2.4 (b), we find



## 2.2. Bibliometric literature analysis and quantitative review

that most data sources for PIML studies come from simulation (30%). Also, the most used bench-marking datasets are the Turbo engine simulation dataset (C-MAPSS and AGTF30) and battery dataset (Oxford and NASA). The studies of PIML models for real systems are limited to small experimental platforms (16%). Those observations can be explained by the lack of exploitable physics-based knowledge of real systems that are usually difficult to model. Especially, a large proportion of PIML research focuses on solving the PHM tasks in the presence of sparse (26%) or noisy data (38%).



**Figure 2.5:** Main application areas and data sources of SSL in PHM.

As shown in Fig. 2.5, one can see that SSL finds significant applications across various industrial domains. The rolling bearing fault diagnostics counts the major part around 11.7%, followed closely by the rotating machinery fault diagnostics at 10.7%. SSL also plays a crucial role in industrial product surface defect detection, with a 9.7% usage. Other notable applications include semiconductor manufacturing monitoring and wind turbine fault diagnostics, each at 5.8%, steel surface defect detection at 3.9%, and electrical equipment fault diagnostics and aircraft turbofan engine life prediction, both at 2.9%. These diverse applications underscore SSL importance in enhancing industry life cycle management, predictive maintenance, and operation safety. In “others”, they include studies with a share of less than 2%, including Centrifugal pump fault diagnostics, air conditioning system fault detection, and so on. Additionally, existing studies relating application of PIML and SSL in PHM are synthesized in Tables B.1 and B.2 of Appendix B.1.

In summary, the statistical analysis reveals that PIML and SSL techniques are predominantly applied in diagnostics and detection studies, with relatively few applications in prognostics. Additionally, there is a noticeable focus on single-scenario case studies, with a significant lack of research addressing cross-scenario applications. These findings underscore *the need for further exploration of prognostic scenarios and the development of models capable of handling cross-scenario studies, which serves as a key motivation for this thesis.*

## 2.3 Qualitative literature review

Section 2.3.1 and Section 2.3.2 provide a comprehensive qualitative literature review, focusing on the advancements and PHM applications of PIML and SSL. Through reviewing the key methodologies, innovative approaches, and challenges, we aim to elucidate the existing landscape and identify opportunities for future research in these emerging fields. In particular, a comparative analysis of existing PhD dissertation, extending beyond the scope of PHM, is presented in Section 2.3.3 to underscore the scientific contributions of this research to PIML and SSL topics.

### 2.3.1 An overview of PIML in PHM

Compared to the existing review papers that we give the introduction in Appendix B.2, our review offers a novel and comprehensive overview of PIML specifically tailored to PHM. We explore this topic from two critical perspectives: “how to inform different types of knowledge” (see Section 2.3.1.1) and “what kind of knowledge can be informed” (see Section 2.3.1.2). This dual focus not only fills a gap in the existing literature but also provides a deeper understanding of the integration of PIML within PHM, making our work one of the first to thoroughly examine PIML in this specific context.

#### 2.3.1.1 Knowledge informed methods

The PIML methods in our research are classified into three categories: Physics-informed input space, physics-embedded algorithm structure, and physics constrained learning, as shown in Fig. 2.6. *This division depends on the role of physical knowledge and its informed position in the ML pipeline.*

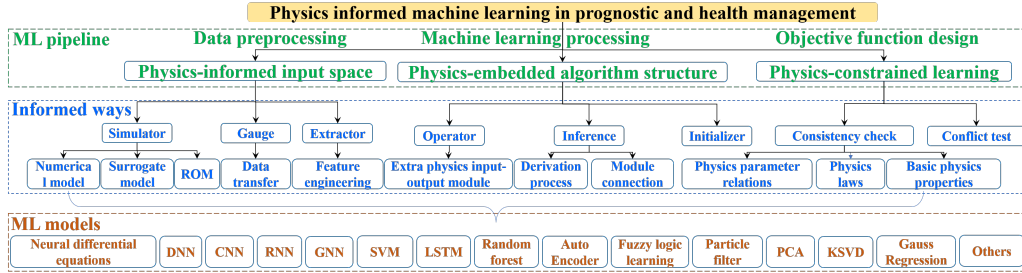


Figure 2.6: Taxonomy of existing PIML methods in PHM.

**1) Physics-informed input space.** This category seeks to gain physics information in the ML input space, distilling the multi-sources and heterogeneous monitoring data [21, 22] by assisting data augmentation, feature transformation, feature selection, dimensionality reduction [23], and information fusion [24].

“Physics-informed input space” can be seen as an extension of the traditional “feature engineering” or “simulation-based data augmentation” processes. We have summarized three major paradigms: “Simulator”, “Gauge”, and “Extractor.” shown in Table. 2.1.

Table 2.1: Summary of physics-informed input space studies in PHM.

Ref.	Application	Knowledge source	Informed ML model	PHM tasks
[25, 26]	Aeronautical structure	Component-based digital twin	Simulator	Classification tree and SVM
[27]	Triplex pump	Component-based digital twin	Simulator	Auto-encoder transfer learning
[28]	Oil productionline	Production-based digital twin model	Simulator	Autoencoder & LSTM
[29]	Rotor	A priori evaluation of feature space separability of loads	Simulator	Hamiltonian auto-encoder NN, PCA, & random forest

Continued on next page

Table 2.1 – continued from previous page

Ref.	Application	Knowledge source	Informed ML model	PHM tasks	
[30]	Electro-Hydrostatic Actuator degradation	Physical degradation model	Simulator LSTM	Degradation prediction	
[16]	Tubofan engine	Engine air path performance model	Simulator DNN	RUL prediction	
[31, 32]	Composite structure	Bonded joints fatigue FE or lattice surrogate model	Simulator FCN	Fatigue prediction	
[33]	Bearing	Time domain statistical feature generation model	Simulator SVM	Diagnostics	
[34]	Aircraft composite structure	A numerical solutions of Lamb waves	Simulator CNN	Diagnostics	
[35]	Industrial production	Time-series derivative weighting for perturbation values	Simulator VAE	Diagnostics	
[36]	Building	Invariable characteristics of building structure	Gauge	Physics-informed multi-source domain adversarial networks	Diagnostics
[37]	Additive manufacturing monitoring	Geometry invariant in thermal history features and trend	Gauge	Tree-based regression	Condition monitoring

Continued on next page

**Table 2.1 – continued from previous page**

Ref.	Application	Knowledge source	Informed ML model	PHM tasks	
[38]	Gearbox	Implicit physical association between unlabeled and labeled data	Gauge	Deep convolutional generative adversarial network	Diagnostics
[39]	Gearbox	Vibration inherent cyclostationary characteristics	Extractor	Autoencoder	Diagnostics
[40]	Bandsaw	Vibration modal analysis and finite element analysis	Extractor	PINN and DCNN	Diagnostics
[41]	Gearbox	Health-adaptive physics time-scale representation embedded input module	Extractor	CNN	Diagnostics
[42]	Electro-mechanical load	Feature space load separability prior evaluating Dynamic Mode Decomposition	Extractor	SVM and DNN	Diagnostics
[43]	First Order Plant with Time Delay system	with control and continuous wavelet transform based system fault feature picture	Extractor	Transferring DNN	
[43]	Air handling units	Importance feature selection based on the semantics of the physical model	Extractor	issenstein generative adversarial network	Diagnostics

According to the summary in Table. 2.1, “simulators” generate data across different health states and system behaviors, enhancing training information. This

requires data synthesis models that balance computational cost and fidelity, including but not limited to using digital twins [25], reduced order models [26], or physics-based numerical simulations [27]. “Gauges” addresses the limitations of simplified simulators by designing transfer criteria. It is concerned with building invariant features or implicit physical associations in data into suitable metrics, as illustrated in building structure diagnostics [36] and additive manufacturing monitoring [37]. “Extractors” guide data preprocessing with physics knowledge to ensure fault physics-related features are included, such as in gearbox diagnostics [39] and bandsaw diagnostics [40].

**2) Physics-embedded algorithm structure.** Regarding “Physics-embedded algorithm structure”, PIML seeks to make the traditional physics-agnostic ML become physics-aware so that the governing processes are added to the design of ML algorithm structures and the parameters searching process. It is prone to integrate the “Hard Constraint Projections (HCP)” [44] with ML, including the three following paradigms: “Basic operator”, “ML Structure blueprint”, and “Parameter initializer,” as shown in Table 2.2.

**Table 2.2:** Summary of physics-embedded algorithm structure in PHM.

Ref.	Application	Knowledge source	Informed ML model		PHM tasks
[45]	Crack growth and filter clogging	Paris laws for fatigue crack and pressure drop analog formula	Operator	ANN	RUL prediction
[20]	Wind turbine	Pairs laws for fatigue crack and pressure drop analog formula	Operator	Wavelet-based feature engineering in the Particle filter framework	Fault diagnostics and RUL prediction

Continued on next page

Table 2.2 – continued from previous page

Ref.	Application	Knowledge source	Informed ML model	PHM tasks
[46]	Lube oil	Degradation model incorporating the shift and diffusion coefficient parameters	Operator	Hybrid data augmentation based on State space model in the particle filter framework Degradation prediction
[47]	Lithium-ion battery	Predict future capacity measurements based on the similarity of capacities in the historical data	Operator	Particle filter Degradation prediction
[48]	Motor bar broken	Fault frequency and square envelope threshold	Operator	CNN Fault diagnostics
[49]	Drill pipe	Embedding hydraulic coefficient relationship between two DNNs	Operator	DNN
[50, 51]	Bearing	Dynamic wavelet or FFT informed layer	Operator	CNN&Resnet Fault diagnostics
[52]	Lithium-ion battery battery	Reduced-order model based on Nernst and Butler–Volmer equations	Operator	RNN RUL prediction
[53]	Tool wear	Sipos empirical wear-time	Operator	Adaptive neuro-fuzzy inference system Degradation prediction

Continued on next page

Table 2.2 – continued from previous page

Ref.	Application	Knowledge source	Informed ML model		PHM tasks
[54, 55]	Material defect	Topology of wave-guided electromagnetic acoustic sensor systems	Operator	Siamese CNN	Fault diagnostics
[56]	Mortar cube crack prediction	Non-linear fracture amplitude modes	Operator	Dynamic mode decomposition	Degradation prediction
[57]	Bearing fatigue	Paris-laws based corrosion	Operator	NN	
[58, 59]	Structure crack	Damage differential equations & Dirichlet boundary based growth laws	Structure blueprint	DeepONet	Degradation prediction
[60, 61]	Crack identification	Differential equation for crack extension	Structure blueprint	Stacked auto-encoder	Degradation prediction
[62, 63, 64, 65]	Aviation structure crack	Crack extension or vibration anomaly models	Structure blueprint	RNN	Degradation prediction
[66, 67, 68]	Aviation structure crack	Implicit recursive structure of cumulative bearing damage, crack extension or vibration anomaly models	Structure blueprint	RNN	
[68, 67, 63]	Structure fatigue	Eulerian integration for fatigue crack extension	Structure blueprint	RNN or CNN	Degradation prediction

Continued on next page



Table 2.2 – continued from previous page

Ref.	Application	Knowledge source	Informed ML model		PHM tasks
[52]	Batteries RUL prediction	Governing differential equations based on measured capacity & voltage curves	Structure blueprint	RNN	RUL prediction
[69]	Tool wear	Empirical rules for cutting temperature, speed and tool life	Structure blueprint	Logistics classifier	
[70, 71]	Structure damage	Structural changes due to damages	Structure blueprint	Stacked NODE	Fault diagnostics
[72]	Batteries	Charge losses and changes	Structure blueprint	UODE	Degradation prediction
[73, 74, 75, 76, 77]	Grid and Buses FD	Physics spatial or spectrum associativity	Structure blueprint	Graph NN	Fault diagnostics
[78]	Torsional vibration dampers	The stiffness and damping coefficients are used as nodes of NN	Structure blueprint	Custom ANN	
[79]	DRAM error	Spatial dependence of the DRAM	Initializer	SVM, NN, Boosted Trees, Naive Bayes, Random forest	Fault diagnostics
[80]	Bearing	Interpretable weights based envelope spectrum	Initializer	Supervised learning dichotomy	Fault diagnostics

Continued on next page

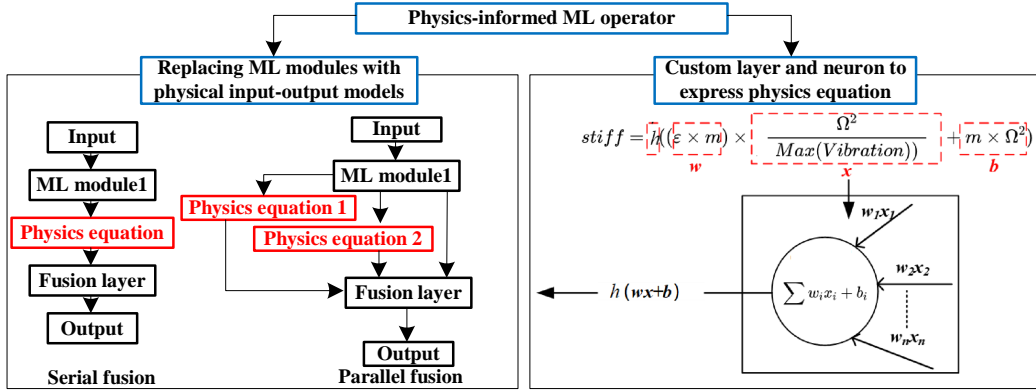
Table 2.2 – continued from previous page

Ref.	Application	Knowledge source	Informed ML model	PHM tasks	
[81]	Casting defect	One-dimensional heat transfer equation	Initializer	Non-negative matrix factorization	Condition monitoring
[82]	Materials cracks/fractures	Geomechanical alteration index cluster basis	Initializer	K-Means cluster	Fault diagnostics
[69]	Tool wear	Decision space parameterized by cutting speed and temperature	Initializer	CNN	Degradation prediction
[57]	Power grids	Wind oscillation equations and grid equations	Initializer	Gaussian Process Regression	Fault diagnostics
[83]	Offshore wind turbine	Degradation excess matrix	Initializer	Bayesian network	Degradation prediction

The “Operator” paradigm incorporates physics knowledge into ML modules to better capture input-output relationships, as shown in Fig. 3.9. It is the most commonly used and basic paradigm. Two approaches are proposed: 1) Replacing ML modules with physical input-output models, and 2) Designing custom layers and neurons to express physics equations. The first approach transforms raw data into health indicators using physically meaningful methods, such as the wavelet layer discussed in [50], while the second expresses physical functions within ML modules, as demonstrated in [18]. Both methods aim to enhance system accuracy, robustness, and interpretability by leveraging physical insights [84, 65].

The “Blueprint” paradigm focuses on identifying topological similarities and mapping unit dependencies from geometric structures, system behaviours, or internal material interactions to prioritize physical reasoning in ML model training [85, 73]. It involves tailoring neural networks according to physical laws [86], modeling potential energy functions [87], and representing dynamic behaviors through recursive structures [67]. This approach helps optimize ML training by abstracting system behaviour from physics-based models and implementing specific physical relationships within the ML structure [60, 66, 70, 71].

The “Parameters Initializers” paradigm concentrates on selecting and assigning ML parameters and hyperparameters based on physical principles. This in-



**Figure 2.7:** Two ways for embedding physics knowledge as a ML operator.

The left portion of the diagram depicts the operator formed by substituting the original ML input/output module with the entire formula, while the right portion illustrates the operator constructed by equating the ML model’s learnable parameters to the unknown parameters in the physical model.

cludes weight selection based on physical energy minimum states in Markov random fields [79], parameter initialization using solutions from physical models [20], and setting initial parameters as probability distributions derived from empirical models. By grounding parameter initialization in physical understanding, this approach aims to improve the starting point and overall performance of ML models in physics-related applications, as demonstrated in techniques like non-negative matrix factorization for casting defect monitoring [81].

**3) Physics-constrained learning.** Unlike the rigid constraints inherent in “physics-embedded algorithm structures”, PIML can also incorporate soft constraints that allow for the approximate satisfaction of physical principles via the objective function’s design. These soft constraints can manifest in various forms such as integration, differentiation, probability, and logic rules. Depending on the different physics-informed objective functions, this study categorizes “physics-constrained learning” into two paradigms: “consistency check” and “conflict test” as detailed in Table 2.3.

**Table 2.3:** Summary of Physics-constraint learning in PHM.

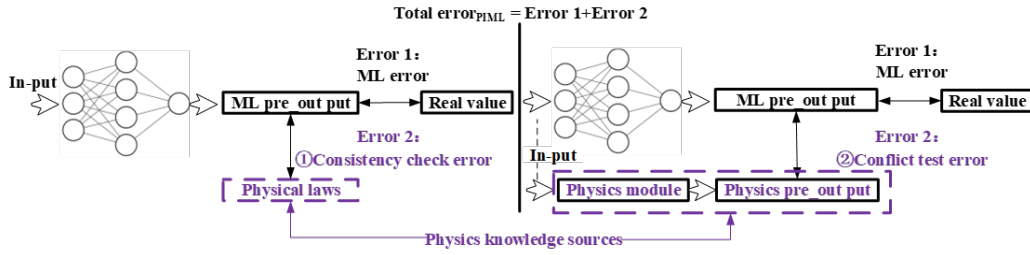
Ref.	Application	Knowledge source	Informed	ML model	PHM tasks
[88]	Turbo engine	Loss based on PDE residuals	Consistency	Stacked CNN	RUL prediction
[89]	Deformation identification	Normalized physics model's modal residual	Consistency	DNN	Fault diagnostics
[90]	Material damage	Finite Element Analysis	Consistency	DNN	Fault diagnostics
[91]	Bearing	Reliability model based on Weibull distribution property	Consistency	ANN	Fault diagnostics
[92]	Vehicle sensor	Residue generation based on transferable operators	Consistency	Neyman-Pearson test	Fault diagnostics
[86]	High impedance fault detection	Elliptic equation of rotational trajectories of the voltages and currents	Consistency	Autoencoder	Fault diagnostics
[93]	Building	Attribute-category matrix	Consistency	MatConvNet	Fault diagnostics
[94]	Ocean current turbine	Characteristics in frequency domain of the mean water flow velocity in the fan balance	Consistency	PCA and CNN	Fault diagnostics

Continued on next page

Table 2.3 – continued from previous page

Ref.	Application	Knowledge source	Informed	ML model	PHM tasks
[95]	Metal damage	Atomic update based on the regularization term of the one-dimensional wave equation	Consistency	K-SVD	Fault diagnostics
[96]	Workshop machinery	Fault frequency domain feature loss related Pearson correlation coefficient	Consistency	Deep convolutional autoencoders	Fault diagnostics
[97]	Damage stress prediction	FEM based stress distribution	Conflict	LSTM	Fault diagnostics
[98]	Bearing	Expert experience-based fault degree threshold model	Conflict	CNN	Fault diagnostics
[99]	Steel building damage	Output of a finite element model	Conflict	DNN	Fault diagnostics
[100, 101]	Wind farm & gas turbine	Physically complete historical dataset	Conflict	ANN	Fault diagnostics

In Table. 2.3, both paradigms aim to converge ML results towards physical consistency, as shown in Fig. 2.8, balancing the trade-off between data-driven learning and physics-based constraints. The total error of the PIML model includes a traditional ML prediction error (“Error1”) and a physical consistency error (“Error2”). In general, the numerical best fit to the available data (residual loss) and the consistent satisfaction of physics principles (boundary loss) show discrepancies [102].



**Figure 2.8:** Two ways to construct physics-constrained learning.

In brief, in the “consistency check” approach, the objective function is designed to ensure that the ML output conforms to physical principles. This can involve incorporating residuals from physical equations, normalized model residuals, or domain-specific knowledge directly into the loss function. Examples include using PDE residuals for turbo engine RUL prediction [88], modal residuals for deformation identification [89], and finite element analysis for material damage assessment [90].

The “conflict test” paradigm, on the other hand, builds the objective function based on the discrepancies between the ML output and the physical model output. This approach often involves comparing ML predictions with outputs from physics-based models or expert knowledge. For instance, Finite element method (FEM) based-based stress distribution was used for constraining the damage stress prediction [97], while expert experience-based fault degree threshold models were incorporated for bearing fault diagnostics [98].

### 2.3.1.2 Informed knowledge types

Physics knowledge is the prerequisite for implementing PIML. In review [103], the authors propose categorizing the knowledge sources according to their origin. **However, our thesis found the PIML implementation methods depend on the form of knowledge rather than the source of knowledge. Similar forms of knowledge may be embedded in different cases and frameworks.** For example, the proposed PIML frameworks in papers [99] and [90] come from different fields (building construction and material industry) with different knowledge sources, but both of them use the same knowledge form, i.e., finite element methods, to build the “consistency check” loss function. In addition, the exclusive reliance on physics knowledge is limited when the underlying system complexity increases. Concomitantly, the data that can accurately represent this knowledge is becoming increasingly large and complex [104].

Therefore, our thesis categorizes the physics knowledge forms into three classes, as shown in Fig. 2.9.

1. Highly condensed and universal mathematical or symbolic laws    2. The description of physical entity structure or modeling derived structure

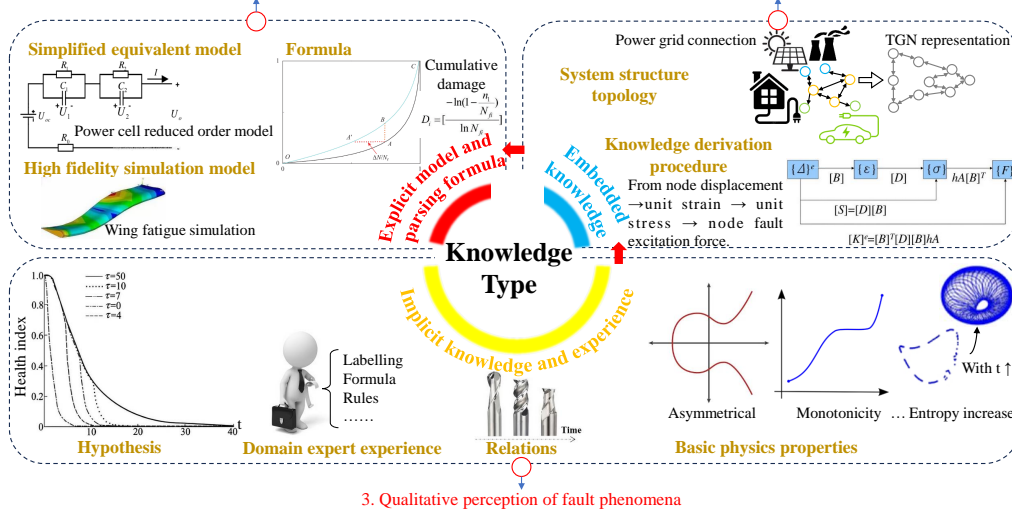


Figure 2.9: Different knowledge forms of PIML in PHM.

- First category: Explicit knowledge related to analytical failure models.** The explicit knowledge is represented by analytical models or equations of system dynamic behaviors, such as the generator of inertia constants, damping coefficients, and rotating speed in rotor dynamics [105]. They are mathematically and physically unambiguous, formal, symbolic and structured. Particularly, in PHM, they demonstrate the quantifiability of the failure processes, including algebraic, governing equations, and probabilistic relations.
- Second category: embedded knowledge related to a structure or specific process.** It is locked into the physics derivation process, system convention, structure, or layout. It provides information related to the sequence orders and the requirements of each process step or component structure. It uses ML modules to express information concerning the system structure [86], the unit dependencies [79], or the system topology framework. In particular, some physics is non-symbolic and non-explicit, merely an input-output or mutual verification relationship between the derivation procedures.
- Third category: Tacit knowledge relating wide range of physical information.** It involves hypotheses, expert rules and experiences, and also diverse underlying physical properties. It refers to the knowledge about degradation which is somewhat intuitive and difficult to quantify.

Building knowledge as constraints involves several approaches that enhance

the model’s physical consistency and interpretability. One method is explicit knowledge integration, where analytical equations are used to create simulators, extractors, or operators, ensuring that the model adheres to known physical laws. Embedding this explicit knowledge closer to the output layer improves the physical consistency of the predictions. Another approach is embedded knowledge structuring, where ML models are designed to mimic real physical processes or derivations. This involves creating structure blueprints based on physical relationships or embedding known steps as local operators within the ML framework, allowing the model to replicate the flow of information as seen in actual physical systems. Tacit knowledge transformation is also employed, where tacit knowledge is converted into a form that can be integrated into objective functions or used to design physics similarity tests and conflict loss functions. Additionally, knowledge discovery integration leverages ML models to uncover fault-related information and employs stacked architectures for knowledge discovery and validation [36, 56, 72, 106].

When choosing knowledge for different PIML frameworks, it’s crucial to consider several factors to optimize model performance and applicability. Framework compatibility is key, where the type of knowledge is matched to the structure and capabilities of the PIML framework. For example, neural network-based frameworks may benefit from embedded knowledge, while probabilistic frameworks may require explicit knowledge represented as prior distributions or constraints. The problem domain and data availability also guide the selection process; explicit and embedded knowledge is prioritized in data-scarce scenarios, while tacit knowledge discovery techniques are more applicable when data is abundant. Additionally, the interpretability requirements and computational efficiency of the model should be balanced, favoring explicit knowledge for higher interpretability and considering the computational cost of integrating different knowledge types. Other considerations include uncertainty handling, where probabilistic approaches may be needed, multi-physics integration for complex systems, and adaptability to ensure the model can evolve with new information. By carefully evaluating these factors, researchers can effectively integrate various forms of knowledge into PIML frameworks, enhancing their utility in different PHM applications [86, 57, 98].

### **2.3.2 An overview of SSL in PHM**

Unlike PIML, which incorporates external constraints, SSL focuses on mining the inherent structure or latent information within the data to generate pseudo-labels, facilitating training on unlabeled data.



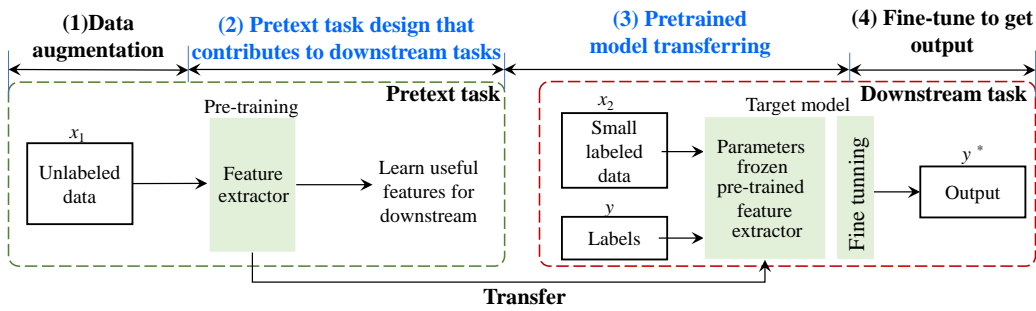


Figure 2.10: The main steps in self-supervised learning.

Typically, SSL involves a self-supervised pretext task followed by a fine-tuning task aimed at accurately predicting diagnostics or prognostic outcomes, particularly utilizing features learned during the pretext phase, as shown in Fig. 2.10.

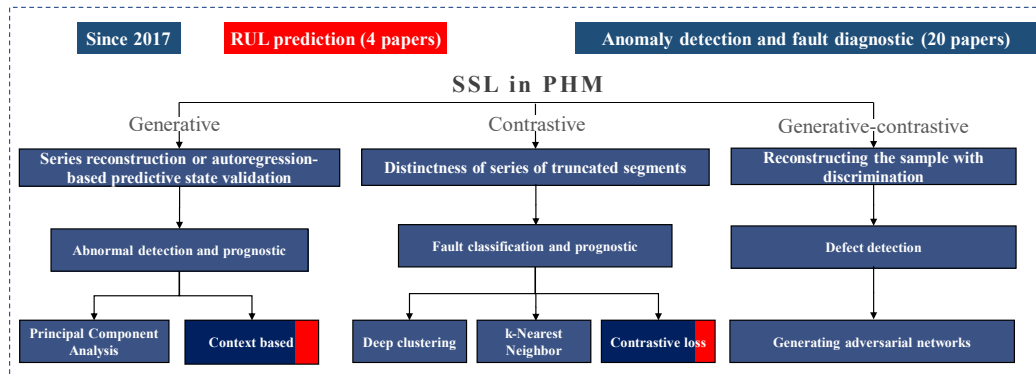


Figure 2.11: Taxonomy of self-supervised learning in PHM.

SSL was introduced to PHM around 2017, with 35 related research articles published. These can be categorized into three main SSL paradigms: “Generative” and “Contrastive” approaches ranging from fault detection and diagnostics to failure prognostics. The “Generative-Contrastive” category, on the other hand, is commonly used in RUL prediction and fault diagnostics tasks. These can be represented by Fig. 2.11.

A comprehensive overview to examine how these approaches are practically implemented in the context of PHM tasks is summarized in Table 2.4. In this table, “D-i” denotes incorporating downstream information into the pre-training stage, and “P-f” represents frozen pre-training knowledge into the downstream finetuning stage.

**Table 2.4:** Summary of SSL in PHM.

<b>Insights</b>	<b>Pretext tasks</b>	<b>D-i</b>	<b>P-f</b>
Siamese CNN for power line <b>abnormal detection</b> [107].	<b>Generative group:</b> Some input image parts are masked, and the tower-conductor region is reconstructed using a two-branch structure.	✓	×
Multi-mode non-Gaussian variational autoencoder for <b>anomaly detection</b> in complex electromechanical equipment [108].	<b>Generative group:</b> Capture the input data’s underlying distribution for effective reconstruction and anomaly scoring.	×	✓
Kernel PCA for Metal etching process <b>fault detection</b> [109].	<b>Generative group:</b> The Kernel PCA model is trained on normal samples, and faulty samples are detected using a learned reconstruction error threshold.	×	✓
Encoder only transformer for <b>abnormal detection</b> in Tennessee Eastman Process dataset [110].	<b>Generative group:</b> Randomly masking inputs and training the model to reconstruct the masked portions.	×	✓
Sparse autoencoders for motor <b>fault diagnostics</b> [111].	<b>Generative group:</b> Reconstruct the 2D time-frequency spectrogram through encoding-decoding, using the final encoding layer’s output for fault diagnostics.	×	✓
Aluminum alloy structures fatigue damage <b>prognostics</b> [112]	<b>Generative group:</b> Autoencoders (AE) reconstruct input, Autoregressive (AR) models predict the next timestep, both capture sequential patterns and dependencies.	✓	✓
1D ResNet-18 for Bearings <b>RUL prediction</b> [113].	<b>Generative group:</b> Generates pseudo-labels for data transformation using semantic and manifold regularization to obtain unsupervised consistency.	×	✓

Continued on next page

Table 2.4 – continued from previous page

Insights	Pretext tasks	D-i	P-f
Variational automatic encoder for C-MAPSS <b>RUL prediction</b> [114].	<b>Generative group:</b> The model is trained to reconstruct the input unlabeled data.	×	✓
Masked autoencoders for machine tool <b>RUL prediction</b> [115].	<b>Generative group:</b> Enhanced RUL prediction model initialization through masked patch reconstruction.	×	×
Gated recurrent unit for <b>RUL prediction</b> in Prognostia bearing dataset [116].	<b>Generative group:</b> Predicts future vibration data patterns by assessing correlations between measurements at different time steps.	×	✓
Variational-autoencoder for Tool <b>wear prediction</b> [117].	<b>Generative group:</b> Utilizing part or all input data as labels in input reconstruction process.	×	✓
Two 3-layer 1-D ResNets as backbones for different time series classification encoders in <b>fault detection</b> [118].	<b>Contrastive group:</b> Contrastive pre-training clusters similar time and frequency representations in time-frequency space.	×	×
CNN for wind turbine blade <b>damage detection</b> [119].	<b>Contrastive group:</b> Shared CNN branches distinguish diverse samples via output differences.	×	✓
GRU for <b>fault detection</b> in bearings and gears [120]. CNN for wind turbines <b>fault diagnostics</b> and <b>abnormal detection</b> [121]	<b>Contrastive group:</b> It treats augmented samples as positives and randomly selected samples as negatives to learn compact representations.	×	✓
Deep convolutional neural network for bearing incipient <b>fault detection</b> [122].	<b>Contrastive group:</b> In dual branches structure, pair augmented samples closely (positive pairs). Separate distinct samples widely (negative pairs). Utilize InfoNCE loss for similarity and dissimilarity. Update one branch's parameters, while the other update with momentum.	×	✓

Continued on next page

Table 2.4 – continued from previous page

Insights	Pretext tasks	D-i	P-f
Domain adversarial neural networks for rolling bearing <b>fault diagnostics</b> [123].	<b>Contrastive group:</b> Iteratively label target samples with a source domain classifier and optimize the network by minimizing classifier and domain discriminator losses.	×	✓
Signal momentum contrast network for Aero-Engine bearing <b>fault diagnostics</b> [124].	<b>Contrastive group:</b> The query network extracts input data features, while the key network, updated through momentum and InfoNCE loss, supplies reference features for comparison.	×	✓
Down-sampling and interaction network in Paderborn University (PU) dataset on <b>fault diagnostics</b> [125].	<b>Contrastive group:</b> Employs classifier to capture long-term temporal relations between past and future segments	×	✓
Siamese autoencoder for bearing <b>fault diagnostics</b> in Politecnico di Torino dataset [126].	<b>Contrastive group:</b> Enhance feature similarity between raw data and their geometric transformations in the encoder branches.	×	✓
1D ResNet encoder for cutting tool and bearing <b>fault diagnostics</b> [127].	<b>Contrastive group:</b> Maximizing agreement between differently augmented views of a signal sample.	×	✓
ResNet-18 for UoC parallel gearbox <b>fault diagnostics</b> [128].	<b>Contrastive group:</b> Similar samples are kept close while different samples are pushed further apart.	×	✓
Variante generative adversarial network for <b>fault diagnostics</b> in rotating machine [129].	<b>Contrastive group:</b> Maps similar domains using classifier features, quantifies differences with MMD, and promotes latent representation interpolation.	×	✓
One-stage momentum encoder for cross-domain bearing <b>fault diagnostics</b> [130].	<b>Contrastive group:</b> Lower loss for high positive similarity, low negative similarity.	×	×

Continued on next page

Table 2.4 – continued from previous page

Insights	Pretext tasks	D-i	P-f
ResNet for bearing fault diagnostics [zhang2021self].	Contrastive: Augment wavelet features, then employ SimCLR to maximize consistency across data versions.	×	×
Siamese DNNs for C-MAPSS RUL prediction [131].	<b>Contrastive group:</b> Learn a latent space that clusters similar RUL values and distinguishes dissimilar ones.	×	✓
<b>RUL prediction</b> on CMAPPS engine dataset, NASA, and CALCE battery dataset [132].	<b>Contrastive group:</b> Adding a cycle loss term to InfoNCE function amplifies weights for larger cycle differences and diminishes them for smaller ones.	×	✓
k-nearest neighbor for <b>fatigue life prediction</b> in composite materials [133].	<b>Contrastive group:</b> Maximize neighbor entropy, then predict using a convex combination.	×	✓
Few-shot <b>fault diagnostics</b> of hoisting systems. [134].	<b>Generative-Contrastive group:</b> Train the GAN to recognize and generate these waveforms, and identify the variations created by the translation of the original data along the X-axis.	✓	×
Tensor domain-adversarial network with deep auto-encoder for multi-bearing dataset <b>RUL prediction</b> [135, 136].	<b>Generative-contrastive group:</b> Pretrain the network to fit pseudo-failure thresholds for reconstructing increasing vibration sequences monotonically.	✓	✓

According to the Table 2.4, we found that the “Generative” SSL approach focuses on “series reconstruction or auto-regression based predictive state validation.” This method teaches models to recreate data patterns or features, enabling them to rebuild full sequences from partial data by leveraging natural patterns. Models also learn to predict future sequence events based on earlier data [112]. The approach employs context-based models such as autoencoders, variational autoencoders, recurrent neural networks, and transformers. It is predominantly used for fault detection and failure prediction tasks, with limited application in fault diagnostics. However, it faces challenges when dealing with real-world

data that significantly deviate from the training dataset, especially in the presence of anomalies [137].

“Contrastive” learning-based approach aims to discern the “uniqueness of truncated segment series.” This method trains models to identify data variances, helping acquire meaningful feature representations. It encourages the model to recognize similarities among related samples while differentiating unrelated ones. The approach primarily uses contrast loss functions (e.g., infoNCE [138]) and specific processing structures (e.g., Siamese structures [139]). In RUL (Remaining Useful Life) prediction, it assumes a direct correlation between the temporal difference of samples and the variance in degraded features. This approach is more common in fault diagnostics but less widespread in failure prediction tasks. However, it faces challenges in constructing suitable negative samples and aligning with the ultimate goal of predicting degradation sequence trends [140].

“Generative-contrastive” approach combines elements from the first two groups. This method ensures the preservation and accurate reconstruction of discriminative features, maintaining essential distinguishing characteristics. It’s exemplified by techniques like GANs [134]. This approach utilizes a contrastive objective function to overcome limitations in the generative group. Compared to the contrastive group, it has an additional decoder that accurately depicts primary data traits, such as waveform reconstruction in monitoring data [134]. The reconstructed samples closely correspond to the degradation features [136], allowing for more insightful analysis. However, current designs of these architectures do not adequately address the representation of the degradation, and there is a mismatch between the upstream-generated contrast task and the downstream RUL prediction task.

### 2.3.3 Focus of existing doctoral theses

This section reviews the theoretical research presented in existing PhD dissertations on PIML and SSL, to identify potential methodological research gaps. It is important to note that these methodologies were originally developed outside the PHM domain. By examining the challenges encountered across various fields, this review seeks to contribute novel insights for the application of PIML and SSL in PHM, thereby avoiding the redundancy of reinventing existing solutions. The key works and contributions of these dissertations are detailed in Appendix B.3. Additionally, we analyze the major unresolved challenges in PIML and SSL, which are summarized in Table 2.5 and 2.6.

**Table 2.5:** Unresolved challenges in PIML.

Unresolved challenges	Citation	Remarks
Scalability, Computational Complexity, and Training Stability	[141, 142, 143, 144, 145, 146]	PIML methods, especially PINNs, are computationally intensive, difficult to scale, and prone to training instability and convergence issues, particularly in high-dimensional problems or large datasets.
Sensitivity to Hyperparameters, Network Architecture, and Training Dynamics	[147, 148, 149, 145, 150, 147]	PIML models are sensitive to hyperparameter selection, network design, and can struggle with handling discontinuities and sharp gradients, necessitating careful tuning and design.
Incorporation of complex physics, boundary Conditions, and multi-physics phenomena	[146, 151, 142, 152, 153, 144, 154, 155, 156, 157, 158, 159]	Embedding complex physical laws, boundary conditions, and multi-physics phenomena into PIML models is challenging, often requiring problem-specific adaptations and specialized architectures, otherwise will influence the generalization, and extrapolation.

**Table 2.6:** Unresolved challenges in SSL.

Unresolved challenges	Citation	Remarks
Lack of supervisory signals	[160, 161, 162]	SSL relies on automatically generated pseudo-labels or proxy tasks, which may not always align well with the downstream task of interest. This can limit the quality of the learned representations.

Continued on next page

Table 2.6 – continued from previous page

Unresolved challenges	Citation	Remarks
Sensitivity to data augmentations	[163, 164]	The performance of SSL methods often heavily depends on the choice of data augmentations used to generate different “views” of the input data. Designing effective augmentations requires domain knowledge and can be challenging for some modalities.
Scalability to diverse domains	[165, 166, 167]	Many SSL techniques are developed and evaluated primarily on image datasets. Extending these methods to other domains like text, audio, or sensor data may require non-trivial adaptations.
Limited theoretical understanding	[168, 169]	While empirical results have shown the effectiveness of SSL, the theoretical underpinnings of why and when these methods work well are still not fully understood.

## 2.4 Summary

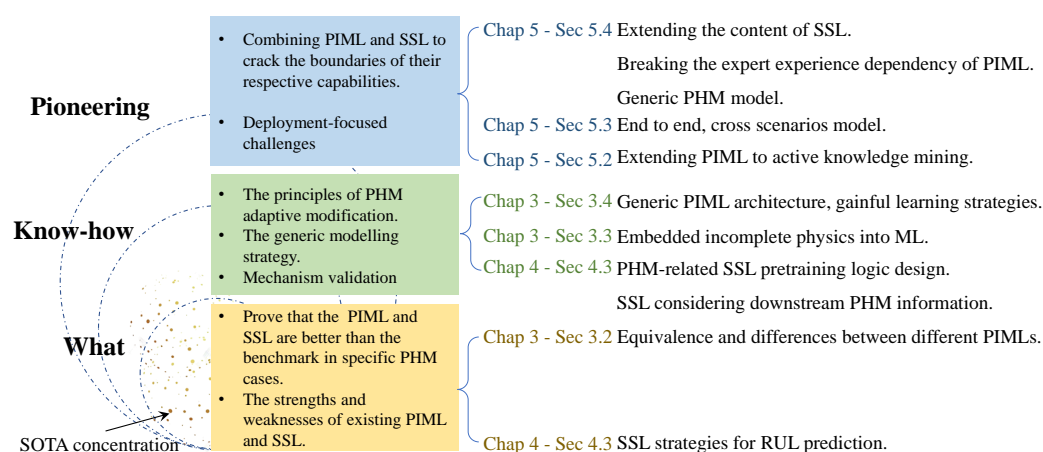
**To the best of our knowledge, the analyses and reviews in this chapter represented the first comprehensive reports of PIML and SSL in PHM.** Because from both qualitative and quantitative perspectives, it conducted a thorough bibliometric analysis of 162 research papers and 42 dissertations to identify research trends, summarize key application cases, and highlight methodological advances in PIML and SSL for various PHM tasks.

In summary, We categorized existing PIML approaches based on how physical knowledge was incorporated into the machine learning pipeline, identifying three major paradigms and their realization processes: “physics-informed input spaces,” “physics-embedded algorithm structures,” and “physics-constrained learning.” We also discussed the different forms of physical knowledge (explicit, embedded, and tacit) and their respective roles in PIML frameworks, proposing choices of knowledge and PIML paradigms. Then we presented a taxonomy of SSL approaches in PHM, focusing on “generative,” “contrastive,” and “generative-contrastive” hybrid methods. The strengths and limitations of each approach for



various PHM tasks were analyzed.

From the literature review, we found that although PIML and SSL represent the forefront of hybrid modeling and learning from unlabeled data, their application within PHM remains relatively nascent. Consequently, less attention has been given to prognostics, and cross-scenario applications are still more of a vision than a reality. In terms of underlying theories, we observed that these methods remain highly task-specific, though we identified common phenomena such as the reuse of similar knowledge representations and pre-training strategies across different cases. Current models and frameworks still lack comprehensive summarization and theoretical analysis, leading to insufficient theoretical support. It is also noteworthy that PIML and SSL were not originally developed for PHM, resulting in challenges when applying them locally in this domain. Based on these findings, we articulated the scientific positioning of the rest parts of our work in Fig. 2.12.



**Figure 2.12: Scientific positioning of our thesis.**

The scientific positioning of our thesis is categorized into three distinct tiers: “What,” “Know-How,” and “Pioneering.” Each tier represents a different level of contribution and disciplinary depth, highlighting the novelty, originality, and pioneering aspects of our research.

- **“What” tier: Establishing the foundation.** The first tier, “What,” encompassed the foundational studies that established the rationale for employing PIML and SSL in PHM. Many studies within this tier demonstrated the superiority of these techniques in specific PHM scenarios, providing a solid basis for understanding their strengths and weaknesses. Our research built on this foundation by filling existing application gaps and conducting a detailed analysis of the equivalences and differences between various

PIML approaches, as discussed in Chapter 3-Section 3.2.1. Additionally, we explored SSL strategies specifically for bearings' RUL prediction in Chapter 4-Section 4.2, refining the application of SSL in PHM contexts.

- **“Know-How” tier: Advancing methodological expertise.** The second tier, “Know-How,” delved into the localization and adaptation of PIML and SSL techniques within the specific context of PHM. This tier represented the core methodological advancements made in Chapters 3 and 4 of the thesis. We focused on developing a generic PIML architecture and gainful learning strategies, as detailed in Chapter 3-Section 3.4, while addressing the challenges of incorporating incomplete physical knowledge into the models, discussed in Chapter 3-Section 3.2. Furthermore, Chapter 4-Section 4.3 introduced the theoretical logic behind PHM-related pre-training designs in SSL, emphasizing the importance of considering downstream PHM information. These contributions were crucial for advancing the practical application of PIML and SSL in PHM, providing the necessary “know-how” for effective implementation in real-world scenarios.
- **“Pioneering” tier: Expanding frontiers and shaping the future.** The third tier, “Pioneering,” represented the cutting-edge contributions of our thesis, where theory and practice converged to push the boundaries of PIML and SSL. In Chapter 5, we extend the concepts of both PIML and SSL, focusing on breaking the expert experience dependency of PIML and exploring deployment-focused challenges. Chapter 5-Sections 5.2 and 5.3 highlight our efforts to extend PIML to active knowledge mining and develop an end-to-end, cross-scenario PHM model. These advancements are particularly groundbreaking, as they propose a comprehensive, lightweight PHM framework capable of addressing complex “4Cs” scenarios with high adaptability and efficiency. Finally, in Chapter 5-Section 5.4, we present the culmination of our research: a generic PHM framework that successfully integrates the extended content of SSL and PIML. This model exhibits the ability to handle “Sparse, noisy data and scarce knowledge” across diverse PHM scenarios, offering a promising solution for future PHM systems.

# Constructing a generic PINN framework for PHM

---

## Contents

---

<b>3.1</b>	<b>Introduction</b> . . . . .	<b>50</b>
<b>3.2</b>	<b>PIML model based on mimetic theory</b> . . . . .	<b>52</b>
3.2.1	Investigating physics integration approaches: A simulated bearing degradation case study . . . . .	52
3.2.1.1	Different PIML models embedding the same physics knowledge . . . . .	53
3.2.1.2	Investigation of the PI-TCN models' performance. . . . .	55
3.2.2	New PIML model based on mimetic theory: Rotor finite element mimetic neural network . . . . .	58
3.2.2.1	Mimetic theory . . . . .	59
3.2.2.2	Rotor finite element mimetic neural network . . . . .	60
3.2.2.3	Validation of the proposed RFEMNN . . . . .	65
<b>3.3</b>	<b>Constraint projection for PIML in few-shot faults diagnostics</b> . . . . .	<b>67</b>
3.3.1	Constraint projection theory . . . . .	67
3.3.2	Enhancing RFEMNN's performance in few-shot learning with "Constraint projection" theory . . . . .	69
3.3.3	Application of constrain theory: Few-shot learning for rotor fault diagnostics . . . . .	70
3.3.3.1	Physics consistency validation results . . . . .	71
3.3.3.2	Rotor few-shot diagnostic results . . . . .	72
<b>3.4</b>	<b>A generic PIML framework</b> . . . . .	<b>73</b>
3.4.1	A two-branch parallel PIML framework . . . . .	74
3.4.2	Pretrained-physics alignment multistep training . . . . .	75
3.4.3	Case study: Battery RUL prediction using small early-stage lifecycle data . . . . .	76
3.4.3.1	Comparison results with SOTA models . . . . .	78
3.4.3.2	Comparison results of prediction performance on a new dataset without re-learning . . . . .	79
<b>3.5</b>	<b>Summary</b> . . . . .	<b>80</b>

This chapter is based on the following publications:

- J2 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2023). Rotor dynamics informed deep learning for detection, identification, and localization of shaft crack and unbalance defects. *Advanced Engineering Informatics*.
- J4 **WeiKun DENG**, Hung LE, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER, Dazhong WU (2024). Generic Physics-Informed Machine Learning Framework for Battery Remaining Useful Life Prediction Using Small Early-Stage Lifecycle Data. *Available at SSRN 4770354* (Submitted to *Applied Energy*, Finished the revision, waiting for the Editor's decision).
- C1 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2022). Physics-informed lightweight temporal convolution networks for fault prognostics associated to bearing stiffness degradation. *PHM Society European Conference*, 7(1): 118-125.
- C4 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2023). A Few-Shot Learning Framework for Rotor Unbalance and Shaft Crack Fault Diagnostic Based on Physics-Informed Neural Network. *Structural Health Monitoring 2023*.
- C5 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJACHER (2024), Hung LE, Dazhong WU. A Novel PIML Architecture with Innovative Learning Paradigm Applied in Battery Prognostics. *CODIT 2024*.

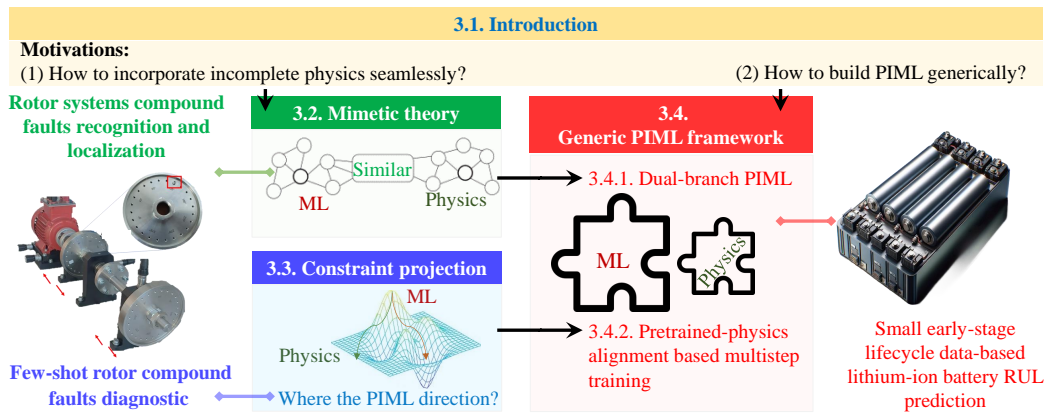
## 3.1 Introduction

The previous chapter highlighted PINNs within PIML as crucial for enhancing PHM. However, integrating physics into neural networks for PHM presents several challenges:

1. Interpreting the acting mechanism and flexibly applying informed knowledge in machine learning.
2. Using incomplete and empirical knowledge to inform ML models for complex PHM tasks, such as fault identification, localization, and diagnostics simultaneously with limited data or prognostics in small early-stage life monitoring data.

3. Ensuring the gainful benefits of the informed ML model when embedding physics, maintaining the reliability, robustness, and accuracy of physics knowledge in ML models to improve performance.
4. Reducing the expert experience reliance on PIML framework customizing design.

This chapter has 5 sections. They aim to address these aforementioned challenges through a structured approach, divided into three subsequent sections, as illustrated in Fig. 3.1.



**Figure 3.1:** Research framework for constructing a generic PIML in PHM.

First, Section 3.2 focuses on the integration of incomplete physics-informed knowledge within ML frameworks, based on the “Mimetic theory.” It first investigated the mechanisms and effectiveness of three physics integration approaches—PI input space, PI algorithm structure, and physics constraint learning—through a comprehensive comparative analysis of their equivalences and differences in a simulated bearing degradation scenario. The insights gathered from this analysis then guide us in refining and applying the most promising integration method to a more complex real-world problem—detecting and localizing combined defects in rotor systems.

Next, Section 3.3 details a specialized training methodology tailored for PIML, aimed at enhancing the performance of machine learning models after the incorporation of physical laws. This section introduces the “Constrained projection” learning strategy, which ensures that the ML model maintains physical consistency while avoiding any potential performance degradation that might arise from the embedding process. The effectiveness of this strategy is demonstrated through its application to a few-shot learning task in compound fault diagnostics.

Finally, Section 3.4 presents the development of a “**Generic PIML architecture**” from an engineering perspective. This architecture is designed to incor-

porate physics, using the “Mimetic theory” as a parallel plug-in to existing ML models. Additionally, it simplifies the “Constrained projection” approach into a sequential multi-step optimization process. The versatility and robustness of this architecture are validated through its application to the RUL prediction of fast-charging lithium-ion batteries, focusing on early-stage lifecycle data.

## 3.2 PIML model based on mimetic theory

In the previous chapter, our literature review identified three distinct methods for integrating physics knowledge into machine learning models: Physics-Informed (PI) input space, PI algorithm structure, and Physics constraint learning. Each method provides a different approach to embedding the same physical principles within the model. Subsection 3.2.1 aims to explore the working mechanisms of these three approaches using a simulated dataset of bearing degradation. Building on the insights gained from this analysis, Subsection 3.2.2 will focus on the most promising knowledge-integration approach, developing a model to tackle the more complex real-world challenge of detecting and localizing combined defects in rotor systems.

### 3.2.1 Investigating physics integration approaches: A simulated bearing degradation case study

This section seeks to address three critical questions: 1) Can the integration of physics knowledge enhance model performance? 2) If so, how does this improvement manifest? And finally, 3) which of the three methods delivers the best performance? To explore these questions, we conduct a detailed case study using simulated bearing degradation data, comparing the effectiveness of each PIML approach in incorporating physics knowledge to optimize model outcomes.

To assess the impact of physics-informed knowledge on machine learning model performance, we utilize a simulated bearing degradation scenario where the RUL of the bearing is predicted using a Temporal Convolutional Network (TCN). This study showcases the versatility of integrating physics knowledge at three different stages of the machine learning pipeline, resulting in the development of three distinct models: the PI input space model, the physics-embedded layer model, and the PI loss model. The numerical simulation of the vibration signals generated during the bearing degradation process, which forms the basis of our analysis, is comprehensively detailed in Appendix C.1.

The proposed TCN is depicted in Fig. 3.2, is introduced as a lightweight and purely data-driven benchmark model. The TCN is designed to predict the normalized RUL of bearings using time-domain statistical features, including mean,

variance, maximum, minimum-maximum, root mean square, skewness, kurtosis, peak factor, waveform factor, impulse factor, and margin values. The model comprises 3,411 parameters, and its architecture incorporates a residual block (resblock) structure that utilizes a causal, separable 1D convolution layer activated by H-swish [170].

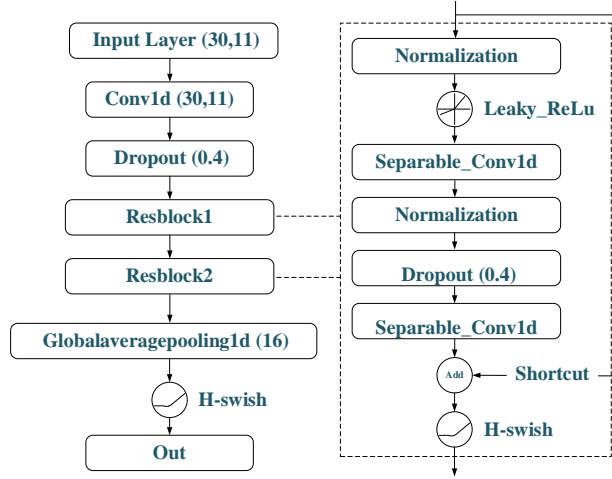


Figure 3.2: Lightweight TCN architecture diagram.

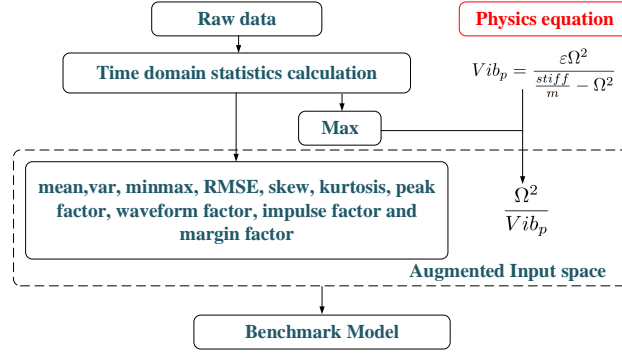
The relationship between stiffness and vibration amplitude shown in Eq.(3.1) [171] are used as the physics knowledge in building the three PIML models, where  $Vib_p$  is the peak value of the vibration signal and  $stiff$  represents the corresponding equivalent contact stiffness level.  $\varepsilon$  denotes the relevant imbalance in the system load. It is the extrinsic excitation of the bearing vibration.  $m$  represents the equivalent system mass.  $\Omega$  is the rotation speed. In real conditions, the exact values of  $\varepsilon$  and  $m$  are unknown. Only the parameters  $\Omega$  and  $Vib_p$  are available in vibration-based RUL prediction.

$$Vib_p = \frac{\varepsilon\Omega^2}{\frac{stiff}{m} - \Omega^2} \quad (3.1)$$

### 3.2.1.1 Different PIML models embedding the same physics knowledge

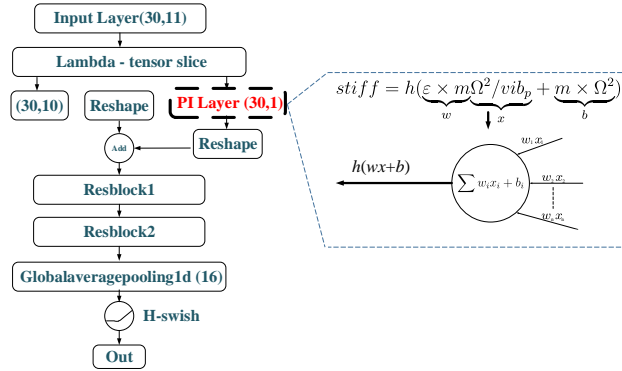
Factor  $\Omega^2/Vib_p$ , designed as a physics-informed feature to predict bearing RUL, reflects stiffness degradation trends, making it a valuable addition to the original set of 11 time-domain statistical features. This feature replaces the Max feature, maintaining the dimensionality of the dataset at  $60 \times 11$ , consistent with the benchmark model. Although  $\Omega^2/Vib_p$  does not directly measure stiffness, its inclusion introduces a physically meaningful quantity into the model, enhanc-

ing predictive capability. The updated model configuration, incorporating this physics-informed health indicator (PHI), is illustrated in Fig. 3.3.



**Figure 3.3:** Updated model configuration with physics-informed health indicator.

Eq. (3.1) can be also integrated into ML framework as an input-output module, compensating for model incompleteness, as depicted in Fig. 3.4. In this framework, the unknown function  $g(\cdot)$  is approximated by a custom neural network layer function  $h(\cdot)$ , enabling the extraction of the PHI  $\Omega^2/Vib_p$  via a transformation layer. This transformation processes the physics-based input, allowing the subsequent custom hidden layer to approximate stiffness degradation functions. The unknown parameters  $\varepsilon$ ,  $m$ , and  $U_{stiff}$  are treated as trainable variables, updated during training to optimize model predictions.

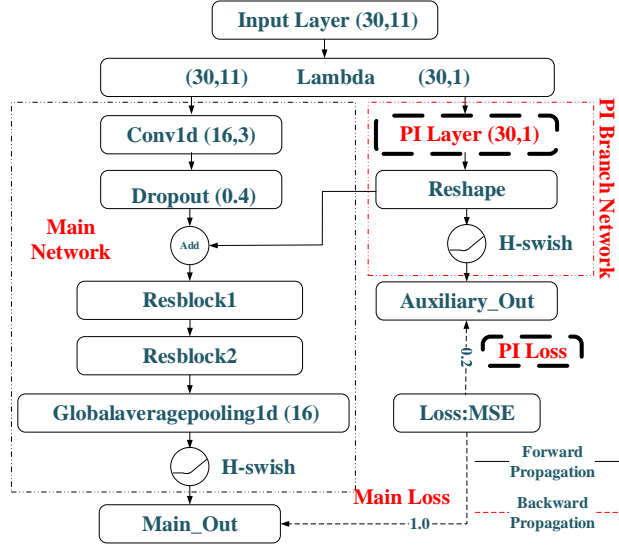


**Figure 3.4:** Embedding physics equations into a neural network layer.

In addition, Eq.(3.1) can also be used to constraint a constraint module, ensuring consistency by evaluating discrepancies between the physics-informed and original NN outputs. An output layer following the physics-informed layer generates a hidden indicator, influencing hyper-parameter optimization through



conflict loss, as shown in Fig. 3.5. This shared optimization process promotes physics consistency within the TCN.



**Figure 3.5:** Constructing Physics-Informed Loss based on model conflicts.

### 3.2.1.2 Investigation of the PI-TCN models' performance.

During training, a 1000-epoch regime with early stopping (patience of 80 epochs) is used. Parameters are uniformly initialized, with a batch size of 128 and an input shape of  $30 \times 11$ . The Adam optimizer [172] is consistently applied across all models.

Fig.3.6 presents predictions from various models on 10 randomly selected test set trajectories, while Fig. 3.7 displays box plots illustrating differences between predicted and true RUL for the entire test set. These results highlight PI-TCN models' performance compared to the benchmark data-driven model, indicating diverse possibilities for incorporating physics knowledge and potential benchmark improvements. Notably, the PI loss model demonstrates superior performance, with an error range of [17.97, 15.65], outperforming benchmark, physics-embedded layer, and PI loss model errors [-95.51, 79.83], [-26.85, 36.66], and [-33.38, 26.11] respectively.

Comparing the performance of the proposed PI-TCN models with the TCN model on the overall test sets, we find that:

1. All the PI-TCN model's predictions are more accurate with smaller prediction error limits than the Benchmark model.

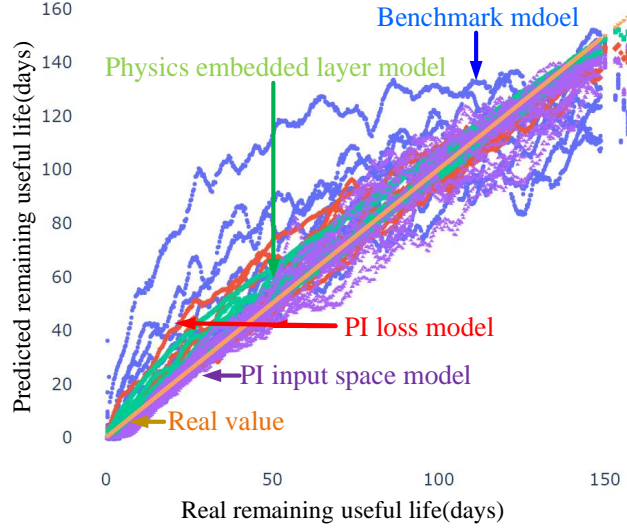


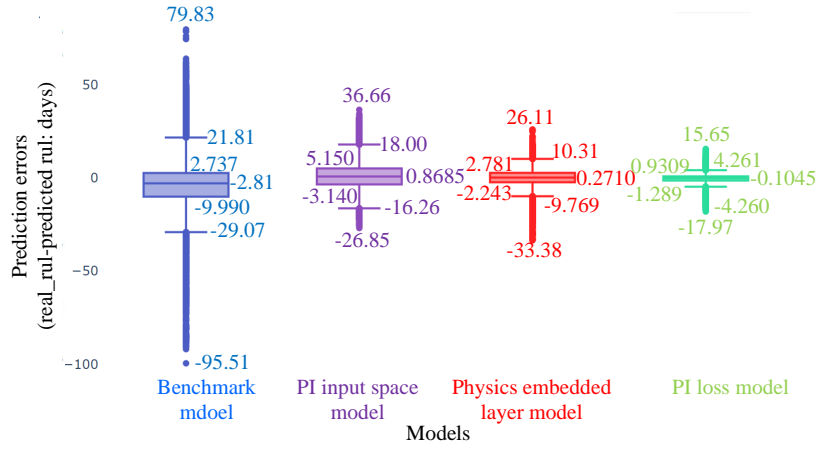
Figure 3.6: Prediction results of different models.

2. Among the three different PI-TCN models, *PILLM* (represented by the green line) has the best prediction stability with the most compact upper and lower error limits and the minimum error means, as presented in Fig.3.7, showing the effectiveness in informing physics by the custom structure design.

A deep discussion on the PIML model performance, particularly the impact of embedded physics knowledge on bearing RUL predictions is detailed in Appendix C.2. We use channel-by-channel testing on layers with embedded physical knowledge, assessing their influence on final test loss across various weight compression ratios. Additionally, we analyze the output of hidden layers corresponding to each channel, generating information output heatmaps to visualize the distribution of channel information and the degree of correlation between channels. The analysis reveals that the location and manner of embedding physical knowledge significantly affect its efficacy. Generally, embedding closer to the input layer proves more effective but risks dilution by subsequent network layers. The study identifies a trade-off between highlighting physical features and obtaining robust integrated representations. Notably, in the PIFM, channels containing  $\Omega^2/Vib_p$  show predominant focus, indicating that changes in the weighting of this information have the greatest impact on loss changes.

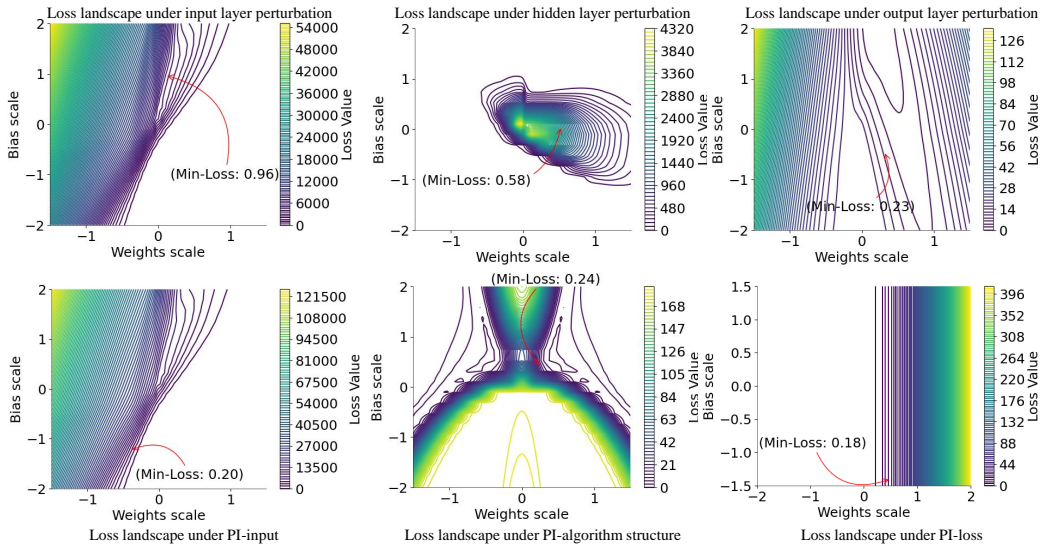
Additionally, we conduct a thorough analysis of the loss landscape to evaluate the optimization behavior of the PI models. To achieve this, we use the weights and biases from the trained models as reference points and generate a grid of equally spaced points that represent scaling factors (ranging from 0 to 1) applied to these weights and biases. The loss function is then evaluated at each point on the grid, allowing us to create a detailed map of the loss landscape, as shown in

### 3.2. PIML model based on mimetic theory



**Figure 3.7:** Model evaluation results.

Fig. 3.8. In this figure, the horizontal axis represents the scaling factor for the original weights of the layer, while the vertical axis represents the scaling factor for the original biases. The red arrow curve highlights the path of minimum loss.



**Figure 3.8:** Loss landscapes of three PIML models.

From Fig. 3.8, we can observe that:

- (1) Both standard and PI models exhibit a distinct, narrow “canyon” of minimum loss, indicating a high sensitivity to small perturbations in weights and biases.

- (2) The PI model demonstrates gentler gradients around the minimum loss path and a broader “basin” near the optimum, suggesting enhanced fault tolerance and a reduced risk of overfitting.
- (3) The loss range for the PI model (13,500 to 121,500) is 2.25 times that of the benchmark model, indicating a wider exploration space.
- (4) The minimum loss value decreases from 0.96 in the benchmark model to 0.20 in the PI model, demonstrating superior optimization.
- (5) In the middle hidden layer, the PI model presents a narrower, more pointed “canyon” running vertically, suggesting a more constrained optimal weight range.
- (6) In the output layer, the PI model exhibits a wider, parabola-like minimum loss path, indicating greater robustness to weight perturbations.

These findings underscore the substantial impact of incorporating domain-specific physical knowledge into ML architectures, leading to models with enhanced robustness, accuracy, and generalization capabilities. In summary, we proposed using “PI-algorithm structure” to gradually integrate physics. *It is realized by the segmentation of comprehensive knowledge into sub-items, which are then strategically embedded across various NN hidden layers.* This approach establishes a data flow consistent with physics principles and enables a gradual integration of physics knowledge, thereby managing its impact on the loss landscape more effectively.

#### **3.2.2 New PIML model based on mimetic theory: Rotor finite element mimetic neural network**

Building on the findings from the previous section, where the “PI-algorithm structure” was identified as a promising approach for gradually integrating physics into machine learning models, this subsection introduces a new PIML model based on mimetic theory. We begin by presenting the fundamentals of mimetic theory in Subsection 3.2.2.1 which provides a framework for embedding incomplete physical knowledge into neural networks. Following this, we apply this theory to develop the Rotor Finite Element Mimetic Neural Network (RFEMNN) model in Subsection 3.2.2.2, specifically designed to tackle the challenges of diagnosing compound faults in rotor systems, see Subsection 3.2.2.3.

3.2.2.1 Mimetic theory

Mimetic theory in the context of neural networks, see Fig. 3.9, involves the decomposition of physics formulas into the NN fundamental operators and associated learnable parameters. Each operator is mimicked by a corresponding neural network component, such as weights, biases, and tensor operators, which are designed to replicate the behavior of these physical operators. The relationships between these operators are then mimicked through the architecture of the neural network, including the design of layer connections and activation functions. This approach allows the neural network to embody the underlying physical laws, resulting in a physics-informed model that can more accurately predict outcomes by incorporating domain-specific knowledge.

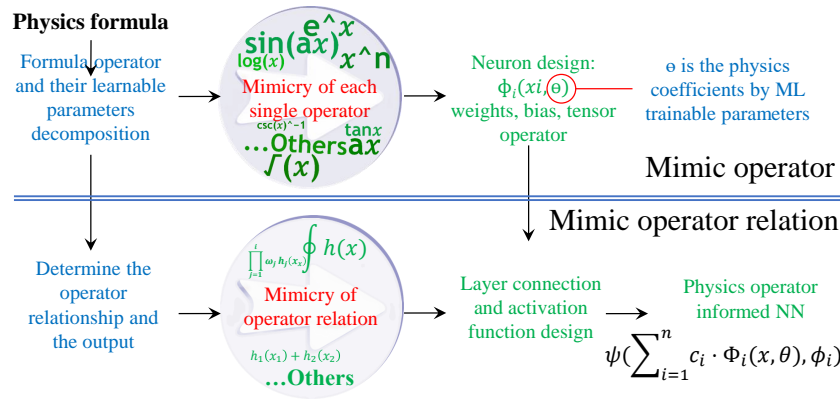
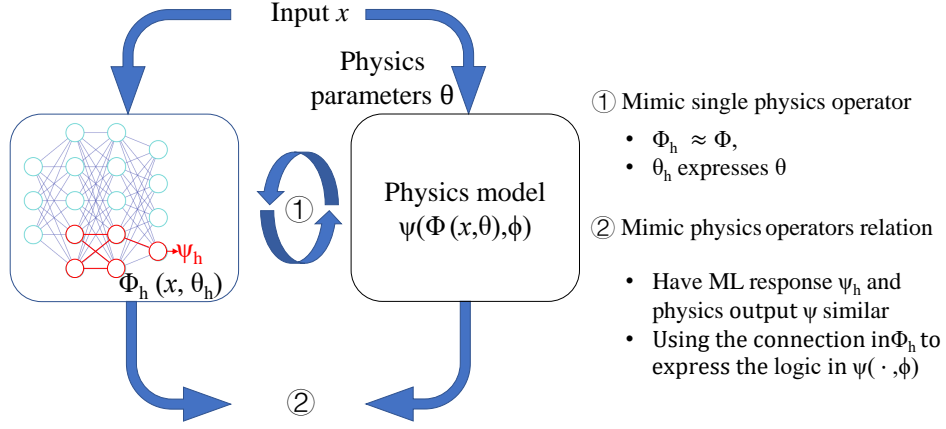


Figure 3.9: Illustration of mimetic theory in the context of neural networks.

We call the NN’s computational units operators, which define the data flow through layers such as convolutions and activation functions. The targets of the mimetic theory are shown in Fig. 3.10. At the “Mimic operator” level, the physical formula is broken down into its constituent mathematical operators, and then individual neurons or a layer of neurons are designed to replicate these operators. Each neuron can be viewed as an operator with a tensor calculator capable of executing various mathematical transformations. The corresponding variable-dependent coefficients of this physical operators physics formula are then denoted as the learnable weights  $\mathbf{W}$  in the layer, and the variable-independent coefficients to be determined are denoted as the bias  $\mathbf{b}$  in the layer.

The “Mimic operator relation” integrates the results from the “Mimic operator”, aiming to simulate how these operators interact with each other in the context of the overall physical formula. This involves designing the connections between layers (interlayer relationships) to reflect the composite structure of the physical formula  $f$ . For instance, if the formula involves the product of two



**Figure 3.10:** Mimetic theory for deriving logical and operator content alignment.

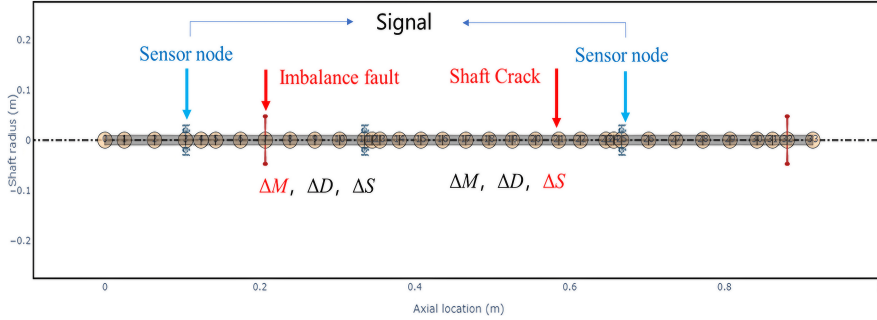
operations, the network would have a structure where the outputs of neurons representing these operations are multiplied.

### 3.2.2.2 Rotor finite element mimetic neural network

Building on the foundation of mimetic theory introduced in the previous subsection, this section presents the RFEMNN, a model specifically designed to mimic rotor dynamics for diagnosing compound faults in rotor systems. The RFEMNN model is structured to replicate the diagnostic logic inherent in rotor dynamics through its data flow, allowing it to effectively capture the complex interactions and behaviors governed by physical equations. In this context, RFEMNN serves as an empirical validation of mimetic theory, enabling us to explore its practical application and identify any challenges that arise in modeling real-world rotor faults. The rotor dynamics, central to this model, are governed by the fundamental equations outlined in Eq. (3.2):

$$M \underset{\text{known}}{\ddot{q}} + Sq + D\dot{q} = F(q, t) \quad (3.2)$$

Where  $q$  is the vibration displacement,  $\dot{q}$  and  $\ddot{q}$  are the velocity and the acceleration of the rotor. Among those parameters, only the  $\ddot{q}$  is monitored in this work by vibration signals collected using accelerometers.  $M$ ,  $S$ ,  $D$ , and  $F(\cdot)$  are the mass, stiffness, damping, and excitation force of the rotor. *It is important to note that the variables  $M$ ,  $S$ , and  $D$  are matrices whose elements are contingent upon environmental and degradation factors, with only partially known mathematical expressions. In practical applications, the precise expressions and values of  $F(x, t)$  are difficult to ascertain due to incomplete knowledge of the excitation.*



**Figure 3.11: FEM-based rotor fault diagnostic.**

Fault responses and locations are captured by the variation of  $M$ ,  $S$ ,  $D$  in the finite element matrix as shown in Fig. 3.11. In the rotor dynamicis-based diagnostic process, fault identification is done based on the matrix type of abnormal changes in the matrix elements and the position of these elements in the matrix, combined with response analysis [173]. We simulate this process through neural networks.

**Overview of the proposed RFEMNN model.** We developed the RFEMNN using the CNN-LSTM hybrid architecture, which, to the best of our knowledge, is the first diagnostic model capable of performing both classification and regression tasks simultaneously.

RFEMNN model synthesizes known physical relationships among the variables  $M$ ,  $S$ ,  $D$ ,  $F$ , and  $q$  by designing specific NN layers and connections as represented in Fig. 3.12, it comprises two main modules: the first simulates the structure of rotor finite element method (FEM) enclosed by a green dashed line, while the second emulates a rotor FEM solver for fault diagnostics, indicated by a red dashed line.

RFEMNN generates three outputs: fault positions, fault types, and vibration features. The finite element division ratio of the shaft elements is determined using “CNN layers 1” based on the input raw signals. The “Out 3” output, indicative of rotor vibration features, is characterized by a codec-like structure, comparing predicted features with actual features extracted from raw vibration signals to constrain the model that has the physics consistency with the real dynamic system behaviour.

Fault positions (“Out 2”) are predicted through regression analysis of variations in structural parameters across different shaft nodes, a process encapsulated in the “Structure changes” layer. This embedded physics knowledge within the hidden layers facilitates accurate localization of faults.

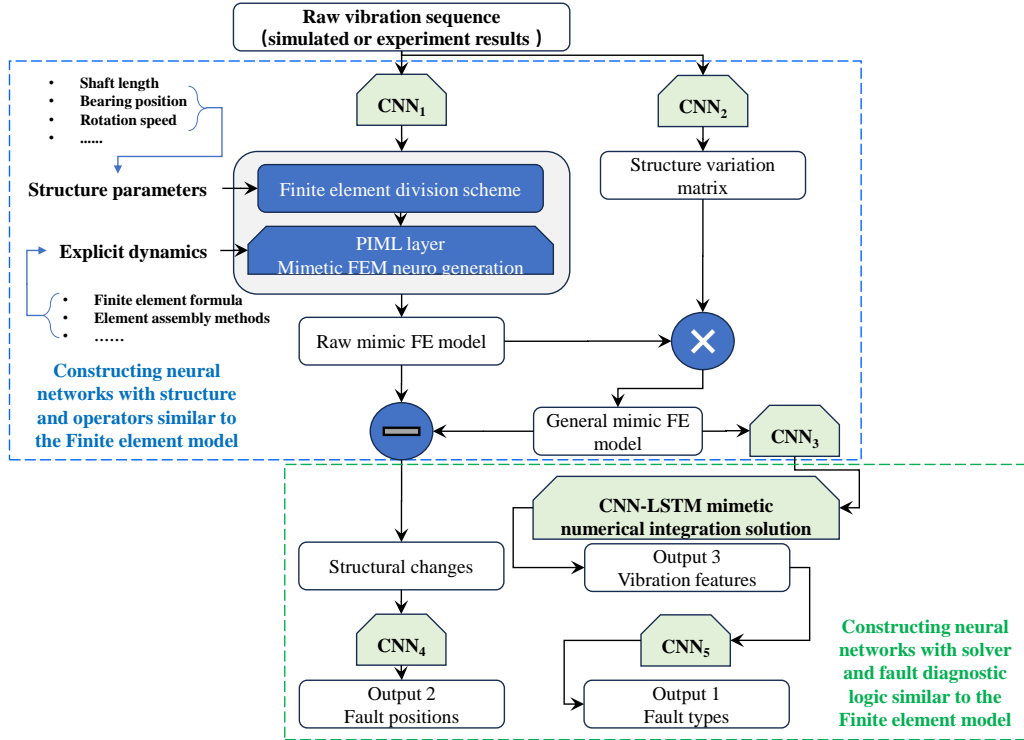


Figure 3.12: Schematic diagram of the proposed RFEMNN model.

Finally, fault types (“Out 1”) are output from “CNN layers 4,” which utilizes both the “Structure changes” and the temporal vibration features from “Out 3.” The “Structure changes” incorporate alterations in the matrices  $M$ ,  $K$ , and  $D$ , which simulate the diagnostics based on the physics properties changes.

**Customizing layers to mimic the Finite Element Model.** The RFEMNN framework inputs raw vibration signals and employs a specialized NN architecture to simulate the FEM generation and assembly processes. At its core, a CNN layer processes these signals to simulate the estimation of the shaft elements’ proportional lengths. The total shaft length determines the specific lengths of elements, denoted as neuron output values  $Len_i$  for  $i = 1, 2, 3, \dots, n$ , which are subsequently utilized to compute the equivalent elementary mass  $m_i$ , stiffness  $s_i$ , and damping  $d_i$  in Eq. (3.2.2.2).

The properties of each finite element, such as node position, cross-sectional area, moment of inertia  $I$ , Young’s modulus  $E$ , and mass, are encapsulated in a vector and serve as inputs to the “PIML layer.” This layer adheres to the finite element assembly rules, ensuring that the displacement at one shaft element’s right endpoint aligns with that of the next element’s left, as shown in Fig. 3.14.



### 3.2. PIML model based on mimetic theory

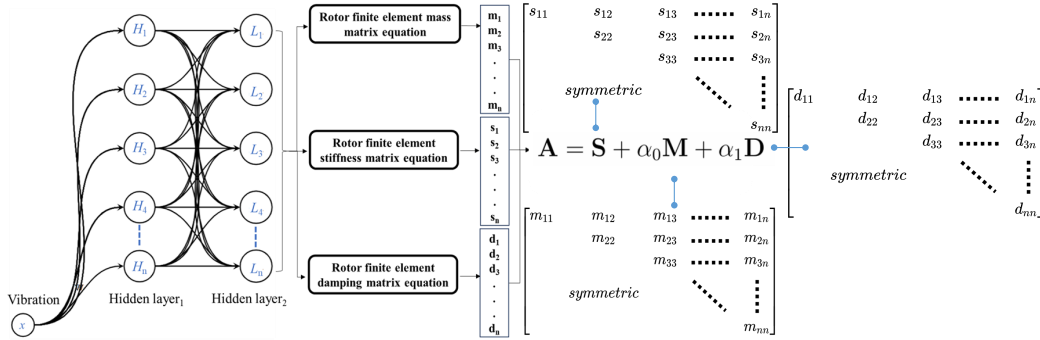


Figure 3.13: Structure of a FEM elementary neuron in NN layer.

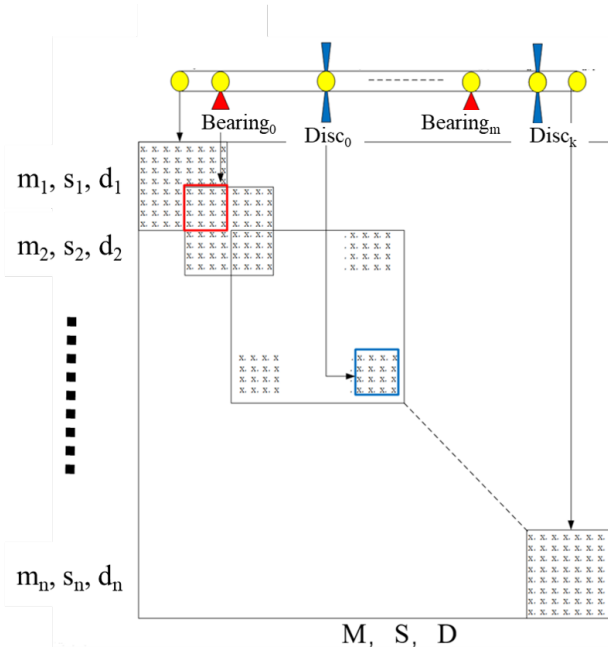
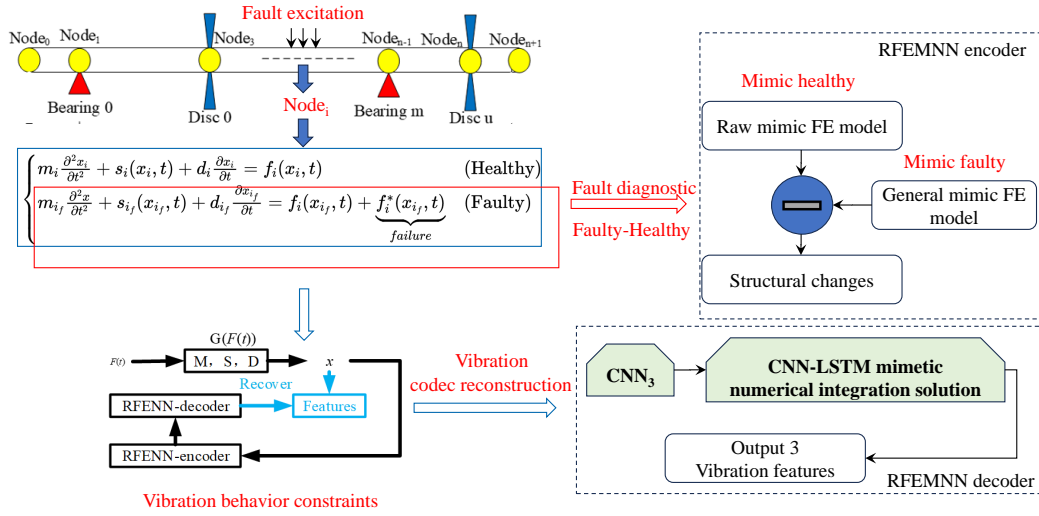


Figure 3.14: Building customized layers according to FEM assembly rules.

RFEMNN further incorporates disc mass, bearing stiffness, and damping into its neural network layer. Distributing these attributes based on the shaft geometry and bearing node positions though estimating these parameters remains challenging, as suggested by Table C.3 in Appendix C.4. The assembly of the stiffness, mass, and damping layers results in a banded asymmetric matrix spanning dimensions.

$$\begin{aligned}
 \mathbf{m}_i &= \frac{\rho A \sum_{n=1}^{i-1} Len_n}{420} \begin{pmatrix} 156 & 22Len_i & 54 & -13Len_i \\ 22Len_i & 4Len_i^2 & 13Len_i & -3Len_i^2 \\ 54 & 13Len_i & 156 & -22Len_i \\ -13Len_i & -3Len_i^2 & -22Len_i & 4Len_i^2 \end{pmatrix} \\
 \mathbf{s}_i &= \frac{EI}{(\sum_{n=1}^{i-1} Len_n)^3} \begin{pmatrix} 12 & 6Len_i & -12 & 6Len_i \\ 6Len_i & 4Len_i^2 & -6Len_i & 2Len_i^2 \\ -12 & -6Len_i & 12 & -6Len_i \\ 6Len_i & 2Len_i^2 & -6Len_i & 4Len_i^2 \end{pmatrix} \\
 \mathbf{d}_i &= \alpha m_i + \beta s_i
 \end{aligned} \tag{3.3}$$

**Mimic FEM-based diagnostics by customizing layer connections.** RFEMNN simulates both the healthy and faulty states of the rotor, facilitating advanced diagnostics by comparing the information difference in the “General mimic FE model” against the “Raw mimic FE model.”



**Figure 3.15:** Vibration behaviour-based physics consistency supervises the mimetic process.

As depicted in Fig. 3.16, a physics function is inserted into the input data between the stacked residual block layer and the LSTM layer. A two-layer LSTM and Dense layers then process the equivalent displacement, and the output is considered a time-domain feature of the acceleration sequence. The model is optimized through supervised training using the labels “out3.”

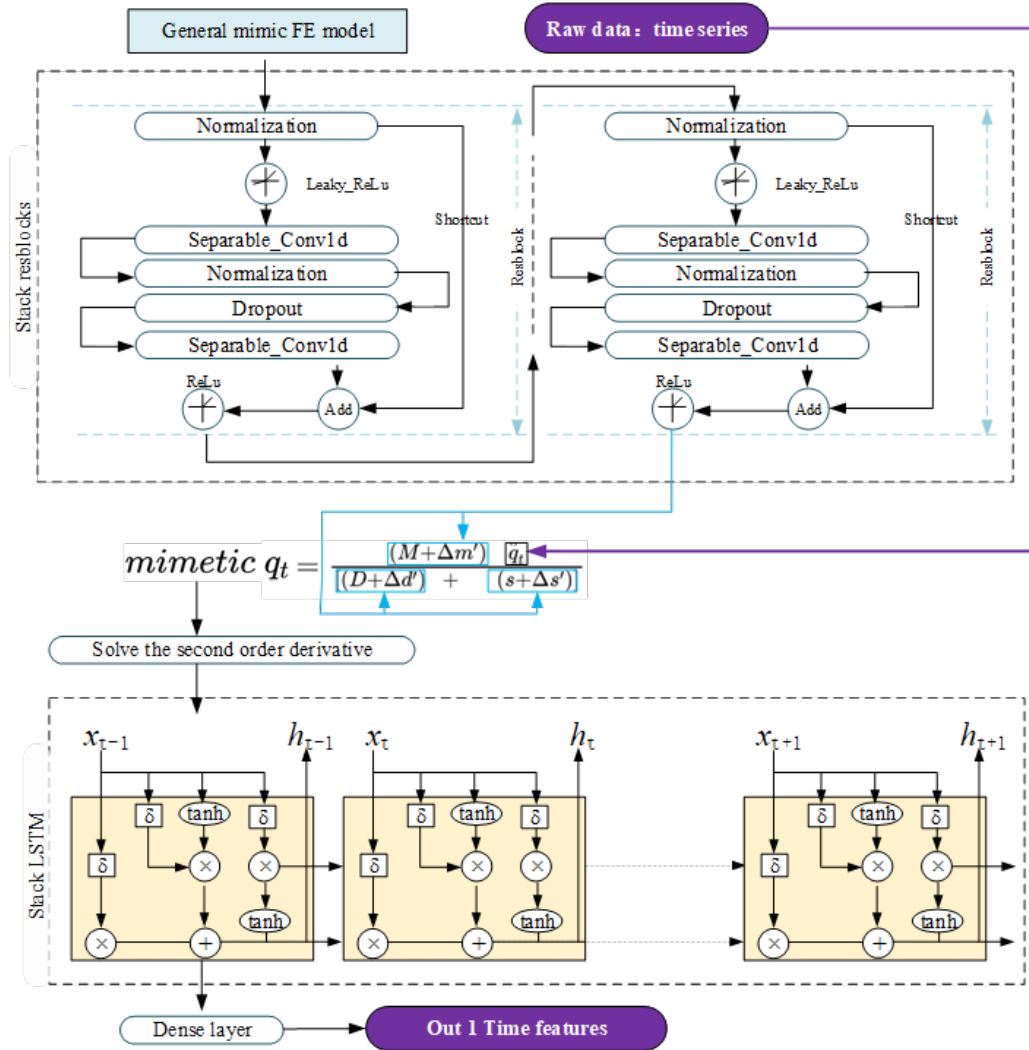


Figure 3.16: Structure of CNN-LSTM layer to mimic the FEM solution process.

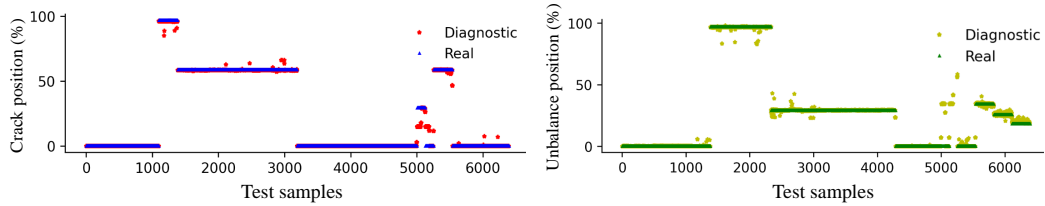
The modules in the blue rectangular portion of Fig. 3.12 are information encoders capturing anomalous mimicry of structural degradation information, which is supervised by comparing the decoded information in the green rectangular portion in Fig. 3.12 with the two types of rotor behaviours in Fig. 3.15.

### 3.2.2.3 Validation of the proposed RFEMNN

To assess the performance of the proposed RFEMNN model, we conduct the unbalanced and shaft crack experiments on platform PT 500, forming a mixed data set with different operating conditions and rotor structures. The experimental details are shown in Appendix C.3. Data conditions were established in which

*health-failure state* data were unevenly distributed, exhibiting a bias towards the *failure category*.

We evaluated the RFEMNN model and different benchmark models' effectiveness by hierarchical 10-fold cross-validation. The metrics used in evaluating its performance are defined in Appendix C.6. The results of fault type identification are presented in Table 3.1 while the ones of the fault localization are shown in Fig. 3.17. A detailed comparative analysis of the RFEMNN's performance against benchmark models, focusing on fault identification and location accuracy, is presented in Appendix C.7.



**Figure 3.17:** Average results of fault localization with 10 fold-cross validation on diagnostics fault location.

**Table 3.1:** Comparison of the proposed RFEMNN model with the SOTA models.

$Len_U$ : Average positioning accuracy for unbalanced defects,  $Len_C$ : Average positioning accuracy for shaft cracks,  $\bar{p}$ : Average positioning accuracy for all faults,  $A$ : Accuracy,  $Fa$ : False alarm,  $M$ : Missing rate,  $NO\_RFEM\_NN$ : Replacing the physics informed layers by normal CNN layer to keep the control variables on the embedded physics.

Model	Input	$A$	$Fa$	$M$	$Len_U$	$Len_C$	$\bar{p}$	Total A
CNN	Wavelet	75.21%	60.44%	0.73%	-	-	-	-
CNN	Raw	40.59%	0.6%	18.81%	-	-	-	-
DRSN	Wavelet	91.48%	2.09%	1.11%	-	-	-	-
DRSN	Raw	83.07%	35.81%	2.86%	-	-	-	-
Semi-DCNN	Raw	50.80%	57.56%	9.24%	-	-	-	-
STFNN	Wavelet	59.35%	58.62%	2.88%	-	-	-	-
ANN	Wavelet	-	-	-	24.25%	25.90%	75.63%	-
LSTM	Raw	-	-	-	19.84%	23.66%	78.05%	-
ELM	Raw	-	-	-	23.87%	24.85%	75.64%	-
$NO\_RFEM\_NN$	Raw	46.45%	3.74%	22.90	9.89%	17.05%	86.53%	40.19%
<b>RFEMNN</b>	Raw	<b>97.79%</b>	<b>9.20%</b>	<b>1.56%</b>	<b>1.95%</b>	<b>4.37%</b>	<b>96.84%</b>	<b>94.70%</b>

In Figure 3.17, the horizontal axis represents the number of test samples, each

randomly selected from varied experimental fault data. The vertical axis indicates the fault’s relative position along the rotor’s axis. The qualitative results shown in the figure indicate that the RFEMNN can effectively capture the on-axis location where the fault occurs in most of the test samples with a small bias.

According to Table 3.1, the RFEMNN model outperforms other diagnostic models with an accuracy of 97.79%, despite higher false ( $Fa$ ) and missing alarm ( $Mis$ ) rates compared to the deep residual shrinkage networks (DRSNs) model using wavelet spectrum inputs. Crucially, RFEMNN does not require additional feature engineering processes such as wavelet transformation, which simplifies its implementation. Furthermore, RFEMNN yields superior results over DRSNs when both use raw data inputs. However, models like NO\_RFEM\_NN and CNN, despite having low false alarm rates, suffer from poor accuracy and high missing alarm rates, as demonstrated by the confusion matrices in Appendix C.6, where these models predominantly misclassify unbalanced faults as non-faulty. DRSN models exhibit better performance under identical input conditions, underscoring the impact of network architecture on diagnostic efficacy. In defect localization, RFEMNN outshines all other models, achieving a top diagnostics performance metric ( $T$ ) of 94.7%. The removal of physics-embedded layers in NO\_RFEM\_NN results in a drastic reduction in diagnostics performance to only 42.43% of its original value, highlighting the critical roles of embedded physics. Further results and an in-depth discussion are provided in Appendix C.7.

## 3.3 Constraint projection for PIML in few-shot faults diagnostics

Building on the successes of the RFEMNN model in enhancing fault diagnostics and localization, the previous section demonstrated its robust performance despite the incomplete implementation of rotor dynamics physics. To further evaluate the generalizability of RFEMNN, especially in scenarios with limited data, we extend our analysis to a “Few-shot” learning framework. This section focuses on optimizing PIML using a constraint projection theory, presented in Subsection 3.3.1, specifically designed to improve the RFEMNN model’s performance under sparse data conditions, see Subsection 3.3.2. Finally, we assess the few-shot adaptability and effectiveness of RFEMNN in Subsection 3.3.3.

### 3.3.1 Constraint projection theory

The general formulation of the convex optimization problem for the PIML

loss function is presented in Eq. (3.4).

$$\min \quad \mathcal{L}(\Phi_\theta(\mathcal{X}), \mathcal{Y}) + \mathcal{L}(\Phi_\theta(\mathcal{X}), \Phi_{phy}) + \mathcal{L}(\|B\Phi_\theta(\mathcal{X}) - b\|, 0) \quad (3.4)$$

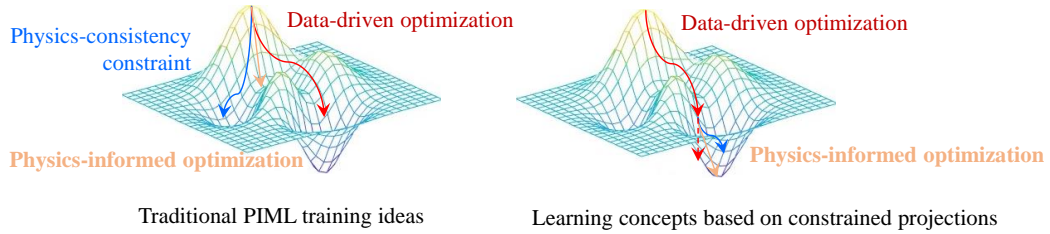
where  $\mathcal{L}(\cdot)$  is the non-convex learning objective function,  $B$  is the constraint matrices with the constraint vectors  $b$ .  $\Phi_{phy}$  is the PIML model which has the physic consistent characteristics such as positive definiteness or sparsity of a certain parameter distribution shape. Pursuing multiple goals simultaneously is unwise because, at a fixed point in the solution space, *different goals do not go in the same direction*, as shown in Fig. 3.18.

We propose to transfer Eq. (3.4) into an approximate convex optimization problem which requires constructing the feasible Karush-Kuhn-Tucker conditions so that any locally optimal solution is close to globally optimal (see Eq. (3.5)).

$$\begin{aligned} \min \quad & \|\Phi_\theta(\mathcal{X}) - \mathcal{Y}\|^2 \\ \text{s.t.} \quad & \\ & B\Phi_{phy}(\mathcal{X}) \leq b, \Phi_\theta(\mathcal{X}) \approx \Phi_{phy} \end{aligned} \quad (3.5)$$

There exist two approaches to solving Eq. (3.5), as illustrated in Fig. 3.18, to avoid falling into sub-optimal local minima since not all constraints align consistently with the ideal task performance of PHM. Our thesis proposed that the gradual integration strategy adopted for the “PI algorithm structure” can be also employed during the process of constraint satisfaction.

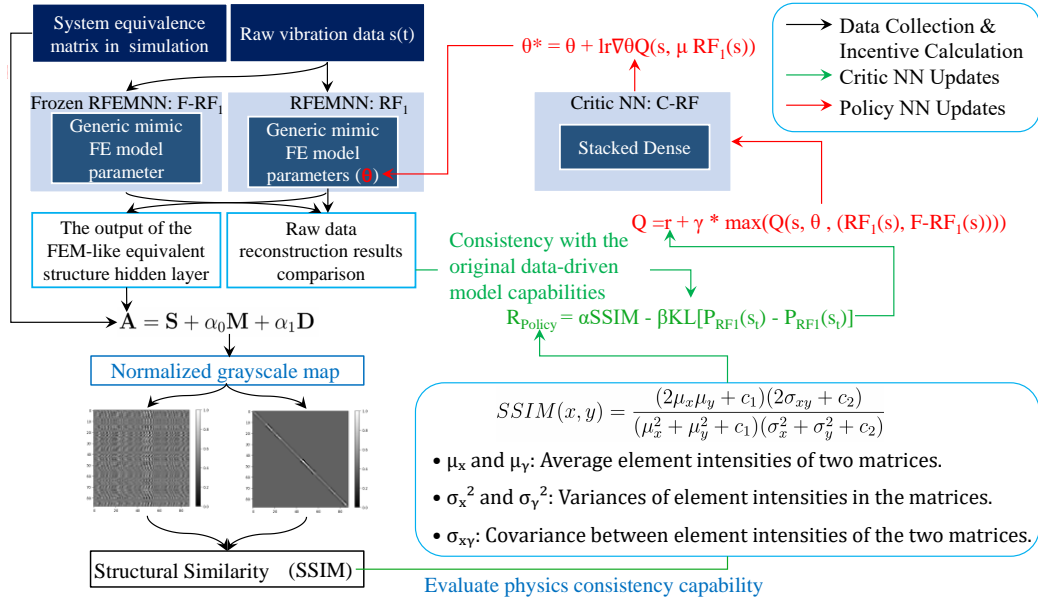
Therefore, we narrow the search space by dividing the problem into multiple manageable stages, as the sequential optimization approach shown in Fig. 3.18. This strategy decomposes the overarching problem into multiple, manageable stages, progressively focusing on achieving the primary objective,  $\Phi_\theta(\mathcal{X}) \approx \mathcal{Y}$ . This is what we called “**Constraint projection**” theory, which involves adjusting a solution to meet a specific constraint by mapping it onto the nearest point that complies with the constraint’s stipulations.



**Figure 3.18:** Learning strategies with different solution ideas.

### 3.3.2 Enhancing RFEMNN's performance in few-shot learning with "Constraint projection" theory

In this section, we realize the "Constraint Projection" method using a deep deterministic policy gradient (DDPG) based reinforcement learning (RL) approach. As illustrated in Fig. 3.19, the RL-driven constrain theory approach leverages the RFEMNN model restricted to the path leading to the specific output, "Output<sub>3</sub>." To ensure that the model maintains its diagnostic, localization, and temporal feature reconstruction capabilities while preserving physical consistency, a pre-trained and frozen RFEMNN, also limited to the "Output<sub>3</sub>" path, is used as a benchmark for data reconstruction during the calculation of policy rewards ( $R_{\text{policy}}$ ).



**Figure 3.19:** Workflow for enhancing RFEMNN performance using constraint projection driven by reinforcement learning.

This setup is designed to maintain the fidelity of the originally trained RFEMNN while avoiding computationally expensive procedures. The reward function,  $R_{\text{policy}}$ , is derived from evaluating the Structural Similarity Index (SSIM) and Kullback-Leibler (KL) divergence between the RFEMNN's outputs and the benchmark, assessing both the consistency of the reconstructed data and the adherence to physical principles.

In addition, a Critic Neural Network (NN) is employed to estimate the Q-value for a given state-action pair. This Critic NN is composed of three dense layers, each with 64 neurons, where the first two layers utilize the ReLU activa-

tion function, and the final layer outputs a single Q-value without any activation function. The current state in this RL framework is the raw vibration data  $s(t)$ , and the action is the output of the RFEMNN ( $RF_1$ ), which includes both the FEM-like equivalent hidden layer output and the raw data reconstruction results.

The optimization process for the RFEMNN follows a continuous three steps loop:

1. **Data collection and incentive calculation.** The RFEMNN ( $RF_1$ ) processes the raw vibration data  $s(t)$ , generating outputs that are then compared with those from the frozen RFEMNN and the system equivalence matrix obtained from the simulation. The reward policy  $R_{\text{policy}}$  is computed using SSIM and KL divergence, focusing on maintaining the original RFEMNN’s diagnostic accuracy and physical consistency.
2. **Critic NN updates.** The Critic NN estimates the Q-value by analyzing the current state and the RFEMNN’s output, learning to predict the long-term cumulative rewards  $Q$ .
3. **Policy RFEMNN updates.** Finally, the RFEMNN’s parameters are adjusted using the policy gradient method, following the update equation for  $\theta^*$ , to maximize the expected Q-value. This process iteratively refines the model parameters to align with  $\theta^*$ , enhancing the model’s performance and consistency.

#### 3.3.3 Application of constrain theory: Few-shot learning for rotor fault diagnostics

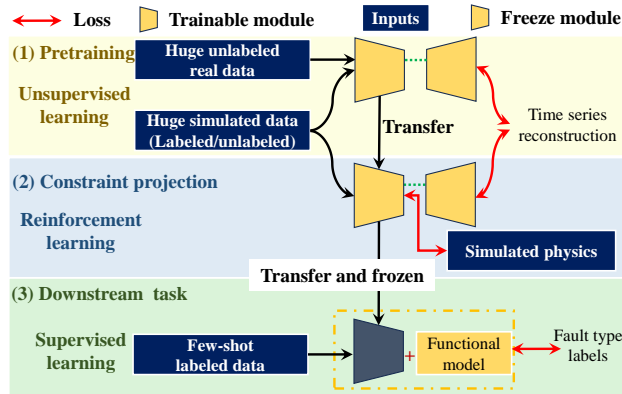
We implement the aforementioned approach to enhance few-shot learning in the RFEMNN model, as shown in Fig. 3.20. The traditional two-step few-shot learning approach [174] is extended into a novel three-phase framework that integrates unsupervised learning, physically-informed reinforcement learning (RL), and supervised fine-tuning.

In the initial pretraining phase, unsupervised learning is applied to a large dataset consisting of both unlabeled real vibration data and simulated data. This allows the RFEMNN to acquire general feature representations, where the “Generic mimic FE model” with only the “Output 3” path as the encoder, as illustrated in Fig. 3.20, learning the encoding features through raw signal reconstruction.

The second phase introduces an innovative “Constraint projection,” wherein deep DDPG is employed on the RFEMNN, as depicted in Fig. 3.19. This step fine-tunes the model to ensure physics consistency by comparing the RFEMNN output with our simulation data trained diagnostic model (the frozen RFEMNN) and



the system equivalence matrix derived from simulation, utilizing our policy reward function as explained in Section 3.3.2. Thus, by practicing our “Constrained projection” theory, we ensure that the selected points on the diagnostic performance loss surface are as close as possible to physically consistent parameter values. This approach systematically aligns the parameter space with the physical constraints while maintaining the diagnostic performance within acceptable bounds.



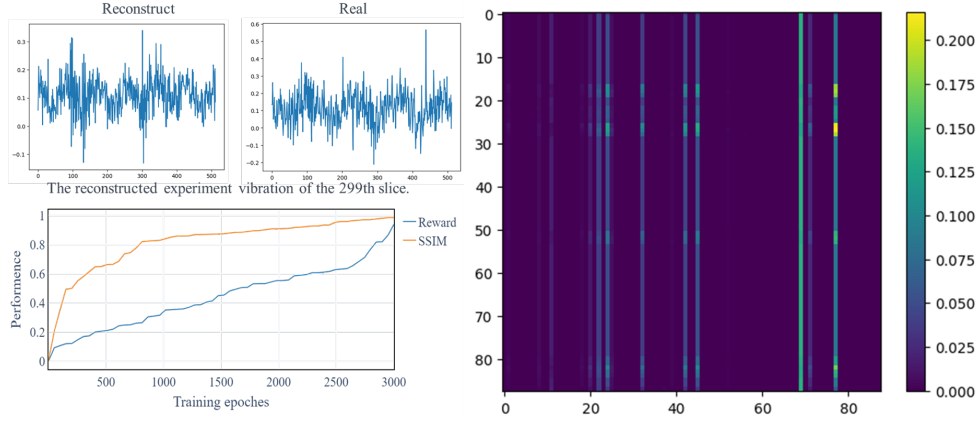
**Figure 3.20:** Applying RL-driven constraint theory in RFEMNN model’s few-shot learning.

The final phase involves supervised learning on a small set of labelled real fault data with the frozen feature encoder- the “Generic mimic FE model” part in RFEMNN, effectively leveraging the rich, physics-consistent representation mappings developed in the preceding steps. We argue that the method effectively addresses the few-shot learning challenge by employing a soft constraint, which mitigates the rigidity associated with the direct embedding of physical knowledge. This approach utilizes a heuristic that prioritizes diagnostic performance by separating primary and secondary contradictions. Initially, the model focuses on maintaining diagnostic accuracy, and then it gradually aligns with physical consistency, freezing this structure to preserve consistency. This process ensures that the model retains insights from unlimited simulated samples without re-learning aspects that cannot be derived from such limited data.

### 3.3.3.1 Physics consistency validation results

To assess the RFEMNN’s physics consistency, we closely monitor the hidden layer weights and reward loss changes during the learning process of the “Generic mimic FE model” part, whose inside weights and connections change during the pre-training and intermediate task phases. The corresponding results are presented in Fig. 3.21.

### 3.3. Constraint projection for PIML in few-shot faults diagnostics



**Figure 3.21:** Vibration reconstruction (left) and physics consistency (right) investigation after RL fine-tuning.

Regarding Fig. 3.21, “Reward” and “SSIM” curves illustrate a dynamic balance between satisfying physics-based knowledge and achieving data-driven signal reconstruction accuracy. The discrepancy between these curves represents the KL divergence of the reconstructed time series results, comparing the Policy RFEMNN model  $RF_1$  with the frozen model  $FRF_1$ , effectively denoting the mean squared error (MSE) in time series reconstruction. By the end of the fine-tuning process, the MSE reduces to 0.04, confirming the high similarity between the real and reconstructed vibration signals. Initially, the hidden layer outputs exhibited distinct channel characteristics with significant differences, which were misaligned with the physics matrix’s band symmetry and diagonal distribution. However, after the RL process, the PIML output closely aligns with the ideal, as evidenced by the SSIM of 0.984 and a reward of 0.941. This improved alignment is shown in the square matrix colour map on the right side of Fig. 3.21.

In particular, we observe that the gap difference between “Reward” and “SSIM” is non-linear, increasing initially and then decreasing. This pattern reflects a corresponding rise and fall in KL scatter. This suggests that the incorporation of physics impacts the ability of the initially learned data-driven model to capture insights relevant to the PHM task. The subsequent decrease shows that without simultaneously optimizing for multiple objectives, including physical consistency, conflicting optimization directions are reduced, leading to a more optimal position on the loss plane.

#### 3.3.3.2 Rotor few-shot diagnostic results

The few-shot learning capability of the proposed method is validated under zero-shot, one-shot, and few-shot settings, where zero-shot uses only simulated faults,

one-shot includes a set of real faults and simulated data, and few-shot employs one of 10 folds of original experimental data.

**Table 3.2:** Diagnostic results across various limited-labelled-data scenarios.

Z: Zero-shot, O: one-shot, F: Few-shot, H: Healthy, U: Unbalance, C: Crack, U&C: Unbalance and Crack, LA: Location accuracy. We mark the correct results in orange.

		Diagnostics results														
		H			U			C			U&C			LA(%)		
		Z	O	F	Z	O	F	Z	O	F	Z	O	F	Z	O	F
Real Result	H	0	680	1255	1102	1134	560	0	1	0	713	0	0	-		
	U	0	245	8	1097	2074	2285	0	1	63	1259	36	0	65.79	92.0	99.2
	C	0	2	0	155	61	0	0	262	411	256	86	0	51.79	84.8	97.1
	U&C	0	0	0	36	158	0	0	2	0	1769	1645	1805	-		

The results in Table. 3.2 reveal that zero-shot learning achieves a diagnostic accuracy of 44.87%, classifying 2866 real samples correctly, but tends to misclassify due to simulation configuration biases, often labelling faults as unbalanced or combined. One-shot learning, enhanced by adding a single real fault data, improves diagnostic performance, notably in crack fault identification. Few-shot learning proves most effective, reaching an accuracy of 90.7% and improving fault types identification, although misclassifications and false alarms for unbalanced and healthy samples persist. Enhanced data slice length and careful selection of simulated samples are suggested to boost diagnostic accuracy and reliability. Despite limitations in zero-shot and one-shot scenarios, the model maintains partial diagnostic capability for fault location, benefiting from the pretraining on the simulated data.

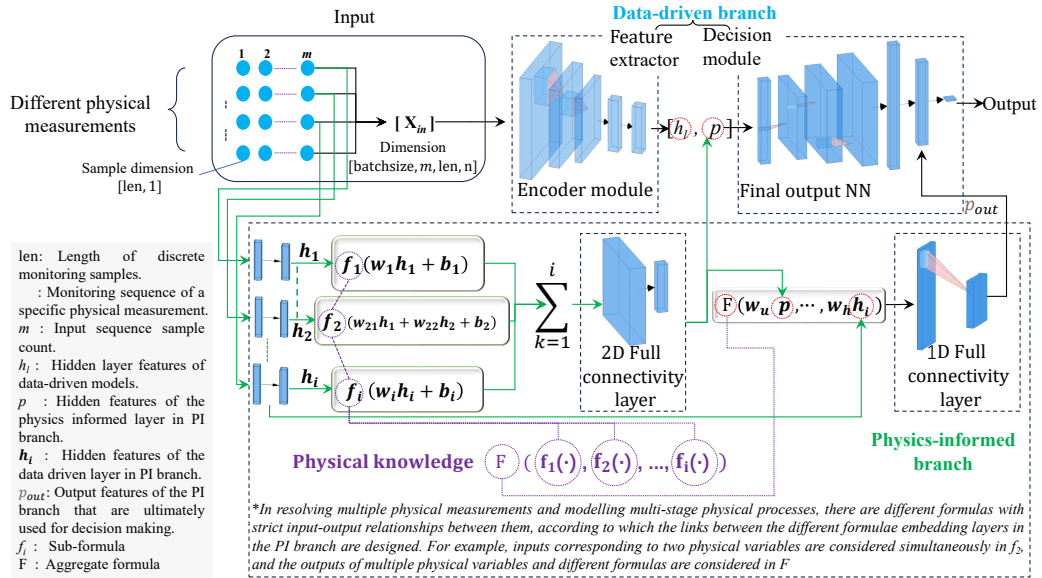
### 3.4 A generic PIML framework

This section proposes a generic PIML architectural framework from an engineering perspective, aiming to integrate the core principles of “Physics” and “ML” into a practical scheme. By reviewing the advantages and disadvantages of “Constrained projection” and “Mimetic theory” based on previous cases, we identify key characteristics for an effective PIML approach: (1) **gradual embedding of constraints within both the ML structure and the training methodology with independence in structure and training**, and (2) **avoiding conflicts between ML and embedded physics by solving the problem sequentially**. With these principles, we refine and simplify the essence of both theories.

Therefore, for “Mimetic theory,” we adopt the parallel architectural approach proposed in Section 3.4.1, viewing **the PI layers as the extension of the model’s original data flow**. To streamline the complex optimization associated with “Constrained projections,” we propose the simplified, **branch-independent and three-step training strategy** outlined in Section 3.4.2. Since the PI branch and the original data-driven part of the PIML model are designed as parallel architectures, **different parts of the model can satisfy distinct optimization objectives in different steps**. This design retains the core concept of calibrating physical consistency based on the ML model’s performance, thereby **establishing the lower performance bound of the PIML model**. Finally, we validate the proposed “Generic PIML framework” on battery remaining discharging cycles prediction, using small early life-stage charging-discharging cycle data in Section 3.4.3.

### 3.4.1 A two-branch parallel PIML framework

The proposed PIML model employs a dual-branch, end-to-end framework, as illustrated in Fig. 3.22. This hybrid model consists of a PI branch and a data-driven branch. The data-driven branch can function independently or in conjunction with the PI branch. PI branch augments the data-driven model by contributing features  $p$  to its latent space and outputs features  $p_{out}$  for final decisions. The PI branch cannot independently generate reliable predictions.



**Figure 3.22:** Embedding physical knowledge into neural network models using parallel architectures.

The data-driven and PI branches share inputs but process them through distinct mechanisms. The data-driven branch integrates all measured samples within a batch of dimensions  $[batch.size, m, len, n]$  through an encoding module designed to extract latent trends. Conversely, the PI branch processes finite physics-based measurements through individual formulas  $f_i(\cdot)$  and specific inter-layer connections as depicted in Fig. 3.22. These connections, tailored based on the relationships between various equations and parameters, culminate in combined feature tensors. PI branch is constructed based on the ‘‘Mimetic theory’’, and the decomposition of the formulae is carried out by placing the PI layer in the data processing, depending on whether the operator is more biased towards processing the input raw data or outputting decision-relevant features.

The data-driven branch’s end-to-end (E2E) deep learning paradigm directly processes raw data to get PHM results without requiring manual signal processing or expert domain intervention. The final feature representation is processed through a dense layer with  $d_1$  neurons to get the feature  $h_l$ .

In the output part, the final ‘‘Full connectivity neural network (FCNN)’’ considers the PI branch’s output  $p_{out}$  in the decision-making process and the data insights from feature  $p$ . FCNN is tailored for the many-to-one process to generate the output of the remaining discharging cycles. The overall framework is modeled and explained in detail in Appendix. C.8.

### 3.4.2 Pretrained-physics alignment multistep training

We present a branch-independent learning paradigm tailored for the proposed dual-branch model, comprising three phases as illustrated in Fig. 3.23. This three-step approach underscores the necessity of starting with a pre-trained ML model, ensuring the PI-ML model’s performance remains at least equivalent to that of the standalone ML model.

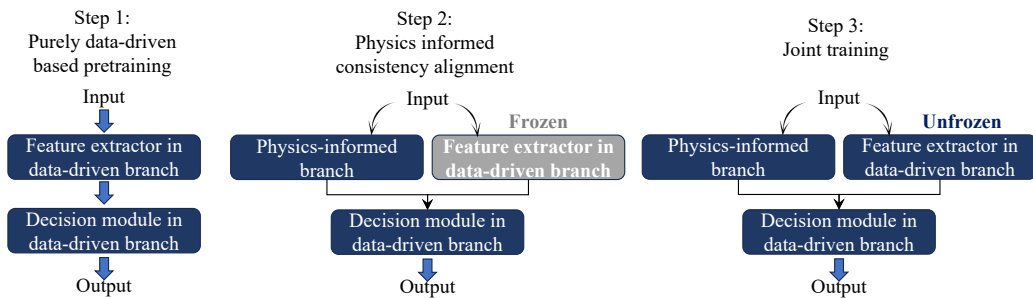


Figure 3.23: Novel PIML learning strategy.

**Step 1: Data-driven pre-training**, we train only the data-driven branch,

setting the PI-branch parameters as untrainable and initializing them with zero-mean to ensure outputs  $p$  and  $p_{out}$  are 0. This phase focuses on optimizing the data-driven branch independently.

**Step 2: Physics-informed alignment** involves loading and freezing the pre-trained data-driven branch. The PI branch’s parameters are then initialized randomly and made trainable, functioning with the data-driven branch as a fixed feature extractor. This setup allows the PI branch to align physically with the pre-trained data, enhancing the model’s coherence.

**Step 3: Joint training** is initiated upon reaching convergence in the data-driven branch and initial training in the PI branch. Both branches are then made trainable to fine-tune the entire model towards a global optimum. This step not only addresses inconsistencies between the branches but also enhances the model by integrating data-derived insights with embedded physical knowledge.

*This innovative training strategy allows us to maintain the original architecture’s data-fitting capabilities and extends PIML applicability to a broader range of scenarios by leveraging mature data-driven models.*

### 3.4.3 Case study: Battery RUL prediction using small early-stage lifecycle data

We develop the SEI-DCN model by incorporating a battery solid electrolyte interphase (SEI) health indicator into the Physics-Informed (PI) branch, as illustrated in Fig. 3.24. This model leverages formula-embedded layers, guided by the SEI growth model proposed by Attia et al. [175], to inform the design of the Dilated CNN (D-CNN) architecture. Further details on the development and structure of the SEI-DCN model can be found in Appendix C.8.

The SEI growth formulas are shown in Eq.(3.6) and (3.7). In Eq. (3.6),  $D$  is the SEI degradation parameter.  $D_0$  represents the initial or baseline SEI degradation parameter.  $E_a$  is the effective activation energy for SEI growth, determined by the Arrhenius relation.  $k_B$  denotes Boltzmann’s constant, a fundamental physical constant.  $T$  is the monitoring temperature of the battery, reflecting the exponential relationship of SEI degradation in accordance with the Arrhenius equation’s principles on temperature-dependent chemical processes.

$$D = D_0 e^{-\frac{E_a}{k_B T}} \quad (3.6)$$

The  $p_{out}$  of is estimated using the equation:

$$p_{out} = \frac{C}{D} + b \quad (3.7)$$

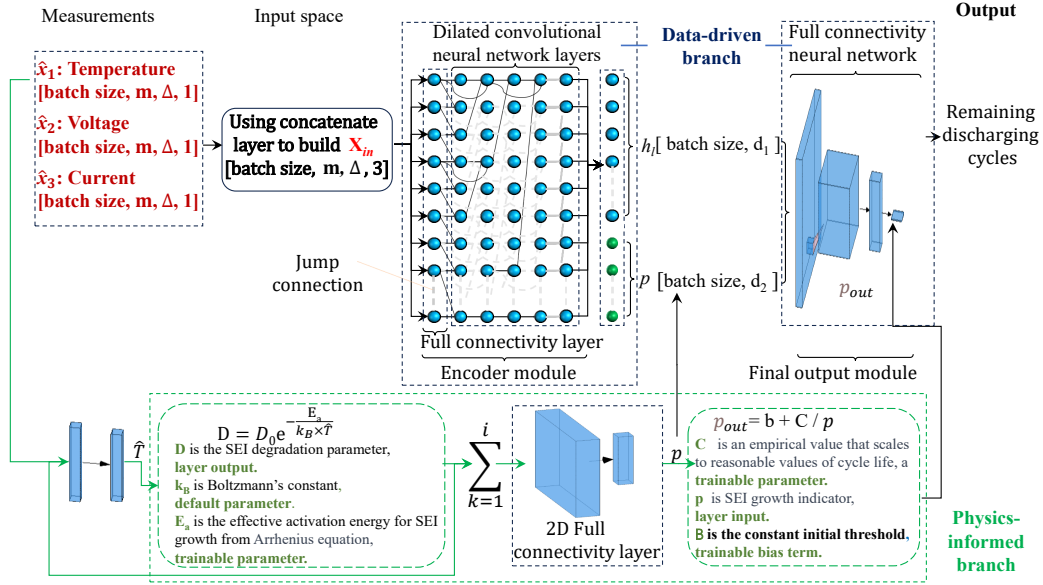


Figure 3.24: SEI informed DCNN model diagram.

In this context,  $p_{out}$  refers to an approximate lifespan indicator, as it is not the final RUL output of the proposed PIML model. This distinction arises due to the simplified nature of the models, often incorporating empirical parameters and assumptions that do not fully capture the complexities of actual battery operation and degradation phenomena. This formulation emulates the Paris laws in material fatigue and posits that a battery's cycle life is inversely proportional to the rate of SEI degradation, adjusted by the empirical factor  $C$ . Considering the input-output relationships and the logical relationship between Eq.(3.6) and (3.7), only the temperature data are taken as the input of the PI branch. and the cross-layer connections are used in Fig. 3.24. Eq.(3.6) and (3.7) are the activation functions of these custom design layers.

A standard D-CNN model developed by Hong et al. [176] served as a benchmark model for investigating the superior performance of SEI-DCN. It was also used as the data-driven branch in SEI-DCN to design a controlled variable comparison test. The selected D-CNN structure had already been validated on our testing dataset. The critical difference between our approach and the selected D-CNN is the addition of a PI branch and our multi-step training method.

We conduct the proposed controlled variable comparison test on the fast-charging lithium-ion battery dataset developed by the MIT-Stanford team and described in the article [175]. This dataset comprises three experiments. The battery sets from "2017-05" and "2017-06" are set up for model training and testing. Initially, a random data splitting method equally divides the dataset into training

and testing subsets. This involves shuffling the dataset indices before splitting them into two groups to ensure a balanced partition. As clarified in [175], the battery sets recorded in “2018-04” are different from the sets in “2017-05” and “2017-06.” It can be used as the second test set to evaluate the model’s generalization ability on new data. More details of this controlled variable comparison test are shown in Appendix C.9.

### 3.4.3.1 Comparison results with SOTA models

Table 3.3 presents a comparison of the proposed model’s performance against several SOTA models from the literature. Notably, the DCNN model from [176] has been reproduced and specifically optimized in this study, serving as the basis for an ablation analysis. Table 3.3 highlights significant advancements in pre-

**Table 3.3: Performance of different models.**

Model	Required cycles length	Predicted cycle error (MAE)
Shallow MLP [176]	4	150
MLP [176]	4	174
Full model [177]	100	99
Variance [177]	100	112
SVR [176]	100	245
CNN [176]	4	82
CNN + LSTM [176]	4	72
DCNN in [176]	4	65
Our optimized DCNN	4	55
<b>Proposed SEI-DCN</b>	4	<b>15</b>

dicting cycle errors for battery life estimation, with the proposed parallel model standing out by achieving the lowest predicted MAE of 15 cycles. This outcome is achieved using a data sample truncated to 2500 seconds, typically spanning 3 to 4 cycles. It demonstrates that our SEI-DCN can deliver more precise usage insights with short-term data, enhancing the adaptability of prediction algorithms to real-time user habits. In addition, using a platform equipped with a single 16 GB NVIDIA T4 Tensor Core GPU, the prediction time for one test sample is approximately  $3.5 \times 10^{-5}$  seconds.



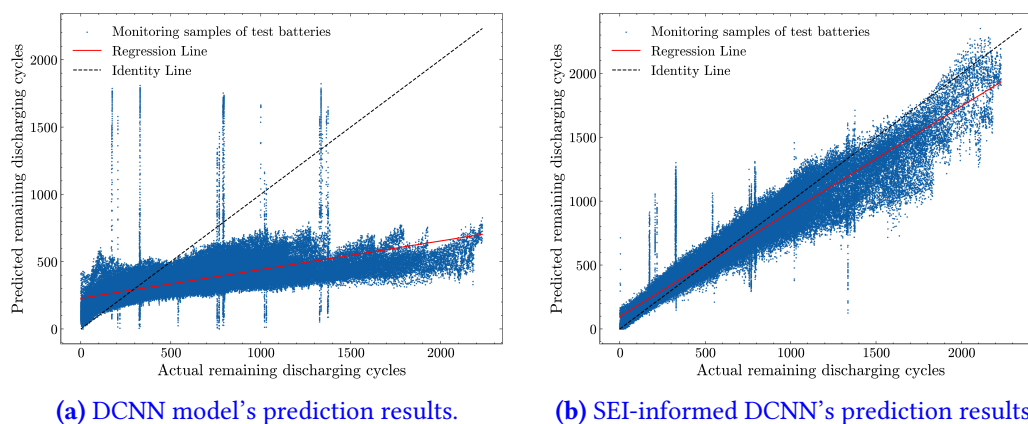
### 3.4.3.2 Comparison results of prediction performance on a new dataset without re-learning

To demonstrate the generalizability and adaptability of the proposed SEI-informed DCNN model, we evaluate its performance on a new dataset collected during the experiment “2018-04”, without any additional retraining. The results are then compared with those of the optimized DCNN model, as illustrated in Fig. 3.25.

In Fig. 3.25, the “Identity line” represents perfect predictions where predicted values match true values. The “Regression line” reflects the overall prediction trend. Ideally, the “Regression line” should align with the “Identity line”, indicating that the model’s predictions are accurate. The blue scatter points in the figure represent the actual predictions made by the model. Through this generalization ability test, several key observations can be made, as illustrated in Fig. 3.25:

1. It is clear that although the purely DCNN model captures the overall trend, there is a significant difference between the predicted and actual values, indicating that the accuracy of the model could be significantly improved.
2. Considering that the RUL span in the training set does not include the case where the remaining discharging cycle life is greater than 2000, it can be seen that DCNN does not have the prediction ability in that case. Still, SEI-DCN does match the real data significantly better, which proves that SEI-DCN has a superior generalization ability by embedding physical knowledge.
3. The predictions of the SEI-DCN model (our approach) have higher variability on this new dataset, along with outliers with relatively large deviations from the true values in the deep degradation stage. However, its performance declined much less than that of the DCNN.
4. The SEI-DCN model’s MAE is 81. Whereas the pure DCNN is 311, which quantifies the significant generalization advantage our the proposed physics-informed approach.

Moreover, in Appendix C.9, we provide a comprehensive performance evaluation and an investigation of SEI-DCN’s action mechanism. We conduct an **in-depth comparison between the DCNN and SEI-DCN models**, focusing on prediction accuracy and error distribution, where the data-driven branch of SEI-DCN mirrors the structure, parameters, inputs, and outputs of the DCNN. The results show that both models capture the linear decay trend of remaining discharging cycles; however, SEI-DCN delivers more accurate predictions with fewer outliers, underscoring the benefit of integrating physics-based knowledge with data-driven approaches. **Detailed error analysis via cell-by-cell box**



**Figure 3.25:** Predicted trajectory results for the new test battery packs’s remaining discharging cycles.

**plots** reveals SEI-DCN’s superior performance, particularly in early and late battery life stages, with a narrower error distribution and enhanced robustness. **Further analysis of the feature integration and branch decision fusion process within SEI-DCN, as demonstrated by channel weight visualizations**, confirms the successful unification of data-driven and physics-informed features. This unification, achieved through a three-step training process, optimizes the sparse distribution of model channel weights across both branches in the knowledge fusion process. Additionally, **the model’s flexibility was assessed by altering the embedded physics knowledge within the PI branch while keeping the data-driven branch constant**, with consistent improvements observed across different physics models, highlighting the model’s adaptability and generalization capabilities even when the physics knowledge does not most directly align with RUL metrics.

### 3.5 Summary

This chapter introduced a flexible PIML architecture framework for integrating physics knowledge with implicit analytical relations and incomplete parameter estimation. It began with a simulated bearing degradation scenario, where the similarities, differences, and risks associated with various PIML approaches were analyzed. The embedded empirical “stiffness-vibration” relations were then verified to enhance the TCN’s performance in bearing RUL prediction, and the mechanisms underlying this improvement were explored, offering tutorial guidance in building different PIML models. The conclusions drawn suggest that unifying existing PIML methods under a “PI algorithm structure” is a better choice.

Inspired by this unification, the “Mimetic theory” was proposed, advocating for gradually embedding comprehensive knowledge into sub-components of the ML algorithm structure to enable a more controlled PIML paradigm during training and result improvement.

This theory was applied to rotor fault diagnosis, using the total mass unit of a rotor finite element as a mimetic basis to simulate inverse dynamics via vibration signals. A band symmetry matrix forced sparsity and geometric specificity in the interlayer weight distribution of the rotor finite element model, integrated into the RFEMNN for diagnosing fault types and locations. Experimental results demonstrated that the RFEMNN model outperformed traditional data-driven algorithms across various rotor structures and speeds, establishing it as the first neural network capable of simultaneously identifying fault types and locations. However, mechanistic analysis revealed that the embedded physical knowledge tended to degrade during training, reducing physical properties and scalability, especially in few-shot scenarios. To address this, the focus shifted from structural design to learning strategies, incorporating “constraint projection” theory from optimization into PIML. This approach distinguished between data-driven and physical consistency goals during training, optimizing structural parameters through reinforcement learning to maintain physical integrity without compromising learning objectives. Through pre-training on simulated data and physical consistency alignment, the constraint projection-enhanced RFEMNN’s few-shot ability in compound fault diagnostics for rotors.

The “Mimetic theory” was further refined through a parallel structural design model with a physics-informed branch and a data-driven branch, trained using a three-stage strategy as a simplified “constraint projection:” pre-training the data-driven branch, training the PI branch while freezing the data-driven output, and then jointly training both branches. This multi-step process enhanced cooperation between branches, approximating the global optimum while preserving local optima. This strategy was applied to RUL prediction for fast-charging Li-ion batteries using an SEI growth-informed Dilated CNN model, reducing the required charge/discharge cycles from 65 to 4, improving prediction accuracy by at least 20 cycles, and demonstrating the robustness and resource efficiency of the generalized PINN design strategy in a new-scenario test and comprehensive mechanism analysis.

This chapter explored different perspectives on how “ML” and “Physics” interact. The “Mimetic theory” aimed to integrate physics within the ML framework to achieve physics-consistent data flow, ensuring structural and output congruence. While most PIML research focuses on innovations in structural and constraint design, often through customized approaches, standardized solutions were provided. The “Constrained projection” theory refined the optimization problem by aligning the divergent optimization directions of ML and Physics,

guiding PIML model parameters toward an optimal solution, a challenge that has been recognized but insufficiently addressed in existing research. To our knowledge, developing a “Generic PIML architecture” remains unexplored due to its dependence on expert experience, lack of established architectures, training guarantees, and difficulties in convergence and effective training, for which we propose a viable solution. These works lay a solid foundation for the generic PHM framework that will be presented in Chapter 5.

# Improved SSL for PHM

---

## Contents

<b>4.1</b>	<b>Introduction</b>	<b>84</b>
<b>4.2</b>	<b>Novel contrastive SSL in sequential mining for RUL prediction</b>	<b>84</b>
4.2.1	Existing contrastive SSL paradigm problem statement	85
4.2.2	Novel contrastive SSL	87
4.2.2.1	Contrastive pairs design	88
4.2.2.2	Proposed contrastive SSL strategy	88
4.2.3	Application of the proposed contrastive SSL for accurate bearing RUL prediction	90
4.2.3.1	Proposed model	90
4.2.3.2	Result discussion	93
<b>4.3</b>	<b>Considering downstream information in SSL pretext task design</b>	<b>95</b>
4.3.1	Analysis of the proposed SSL drawbacks	95
4.3.2	Deep enhancement of the contrastive SSL	97
4.3.3	Validation of the improved contrastive SSL model on bearing prognostics	100
4.3.3.1	Benchmark results	101
4.3.3.2	Validation results of the proposed method	103
<b>4.4</b>	<b>Summary</b>	<b>107</b>

---

This chapter is based on the following publications:

- J5 **WeiKun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJAHHER. “Bearings RUL prediction based on contrastive self-supervised learning.” *IFAC-PapersOnLine* 56.2 (2023): 11906-11911.
- J3 **Weikun DENG**, Khanh T.P.NGUYEN, Christian GOGU, Jérôme Morio, Kamal MEDJAHHER (2024). Enhancing Prognostics for Sparse Labeled Data Using Advanced Contrastive Self-Supervised Learning with Downstream Integration. (Submitted to *Engineering Applications of Artificial Intelligence*, Accepted).

## 4.1 Introduction

In the previous chapter, we explored how PIML models can enhance ML performance by integrating domain-specific physics knowledge, particularly in scenarios with limited data and incomplete physical parameters. However, a significant limitation identified was the reliance on supervised learning, which is often constrained by the scarcity of labeled data. To address this challenge, Chapter 4 shifts the focus to SSL as a strategy to leverage vast amounts of unlabeled data. The main motivation of this chapter is to explore how SSL can enhance ML performance in real-world applications, particularly in PHM, by developing models that are more flexible, adaptable, and capable of learning from diverse data sources.

This chapter has 4 sections, they are structured to explore and validate the application of SSL strategies in PHM, which includes two major contents.

It begins with an introduction to novel contrastive SSL methods in Section 4.2, where the limitations of existing contrastive SSL approaches are identified, setting the stage for constructing the new one. This approach is validated through a detailed case study in Section 4.2.3, where a CNN-LSTM model is built and tested on the “PRONOSTIA” bearing dataset, filling the gaps of contrastive SSL in RUL prediction. The chapter then progresses to a deeper enhancement of the proposed contrastive SSL approach in Section 4.3, advancing the prediction focus from normalized RUL to actual RUL. This section highlights the importance of aligning SSL pre-training tasks with specific downstream prediction tasks, representing a significant theoretical advancement in the development of SSL models for PHM.

## 4.2 Novel contrastive SSL in sequential mining for RUL prediction

In this section, we explore how SSL can enhance ML models by leveraging unlabeled data, complementing the physics-informed approaches discussed in the previous chapter. By utilizing SSL, we aim to increase data availability and address “ill-posed” problems where labeled data is limited. To achieve this, we develop a transformation that maps the original input data into a new representation space, allowing the model to automatically extract relevant PHM information directly from the data structure. This transformation is implemented using contrastive SSL, which trains the model to distinguish between different states by ensuring that their representations are distinct, even if their observed data appear similar. The contrastive loss function, defined in Eq. (4.1), encourages the

model to separate the representations of different states by penalizing similarities between them.

$$\mathcal{L}_{\text{contrast}}(\mathcal{T}) = \frac{1}{N} \sum_{i,j} \max(0, |\mathcal{T}(\mathcal{A}^*(x_i)) - \mathcal{T}(\mathcal{A}^*(x_j))|) \quad (4.1)$$

where  $N$  is the total number of pairs  $(i, j)$  considered,  $x_i$  and  $x_j$  are distinct input states.  $\mathcal{L}$  encourages  $\mathcal{T}$  to map different states to distinct representations in  $\mathcal{Z}$ , even if they yield similar observations in  $\mathcal{Y}$ , pushing apart the representations of data or knowledge into different states. Here,  $\mathcal{A}$  represents the mapping from the observed data to the system's health status, and  $\mathcal{A}^*$  is a specific part of this mapping. By employing this approach, we aim to improve the model's ability to analyze sequential data for fault detection and prediction without relying on labeled data.

The traditional approaches of contrastive SSL, presented in Subsection 4.2.1 have been effectively applied in fault diagnostics, where it helps identify distinct faults within a system. However, when these methods are adapted for RUL prediction, they uncover deeper insights within the time series structure that are often overlooked. These insights stem from the unique characteristics of time-series data, where information is distributed along the time axis across different health states. This temporal distribution is crucial for accurately predicting early wear and tear as well as long-term degradation.

Building on this foundation, we introduce a novel contrastive SSL approach in Subsection 4.2.2 to enhance RUL prediction performance. This new method is tailored to be more effective in real-world applications, particularly in bearing health monitoring, as demonstrated in Subsection 4.2.3.

### 4.2.1 Existing contrastive SSL paradigm problem statement

Traditionally, the contrastive SSL learning process involves the following steps:

#### 1. Pretraining:

- (a) *Contrastive pair selection* - Choose pairs of distinct input states  $(x_i, x_j) \in \mathcal{Z}$  from the unlabeled dataset. These pairs should ideally represent varied conditions or states of the system to enhance the learning of separability in the feature space.
- (b) *Distance computation* - Calculate the distance between the transformed representations of each pair  $(\mathcal{T}(\mathcal{A}^*(x_i)), \mathcal{T}(\mathcal{A}^*(x_j)))$ .

- (c) *Optimization* - Compute the contrastive loss using the specific contrastive loss function defined by Eq. (4.1). Update the parameters of the transformation  $\mathcal{T}$  by minimizing the contrastive loss.

**2. Downstream-training:**

- (a) *Model initialization* - Initialize  $\mathcal{A}$  with the pretrained’s  $\mathcal{A}^*$ ’s weights. Freeze  $\mathcal{A}^*$  to keep its learned representation ability in downstream.
- (b) *Fine tuning* - The rest part  $\mathcal{A} \setminus \mathcal{A}^*$  in  $\mathcal{A}$  is fine-tuned by training it on a labeled dataset,

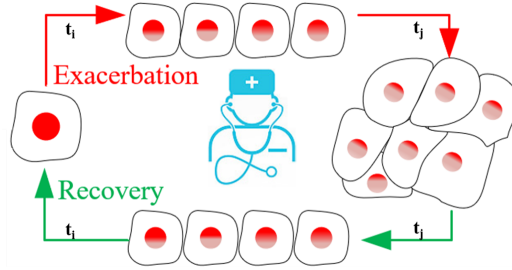
In both phases, pre-training on unlabelled data  $\mathcal{Z}$  forms the foundation for the directed representation learning. Fine-tuning on labeled data allows the model to further refine its use of the learned comparison features. Central to this process are two critical components: “Contrastive pair selection” and the design of a “Distance computation” task for  $\mathcal{T}$ . A key challenge arises by **a) ensuring that data pairs represent distinct system conditions**, enhancing feature space separability—an empirically driven setup with inherent robustness issues. Additionally, traditional contrastive learning methods, such as sequence masking, data rotation, and future state prediction, struggle with a core challenge: **b) ensuring that  $\mathcal{T}$  aligns with degradation-related features**. To address this, many approaches empirically enforce a pretext task logic that keeps temporally close samples close in feature space while pushing temporally distant samples apart, thereby establishing a connection to degradation.

There is a fundamental logical conflict between *the two empirical settings* discussed above. First, most degradation processes exhibit periodic behaviour, making it challenging to accurately determine which two data segments the samples should be drawn from and what degradation stages they represent. Even when a methodology is established, it often becomes specific to the machine or case, losing adaptability when the objective changes. Second, anchoring sample differences to temporal proximity can indirectly indicate degradation. However, this approach is undermined by the weak correlation and the significant impact of noise in monitoring measurements, particularly in high-frequency sampling scenarios. For instance, in vibration-based prognostics, vibration levels often remain stable throughout much of the degradation phase, resulting in minimal variation in characteristic differences between periodic data samples, despite changes in temporal distance. Up to these points, in the next section, we suggest shifting the focus from merely selecting samples to actively constructing them using contrastive manipulations of the same sequence.



### 4.2.2 Novel contrastive SSL

The conceptual framework presented in this study draws inspiration from the medical diagnostic process illustrated in Fig. 4.1. In this process, healthcare professionals evaluate a patient’s health by analyzing sequential medical images acquired at various disease stages in conjunction with pertinent medical records. While there may be similarities in cellular morphology concerning shape and size between different points in the exacerbation ( $t_i \rightarrow t_j$ ) and recovery ( $t_j \rightarrow t_i$ ) phases, physicians can extract valuable insights by discerning information patterns within sequences of medical records.



**Figure 4.1:** Physiologically degraded monitoring processes.

In ML, akin to medical diagnostics, the focus is on identifying positive and negative sequences generated from a single machine’s monitoring data, rather than analyzing correlations across different time intervals, as shown in Section 4.2.2.1. Sequential features, denoted as  $p_t$ , capture hidden patterns that follow the natural progression from  $t_1$  to  $t_n$ , representing the degradation trend. Conversely, negative sequence features, denoted as  $n_t$ , reveal hidden patterns in reverse order, from  $t_n$  to  $t_1$ , indicating an inverse degradation trend. These patterns are extracted using a self-supervised feature extractor, represented as  $\mathcal{A}^*(\cdot)$ . The key differences between the hidden patterns in  $\mathcal{A}^*(p_t)$  and  $\mathcal{A}^*(n_t)$  provide crucial insights into the degradation process. In SSL, varying input sequences to the feature extractor generate distinct trend information (See Section 4.2.2.2). The primary objective of the pretext task is to maximise the distinction between the extractor’s outputs for different sequential directions, achieved through contrastive loss. This methodology enhances the precision of degradation trend analysis and subsequently improves the accuracy of RUL predictions, which is validated in Section 4.2.3.

During the early stages of failure, the degradation process is typically slow, resulting in subtle differences in the hidden patterns  $\mathcal{A}^*(p_t)$ ,  $\mathcal{A}^*(n_t)$ . However, the degradation rate often accelerates as the system progresses toward failure, leading to more significant disparities between positive and negative sequence

features. The extent of these differences is crucial in determining the rate of change in the system's condition over time.

#### 4.2.2.1 Contrastive pairs design

As shown in Fig. 4.2, we truncate the raw time series into overlapping windows, sliding along the timeline in fixed steps, with slices of data along the positive timing in each window as positive samples, such as the data in  $(t_0, t_{t_0})$ . Negative samples are obtained by inverting the time sequence of positive sequential order samples, e.g.,  $(t_{t_0}, t_0)$ . The whole sequence is divided into  $n$  samples.

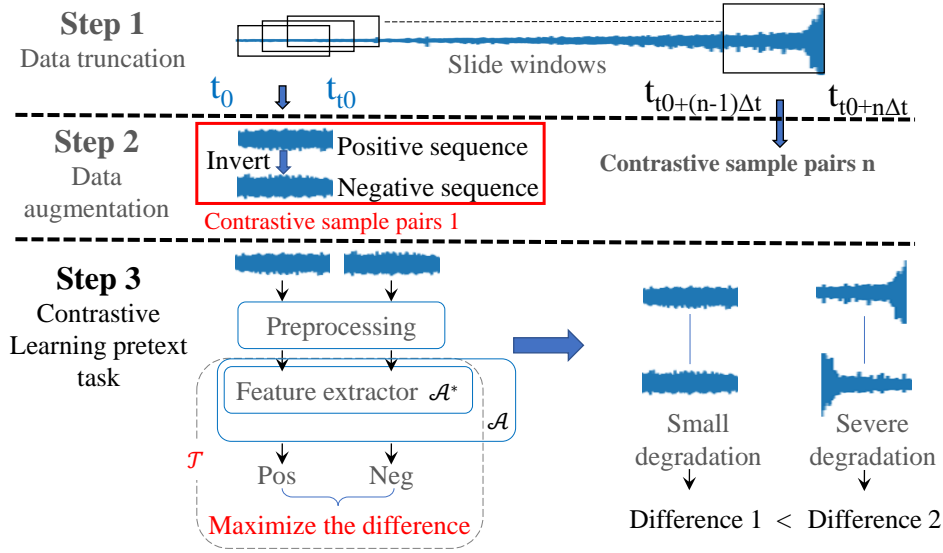


Figure 4.2: Construction of contrastive sample pairs.

#### 4.2.2.2 Proposed contrastive SSL strategy

Eventually, identical data preprocessing operations are applied to each sample pair within each window, and the feature extraction component  $\mathcal{A}^*$  of the PHM model  $\mathcal{A}$  is trained by the transforming task  $\mathcal{T}$ . The objective is to maximize the disparity between the encoded features of positive (*pos*) and negative (*neg*) samples, aiming to learn a degradation-related feature under the metric set  $\mathcal{T}$ , which is physically indicative of the system's state change magnitude over a fixed period across different phases (see Fig. 4.2).

In the context of medical diagnostics, where a fixed level of expertise enables doctors to diagnose a patient's health state, the SSL strategy proposed here

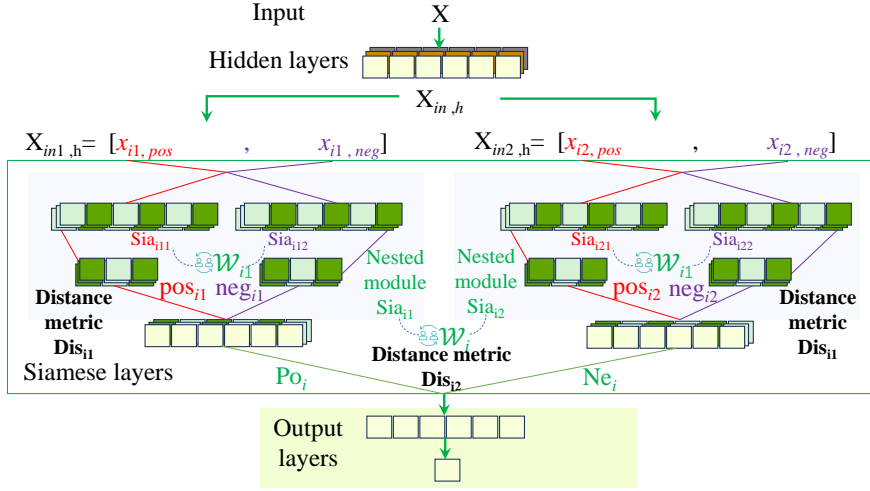


Figure 4.3: Multi-hierarchy Siamese models with nested structural reuse.

replicates this fixed expertise by employing nested structural reuse in Siamese modules. As depicted in Fig. 4.3, the Siamese structure in the hidden layers comprises two parallel branches in  $Sia_{i1}$ , which share weights and process paired inputs independently. When contrastive pairs  $X_i$  are fed into the network, each input, such as  $x_{i1,pos}$  and  $x_{i1,neg}$ , undergoes a separate forward pass through its corresponding branch. Despite the shared weights, each branch generates distinct activations and representations based on the specific features of the inputs.

Furthermore, *the nesting and reusing of Siamese structures facilitates the creation of more complex and hierarchical similarity-based architectures*. By constructing a larger Siamese structure through identical modules  $Sia_{i1}$  and  $Sia_{i2}$  which have the same weights and configuration branches, *developers can establish multi-level comparison systems capable of processing and comparing inputs at varying levels of abstraction*. This hierarchical design is particularly advantageous for tasks requiring multi-scale feature comparison or handling data with inherent hierarchical relationships.

The output representations from different Siamese levels are then compared or combined for tasks such as contrastive learning. To derive a degradation-responsive representation, appropriate distance metrics are designed to constrain the learning of representations at various levels, e.g.,  $Dis_{i1}$  for both  $pos_{i1}$  &  $neg_{i1}$ , and  $pos_{i2}$  &  $neg_{i2}$ .  $Dis_i$  for  $Po_i$  and  $Ne_i$ . We suggest some feasible metrics for designing  $Dis_{i2}$  and  $Dis_{i2}$  in Appendix D.1.

### 4.2.3 Application of the proposed contrastive SSL for accurate bearing RUL prediction

This section validates the proposed novel contrastive SSL by building the CNN-LSTM model with the test on “PRONOSTIA Bearing Dataset.” It is generated from the running-to-failure experiment on the bearing aging platform [178]. In this dataset, acceleration sensors are placed to collect the horizontal and vertical vibration signals with the sampling frequencies 25.6kHz. It is recorded every 10 seconds, with each collection time of 0.1 seconds. They had three different operating conditions: 6 complete degenerate trajectories for training and 11 incomplete trajectories for testing. Detailed information about the dataset and pre-processing steps can be found in Appendix D.2.

This research constructs contrastive sample pairs through the following process:

1. The first two-thirds of each degradation trajectory are randomly cut, with the cut length being half of the entire trajectory, to create unlabeled degradation segments.
2. In each randomly cut segment, twelve consecutive historical samples are selected as a positive sequence, maximizing data utilization.
3. The order of this segment is then inverted to form a negative sequence, and both sequences are combined to create contrastive pairs.

For pre-training, 60,544 labeled samples were generated, with 30,277 samples used for training (each with dimensions  $6 \times 64$ ). For downstream training, 15,857 labeled samples were used, each with dimensions  $6 \times 64$ . Additionally, 25,152 samples were obtained for pretext training, and 7,360 labeled samples were used for downstream training. The labels represent degradation degree in percentage, normalized from the true RUL in seconds for each degradation trajectory.

#### 4.2.3.1 Proposed model

In this section, we validate the proposed SSL theory through a series of comparative experiments involving two models: a benchmark CNN-LSTM (See Section. 4.2.3.1) and an SSL-enhanced CNN-LSTM (See Section. 4.2.3.1). Both models share an identical structure and training set, with the primary distinction being that the SSL-based model leverages a broader range of unlabelled data, extracted from both the training and test sets, for pre-training.

### Benchmark CNN-LSTM.

As shown in Fig. 4.4, CNN-LSTM effectively harnesses the spatial feature extraction capabilities of CNNs and the sequence modeling prowess of LSTMs. The input data first undergoes a wavelet transform, enhancing its suitability for further processing by highlighting essential features while suppressing noise. The CNN modules consist of separable 2D convolutional layers, each followed by batch normalization and ReLU activation. This configuration reduces computational demands while ensuring efficient learning dynamics.

After the CNN layers, the LSTM units receive the spatially refined features, processing them through complex gate-regulated cycles to maintain and manipulate temporal dependencies. Each LSTM unit includes components such as sigmoid and tanh functions to manage the gates' behavior, crucial for learning sequences effectively. The culmination of this process is through a dense layer that aggregates the learned features into a final output, optimized against a mean squared error criterion.

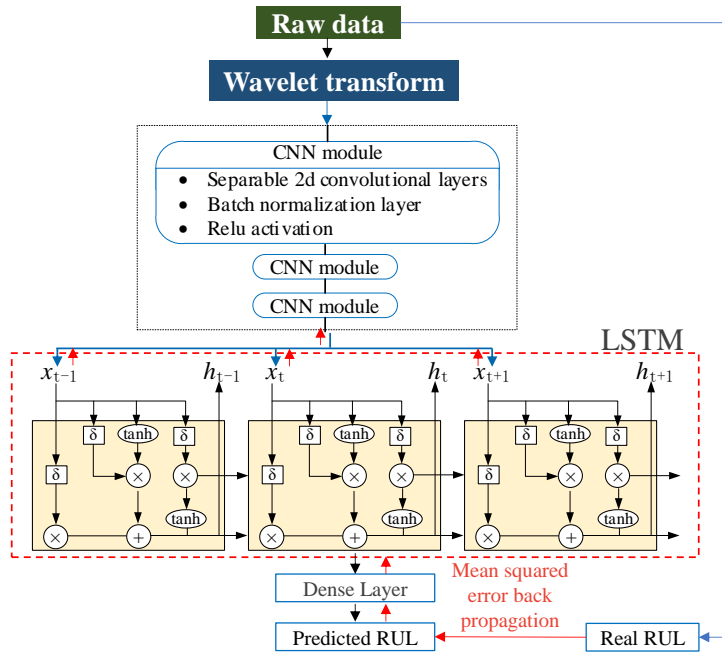
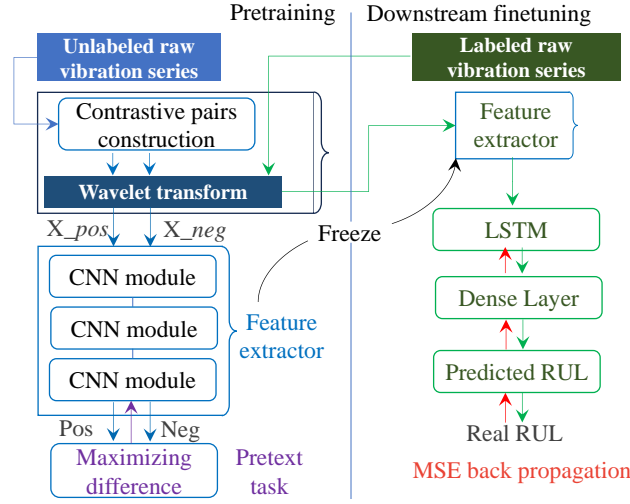


Figure 4.4: Benchmark CNN-LSTM network architecture.

### CNN-LSTM at SSL.

The CNN, depicted in Fig. 4.5, employs a Siamese structure, achieved by reusing a single CNN module across multiple input channels.

## 4.2. Novel contrastive SSL in sequential mining for RUL prediction



**Figure 4.5:** Siamese CNN-LSTM implementation diagram.

CNN-LSTM at SSL comprises two key phases: pretraining and downstream fine-tuning. During pretraining, similar to the benchmark model, unlabeled vibration data is first subjected to a wavelet transform to extract multifaceted features. However, unlike the benchmark, these features are organized into contrastive pairs before wavelet transformation, yielding input features  $X_{pos}$  and  $X_{neg}$ . These inputs are utilized in a pretext task designed to maximize the differentiation between positive and negative encoded features  $Pos$  and  $Neg$ , as generated by the feature extractor. The objective function in Eq. (4.2) seeks to optimize the pretext task by minimizing the error through an inverse relationship that maximizes the distance between  $Pos$  and  $Neg$ .

$$L_{ssl} = \underbrace{\sum_{i=x,y} \frac{1}{\sqrt{(pos_i - neg_i)^2}}}_{loss_1} + \underbrace{out_x - out_y}_{loss_2}. \quad (4.2)$$

Given that the monitoring data is collected from multiple channels, each corresponding to a different direction (e.g., horizontal  $x$  and vertical  $y$ ), it is crucial to prevent metric interference across channels. Consequently,  $loss_2$  is formulated to ensure consistency in processing results across different channels. Simultaneously,  $loss_1$  enhances the CNN's ability to capture sequential information across various directions, where  $pos_i$  and  $neg_i$  represent the feature encoding results of positive and negative sequential orders, respectively. The pretraining phase optimizes  $L_{ssl}$  through backpropagation across the entire CNN till the input layer.

In the downstream fine-tuning stage, the pretrained feature extractor, with its parameters frozen, processes labeled vibration data. The downstream model

## 4.2. Novel contrastive SSL in sequential mining for RUL prediction

mirrors the benchmark structure, with the distinction that the CNN modules in the CNN-LSTM are initialized with pretraining weights. Unlike the pretraining phase, CNN now accepts a single input,  $X_{pos}$ , rather than contrasting pairs. During this stage, MSE backpropagation is applied up to the CNN output layer.

### 4.2.3.2 Result discussion

The training of both benchmark and SSL model uses a batch size of 64, and the early stop mechanism is activated after 50 epochs of patience. The results of the proposed model are also compared to the SOTA methods presented in the research [131]. The qualitative prediction of the degradation trajectory is given in Appendix D.4, while the quantitative performance is analysed as follows.

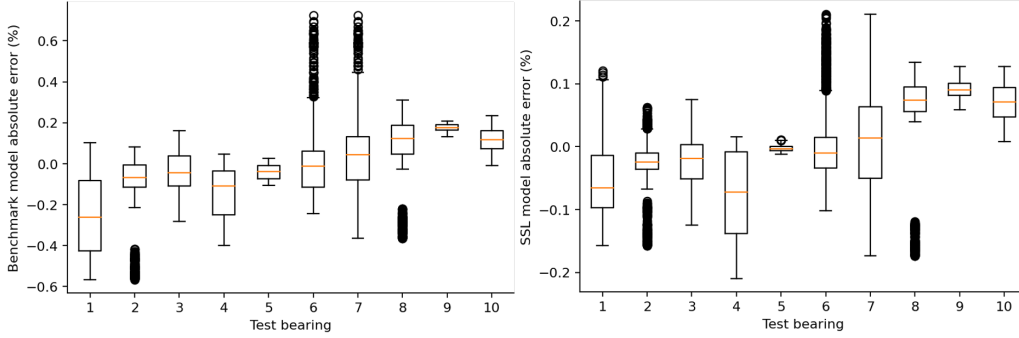


Figure 4.6: Prediction errors of different models.

CNN-LSTM at SSL demonstrates superior prediction performance compared to the SOTA model presented in [116], as illustrated in Table 4.2.

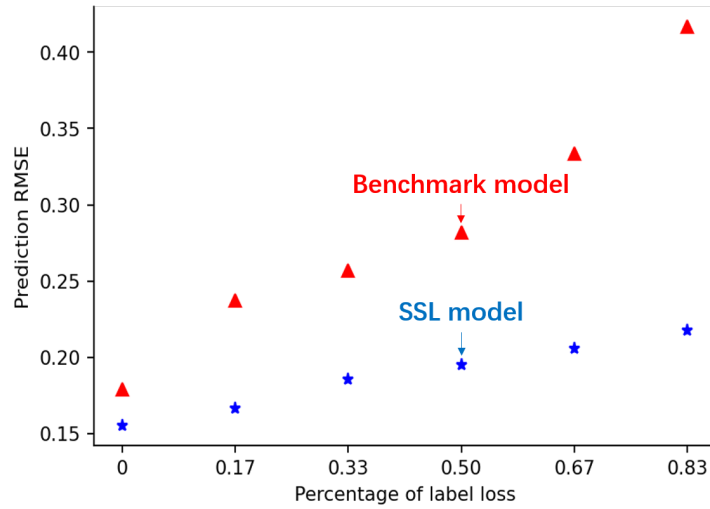
Table 4.1: Comparison of the prediction MAE of different models.

Methods	Testing bearing No.									
	1	2	3	4	5	6	7	8	9	10
SAE+GRU	0.40	0.29	0.35	0.35	0.35	0.33	0.35	0.27	0.34	0.29
<b>CNN+LSTM(Benchmark model)</b>	0.21	0.079	0.10	0.088	0.036	0.15	0.19	0.17	0.083	0.046
CNN+BiLstm	0.35	0.24	0.29	0.29	0.33	0.32	0.34	0.26	0.32	0.23
SSL+GRU(SOTA)	0.34	0.23	0.29	0.28	0.27	0.27	0.29	0.20	0.27	0.22
BiLstm+Attention	0.59	0.23	0.29	0.31	0.28	0.27	0.28	0.21	0.35	0.29
<b>CNN-LSTM at SSL</b>	<b>0.12</b>	<b>0.042</b>	<b>0.057</b>	<b>0.084</b>	<b>0.026</b>	<b>0.061</b>	<b>0.11</b>	<b>0.16</b>	<b>0.093</b>	<b>0.063</b>

Quantitative comparisons between the benchmark model and the proposed SSL model are presented in Fig. 4.6, with the evaluation conducted using the fully

labeled training dataset. The results indicate that the RMSE error bounds of the proposed SSL models are superior to those of the benchmark model.

Additionally, to test the model’s generality for few-shots scenarios, the training sets consist of 6 aging bearing are selected in increasing order from 1/6 to 1 ratio in the downstream supervised fine tuning, as shown on the horizontal axis of Fig. 4.7. The vertical coordinates of Fig. 4.7 indicate the RMSE for the entire test set. *The results show that the SSL framework is more effective for low labeling percentages. Additionally, it can adapt to degraded situations with unknown operating conditions feasibly.*



**Figure 4.7:** Performance of SSL and CNN-LSTM (Benchmark) models at different percentages of the labeled degradation trajectories in the training set.

We propose that the prediction performance of the proposed SSL-based CNN-LSTM model is less sensitive to the percentage of labeled data. When a model trained on an inadequately sized dataset is used to predict the RUL on a different, unknown dataset, there is a high likelihood of over-fitting. This over-fitting is often reflected in the similarity of the feature vectors in the output layer. Specifically, over-fit models tend to produce output feature vectors that are overly specific to the training dataset, failing to generalize effectively. In contrast, SSL algorithms, which leverage a broader unlabeled dataset for feature extraction, are less prone to over-fitting. This is because SSL models develop a more robust and generalized feature extractor that captures essential characteristics across diverse data, enhancing their ability to perform accurately even with limited labeled data. This theory will be further analysed in the next section when we conduct deep optimisation of CNN-LSTM at SSL.



## 4.3 Considering downstream information in SSL pretext task design

The results obtained in Section 4.2.3 underscore the strength of SSL in improving the bearing RUL prediction and few-shot prognostics by learning the PHM-related representations on the unlabelled data. But this method is not the full implementation of “Expanding mapping sources” mentioned in Section 1.3.1. We still have two concerns in the technology development:

- The first concern arises from the methodology outlined in Section 4.2. The proposed contrastive learning approach captures degradation trends using a single localized window of samples, which fails to provide a representation that encompasses the global degradation behaviour.
- The second concern pertains to the dataset. Traditional methods normalize RUL labels by setting an artificial endpoint, such as the operational cut-off time, which may not align with the actual failure point. This normalization introduces bias, reduces the physical interpretability of the RUL, and complicates its practical application in maintenance planning. Although normalized RUL values simplify upstream and downstream ML tasks by providing a consistent range (e.g.,  $[0, 1]$ ), they obscure the true degradation state, as the same normalized value can represent different failure stages across various systems. Conversely, using true RUL values, while more representative of actual conditions, increases the complexity of feature learning due to their uneven distribution in a large scope.

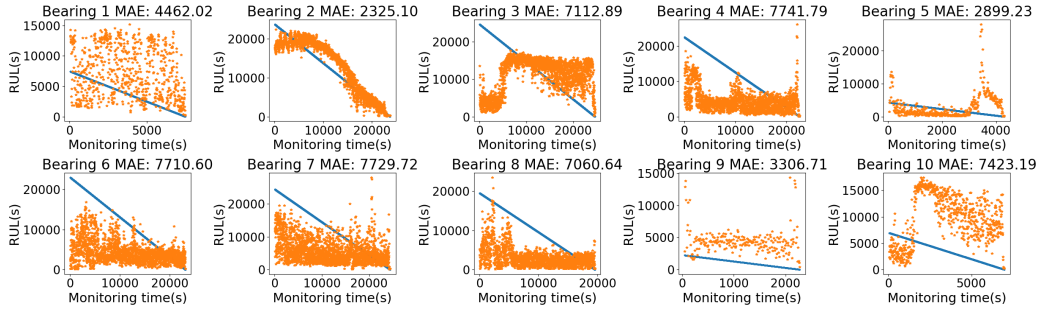
To address these issues, We first analyze the existing problem in Section 4.3.1. Subsequently, in Section 4.3.2, we introduce an improved SSL model that incorporates downstream information into the pre-training process to solve the aforementioned problem. Finally, in Section 4.3.3, we validate the effectiveness of the improved SSL model.

### 4.3.1 Analysis of the proposed SSL drawbacks

When employing our pre-trained model to predict the true RUL by adjusting the downstream supervised labels to the actual values, the results depicted in Fig. 4.8 reveal a significant overall prediction error.

However, the SSL model demonstrates strong predictive accuracy in specific instances, such as with Bearing 2. This suggests that as the complexity of downstream tasks increases, the upstream pre-trained features struggle to provide precise predictions. Therefore, it becomes imperative to reevaluate and redesign the

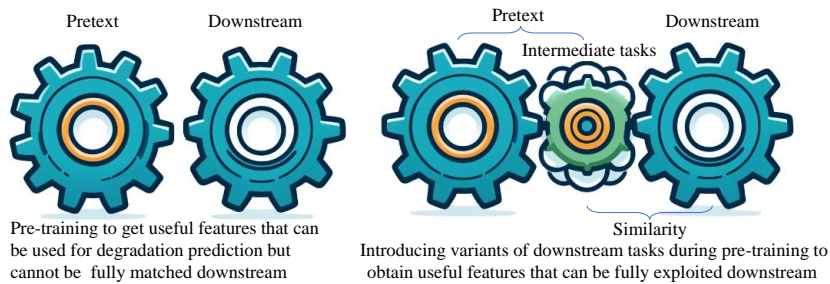
### 4.3. Considering downstream information in SSL pretext task design



**Figure 4.8:** Negative impacts of converting normalized labels to true labels on downstream task predictions.

pretext tasks to improve the model performance in these more challenging scenarios. An effective feature extractor should not be confined to predicting simplified, and normalized data distributions; it must also be adept at handling more realistic predictive tasks, as illustrated in Fig. 4.9. The sub-optimal performance observed during the transition from pre-training to downstream tasks can be attributed to a **misalignment between the learned features and the specific requirements of the downstream task**. While the pretext tasks may produce useful features, these features may not be fully exploitable by the downstream model.

**Downstream task variants.** We innovatively propose to solve the misalignment problem by adding intermediate tasks similar to the downstream task in the pre-training to add variants of the downstream task in the upstream pretext task, as shown in Fig. 4.9.



**Figure 4.9:** Self-supervised learning considering downstream information.

The downstream prediction task can be transferred as a learnable intermediate task by selecting two prediction targets of the same object at different periods and designing the relative relationship or absolute constant features between them as a “ratio-matched intermediate gear” between upstream and downstream.

The “ratio-matched intermediate gear” strategy integrates downstream prediction task considerations into upstream pre-training through two key components: (1) incorporating the downstream task-related structure into the pre-training model, and (2) embedding the downstream task-related loss function into the pre-training process, as detailed in Section 4.3.2.

### 4.3.2 Deep enhancement of the contrastive SSL

As detailed in Appendix D.5, we enhance the proposed SSL-based CNN-LSTM model in three key aspects, guided by the aforementioned “Downstream task variants” theory. The new flowchart is illustrated in Fig. 4.10.

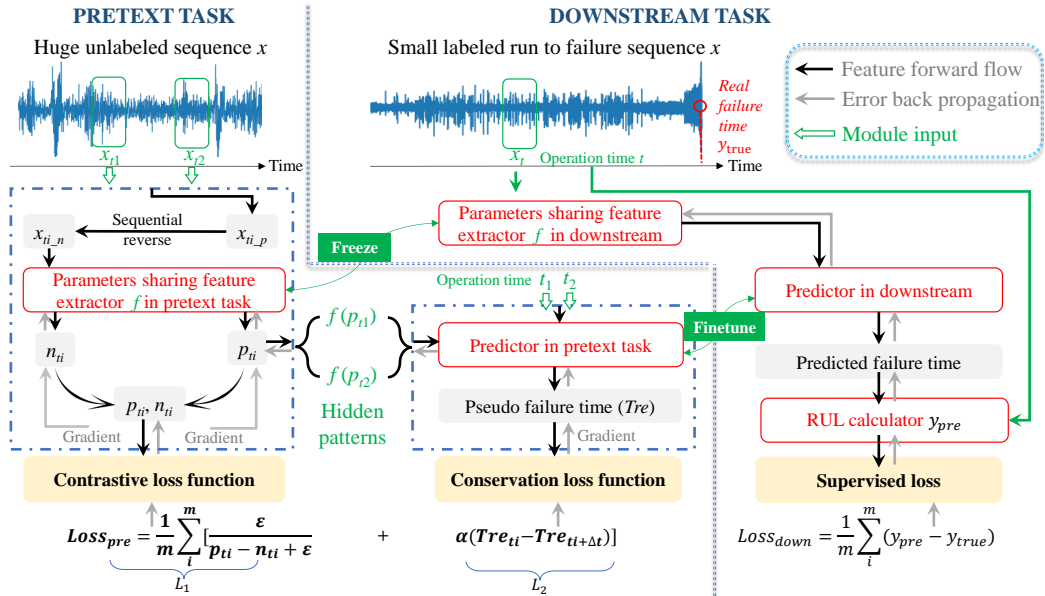


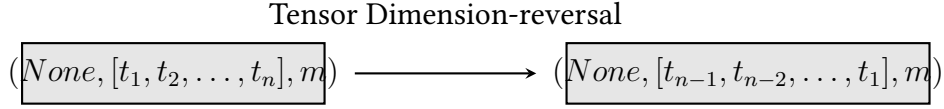
Figure 4.10: Contrastive SSL architecture considering downstream information.

The entire model have three key modules: “Feature extractor,” “Predictor,” and “RUL Calculator.” “Feature extractor” and “Predictor” appear both in “Pretext task” and “Downstream task.” Our enhancements to the existing SSL strategy focus on these three modules and the “Conservation loss” in Fig. 4.10.

**1) Novel End-to-End contrast pair construction approach.** An NN layer-based tensor operation is employed for temporal axis reversal to build contrastive input pairs. As shown in Section 4.2.2, the construction of contrastive pairs often relies on extensive data augmentation or the generation of new data samples, which can be computationally costly and risk overfitting. In contrast, as shown in Fig. D.5, temporal axis reversal is a lightweight technique applied directly to

### 4.3. Considering downstream information in SSL pretext task design

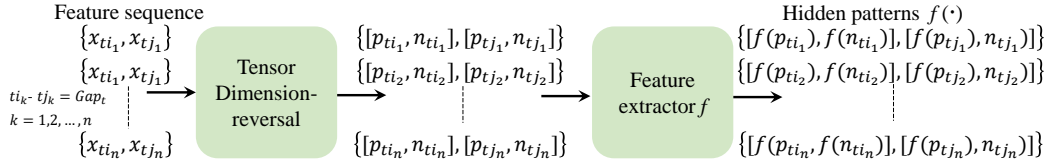
the input tensor, efficiently utilizing the existing dataset without the need for additional sample generation.



**Figure 4.11:** Inversion of the input temporal feature dimension to generate the contrastive tensor sample.

In addition, we propose a new modeling strategy that eliminates the need for feature extraction and data pre-processing in building the input space by utilizing raw time series as inputs in an end-to-end model, thereby simplifying the application process.

**2) Feature extractor and pseudo failure time predictor.** The feature extractor  $f$  extracts hidden patterns  $f(\cdot)$ , which are subsequently used as input for the failure time predictor.



**Figure 4.12:** Basic structure of feature extractor.

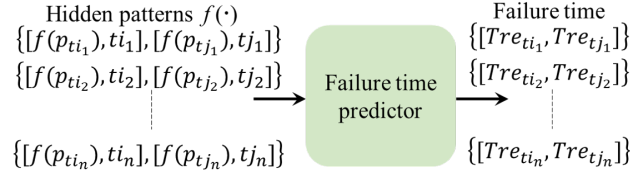
As shown in Fig. D.6, we still use the nested structural reused Siamese structure in employing the feature extractor  $f$ . For example, sample  $x_{t_i}$  and  $x_{t_j}$  represent sequences truncated by two sliding windows. From each window, we obtain temporal frequency features input pairs  $p_{t_i}, n_{t_i}$  and  $p_{t_j}, n_{t_j}$ , corresponding to  $x_{t_i}$  and  $x_{t_j}$ . In the pretext task, the inputs  $p_{t_i}$  and  $n_{t_i}$  of  $f$  are the contrastive input pairs, outputting the hidden patterns  $f(p_{t_i})$  and  $f(n_{t_i})$ . For samples from  $t_j$ , the operation is the same. learning objective that will be defined later.

Failure time “Predictor” produces the failure time  $Tre$  used as input for the RUL calculator. In pretext task, it generates pseudo-failure thresholds  $Tre_{t_i}, Tre_{t_j}$  based on the input pairs  $f(p_{t_i}), t_i$  and  $f(p_{t_j}), t_j$ . These thresholds, denoted as  $Tre_{t_i}, Tre_{t_j}$ , are optimized concerning the “Conservation loss.”

Downstream, the “Feature Extractor” loads pretrained, frozen weights, while the “Predictor” loads pretrained weights and undergoes fine-tuning.

**3) RUL calculator.** To capture the precise failure time ( $Tre$ ) and avoid introducing bias in using a theoretical or practical cut-off time for RUL labeling, we

### 4.3. Considering downstream information in SSL pretext task design



**Figure 4.13:** Basic structure of failure time predictor.

integrate operation time ( $RT$ ) into the RUL prediction model. Instead of forecasting a linear sequence of RUL points, our model predicts a series of fault times ( $Tre$ ), constant for a specific bearing trajectory. RUL is then calculated by subtracting  $RT$  from the predicted  $Tre$  values, as shown in Eq. (4.3).

$$RUL = Tre - RT \quad (4.3)$$

The normalized RUL value is calculated for training purposes as shown in Fig. 4.14.



**Figure 4.14:** The structure of RUL calculator in the proposed neural network.

“RUL calculator” is only used in the downstream’s supervised process.

**4) Contrastive and conservation combined loss function.** In the pretext task phase, we introduce a combined loss function, denoted as the “Contrastive and Conservation Combined Loss,” by adding  $L_2$  to the original contrastive loss  $L_1$ . The feature extractor is guided by  $L_1$ , which is designed to maximize the discrepancy between  $n_{ti}$  and  $p_{ti}$ , thereby capturing the expressive power of the degradation trend within a sliding window. Simultaneously,  $L_2$  incorporates the learned representations of  $n_{ti}$  and  $p_{ti}$ , which provide global insights by being supervised through  $L_1$ . This approach ensures that the learned features are more suitable for usage in the downstream’s “Predictor” and “RUL calculator.”

In the context of bearing degradation, the “ratio-matched intermediate gear” serves as a constant failure threshold across different observation points on the same aging trajectory. We assume identical failure times ( $T_{re}$ ) for samples along the same trajectory, integrating downstream prediction information into feature extraction during pre-training. This forms a consistent pipeline from input features to failure time prediction. Although RUL labels are not available during pre-training, the assumption of equivalent predicted failure times aligns with

### 4.3. Considering downstream information in SSL pretext task design

the physical understanding expressed in Eq. (4.4), which relates RUL ( $RUL_{t_i}$ ) and operation time ( $RT_{t_i}$ ) at time  $t_i$ :

$$RUL_{t_1} + RT_{t_1} = RUL_{t_2} + RT_{t_2} = \dots = RUL_{t_n} + RT_{t_n} = T_{re} \quad (4.4)$$

Thus, calculating the RUL can be seen as determining a series of  $T_{re}$  at different points along the same trajectory. When the predictor estimates the positive encoded features from two time points  $t_1$  and  $t_2$ , their  $T_{re_{t_1}}$  and  $T_{re_{t_2}}$  should be as close as possible, constrained by  $L_2$ , as depicted in Fig. 4.10.

In the downstream phase, the entire model is trained using the MAE loss, defined as:

$$\mathcal{L}_{\text{down}} = \frac{1}{N} \sum_{i=1}^N (y_{\text{RUL}} - \hat{y}_{\text{RUL}}), \quad (4.5)$$

Here,  $y_{\text{RUL}}$  represents the predicted RUL from the RUL calculator layer,  $\hat{y}_{\text{RUL}}$  denotes the labeled normalized RUL in the dataset, and  $N$  stands for the number of samples under test.

#### 4.3.3 Validation of the improved contrastive SSL model on bearing prognostics

We modify the CNN-LSTM model, described in Section 4.2.3, by replacing the CNN stacking in the nested Siamese module with residual module stacking, enhancing the model's expressiveness.

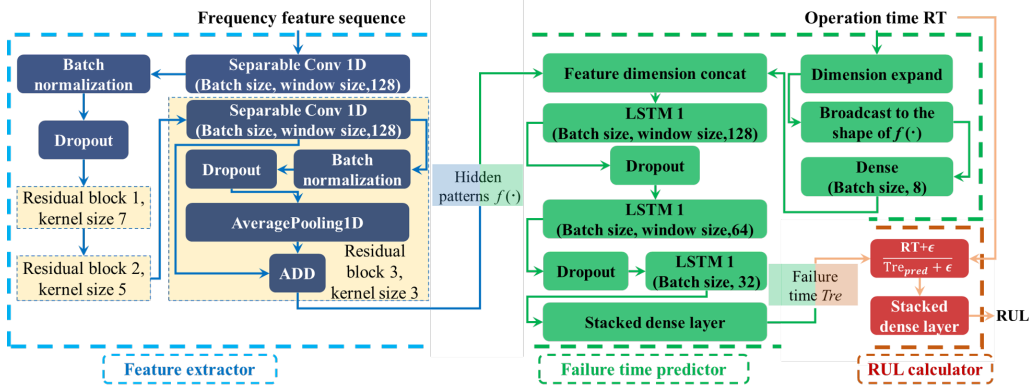


Figure 4.15: CNN-LSTM network architecture.

Fig. 4.15 illustrates the SSL model architecture tailored for downstream tasks. The architecture begins with Separable Convolution 1D layers, known for their efficiency in both parameter usage and computation. These layers are followed

by batch normalization and dropout, which serve to normalize activations and prevent overfitting, respectively. “AveragePooling1D” layer is then applied to reduce data dimensionality by summarizing features within specified windows. At the core of the model are stacked ResNet-like blocks featuring residual connections, which facilitate gradient flow through network layers, thereby mitigating vanishing and exploding gradient issues. These blocks employ varying kernel sizes to capture multi-scale patterns, with residual connections enabling identity mappings as proposed by He et al. [179], thereby stabilizing the learning process in deeper networks.

For failure time prediction, the model leverages LSTM networks to process sequences. In the input preparation phase, feature fusion is performed on the hidden patterns  $f(\cdot)$  and operation time  $t$ . This involves expanding and broadcasting  $t$  to match the feature dimensions defined by  $f(\cdot)$ . A dense layer with 8 units then transforms this broadcasted data into a concise feature vector. The processed features are concatenated with  $f(\cdot)$ , forming a comprehensive input for the subsequent stacked LSTM layers. These LSTM layers, organized hierarchically with decreasing units (128, 64, and 32), progressively refine the extracted features, with each layer followed by dropout for regularization.

The results of the Benchmark model and the improved constrative SSL model are discussed in Subsections 4.3.3.1 and 4.3.3.2.

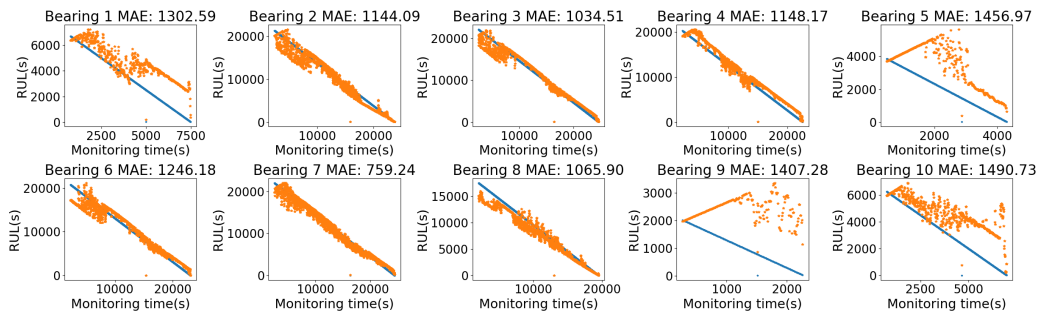
#### 4.3.3.1 Benchmark results

A CNN-LSTM model, mirroring the structure depicted in Fig. 4.15 of the proposed SSL model, serves as the benchmark in our research for investigating the performance of the SSL mechanism. The SSL model has the same structure as the benchmark model, using the hyperparameters provided in the Appendix D.3 and the dimensional structure in Fig. 4.15. The difference is that the frequency feature sequence of the benchmark model is not inverted by Section 4.2.2.1, and the benchmark model does not incorporate SSL pretext task but aligns with the proposed SSL framework in terms of data preprocessing, labeling, and supervised training phases. Consequently, the primary distinction between the two models hinges on the influence of the SSL learning process.

The results obtained with the considered benchmark model (CNN-LSTM without SSL) are provided in Fig. 4.16, where orange dots represent RUL predictions over time, while the blue line indicates the actual RUL until bearing failure. The X-axis and Y-axis represent monitoring time and RUL in seconds, respectively. This figure shows the RUL prediction using the Benchmark model for the testing bearings over time, with Mean Absolute Error (MAE) serving as the accuracy metric. Its overall MAE is 1203.74 (s) and the normalized error is 8.43%.

Several key observations can be made from the results. Firstly, the benchmark

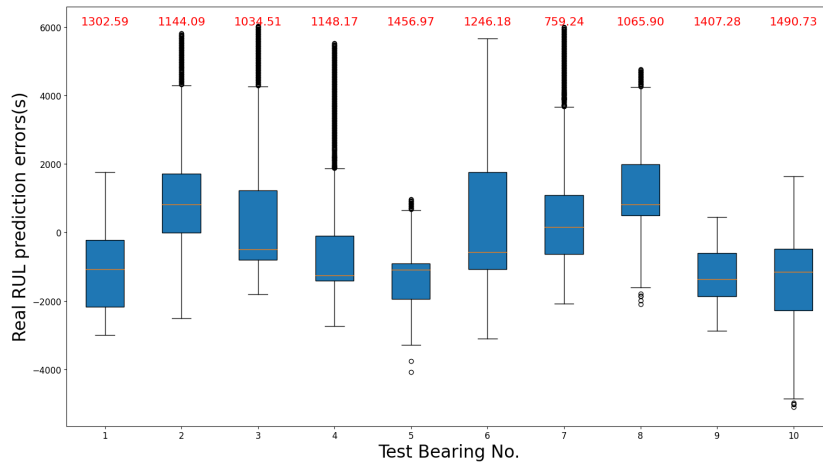
### 4.3. Considering downstream information in SSL pretext task design



**Figure 4.16:** Prediction results of Benchmark model.

model demonstrates an ability to effectively capture degradation trends. However, it is important to note that predicting the true RUL exhibits a notable dispersion of results, largely attributed to the substantial variability in the model’s output. Moreover, the analysis reveals instances where the model tends to significantly overestimate the RUL, suggesting the presence of potentially risky situations in the predictions.

Furthermore, we also present a box plot of predicted errors to illustrate the distribution of predictions for each test bearing, as depicted in Fig. 4.17. The



**Figure 4.17:** Box plots of quantitative statistics of Benchmark model prediction errors.

box plot in Fig. 4.17 visualizes the RUL prediction errors for ten different bearings using the benchmark model, revealing considerable variation in prediction accuracy. The MAE range from 759.24 seconds for Bearing 7 to 1490.73 seconds for Bearing 10, indicating an average discrepancy in predicted versus actual RUL. The interquartile ranges (IQRs) suggest varying levels of consistency across bear-



ings, with Bearing 4 exhibiting the most significant spread of errors and Bearing 8 the least. Notably, several bearings display outliers, representing substantial deviations from typical error values. Negative values in the error data suggest overestimations of RUL, whereas positive values indicate underestimations. This variability across bearings underscores the benchmark model’s inconsistent predictive performance.

#### 4.3.3.2 Validation results of the proposed method

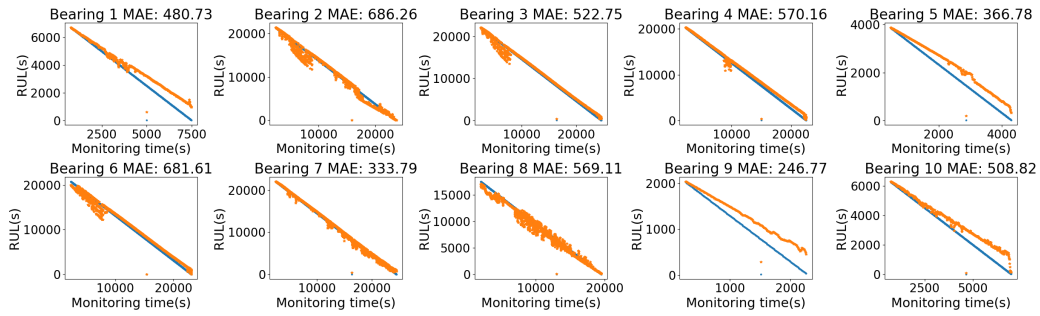
We present the qualitative and quantitative results here, keep the detailed discussions in the Appendix D.6, where the validation results of the proposed method for RUL prediction are discussed, with an emphasis on qualitative results (Appendix D.6.1), prediction uncertainty (Appendix D.6.2), computational cost comparison (Appendix D.6.3), the impact of labelled data availability (Appendix D.6.4), model architecture (Appendix D.6.5), and generalization testing on the tool wear dataset (Appendix D.6.6). Monte Carlo Dropout was used to quantify uncertainty, revealing higher uncertainty during mid-operation stages. The computational cost analysis highlighted that while the proposed model incurs higher Floating Point Operations (FLOPs) due to SSL, it achieves better prediction accuracy compared to benchmark models. The study also demonstrated that the SSL model outperforms traditional methods even with reduced labeled data, indicating its robustness in scenarios with limited data availability. Furthermore, an exploration of the model architecture revealed that selective freezing of expressive pre-training sections enhances performance during fine-tuning. Finally, the generalization test using a milling process dataset validated the robustness and applicability of the SSL method in noisy industrial environments, underscoring its potential for improving predictive accuracy in real-world settings.

**Qualitative results.** The RUL predictions obtained with the proposed SSL model are presented in Fig. 4.18.

Comparing the RUL prediction results obtained by the proposed improved SSL in Fig. 4.18 with the one of benchmark model presented in 4.16, we find that:

- The overall SSL model’s MAE is 496.14 (s) and its normalized error improves from 8.43% to 3.48%.
- The MAE values of the SSL model, shown in Fig. 4.18, are significantly lower than those in Fig. 4.16 for each bearing.
- The model performance on bearings 1, 5, 9, and 10 are greatly improved.

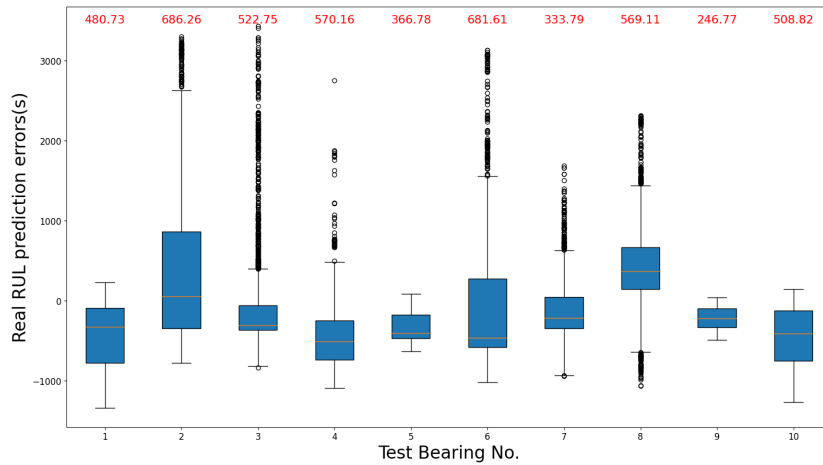
### 4.3. Considering downstream information in SSL pretext task design



**Figure 4.18:** Prediction results of SSL model.

Blue curve is the manual RUL labels, yellow points are predicted values.

- With a large range of variation in the true RUL values, the RUL predictions of the SSL model, shown in Fig. 4.18, have less dispersion.
- Significant errors in early-stage RUL predictions for bearings are primarily due to the lack of sufficient degradation trend data at the beginning of operation. During initial operation phases, degradation patterns are not yet clearly established.



**Figure 4.19:** Box plots of quantitative statistics of SSL model prediction errors.

Upon comparing the box plots in Fig.4.17 and Fig.4.19, one can see that the SSL model surpasses the benchmark model in RUL prediction accuracy. Quantitatively, the SSL model consistently exhibits a lower MAE. For example, the MAE for Bearing 1 is reduced from 1302.59 seconds in the benchmark model to 480.73 seconds in the SSL model. Furthermore, the interquartile range (IQR) is considerably narrower in the SSL model, indicating more consistent predictions. This

### 4.3. Considering downstream information in SSL pretext task design

is exemplified in Bearing 6, where the IQR shows a significant reduction. The SSL model also demonstrates fewer and less pronounced outliers, which suggests an improvement in prediction reliability. Overall, the SSL model clearly shows a quantitative advantage in terms of both accuracy and consistency in RUL prediction. Specifically, we observed that prediction deviations increase as RUL approaches the end, likely because at the experiment’s termination, some bearings have deteriorated significantly while others only slightly. This suggests a true severe failure threshold beyond the experimental termination condition, as detailed in Appendix D.6.5.

**Quantitative results.** We compared our results with multiple SOTA methods presented in [116]. Note that in the existing studies, only normalized RUL predictions are considered. To compare with these results, we also assess the normalized RUL predictions by our improved SSL model, as shown in Table 4.2.

**Table 4.2: Comparison of different models on the normalized RUL predictions.**

Methods	Testing bearing No.									
	1	2	3	4	5	6	7	8	9	10
SAE+GRU	0.40	0.29	0.35	0.35	0.35	0.33	0.35	0.27	0.34	0.29
<b>CNN+LSTM(Benchmark model)</b>	0.19	0.054	0.047	0.057	0.38	0.060	0.035	0.061	0.70	0.24
CNN+BiLstm	0.35	0.24	0.29	0.29	0.33	0.32	0.34	0.26	0.32	0.23
SSL+GRU(SOTA)	0.34	0.23	0.29	0.28	0.27	0.27	0.29	0.20	0.27	0.22
BiLstm+Attention	0.59	0.23	0.29	0.31	0.28	0.27	0.28	0.21	0.35	0.29
<b>CNN+LSTM with contrastive SSL (No Downstream Info)</b>	<b>0.12</b>	<b>0.042</b>	<b>0.057</b>	<b>0.084</b>	<b>0.026</b>	<b>0.061</b>	<b>0.11</b>	<b>0.16</b>	<b>0.093</b>	<b>0.063</b>
<b>Proposed model</b>	<b>0.072</b>	<b>0.032</b>	<b>0.024</b>	<b>0.028</b>	<b>0.095</b>	<b>0.033</b>	<b>0.015</b>	<b>0.032</b>	<b>0.12</b>	<b>0.081</b>

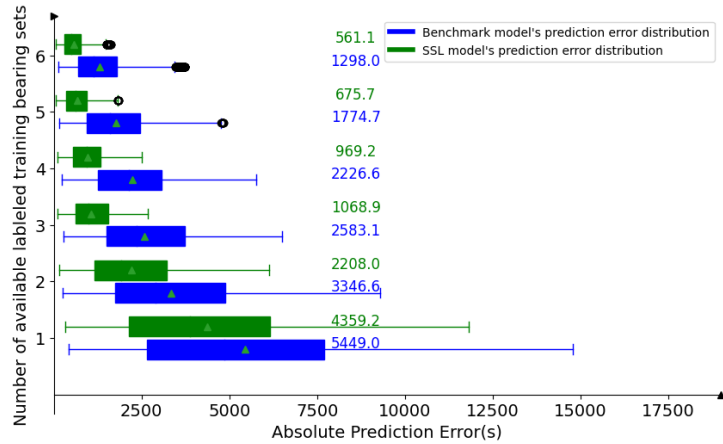
Table 4.2 provides a comparative analysis of the different SOTA methods applied to RUL prediction across the ten test bearings of the dataset. From the table, we can observe the following key points: The proposed model consistently demonstrates superior performance across all test cases, achieving significantly lower MAE values compared to other models. While the benchmark CNN-LSTM model shows relatively good performance, it still falls short of the proposed model’s results.

Comparing the proposed model with the CNN-LSTM model using contrastive SSL without downstream task information reveals a noticeable performance improvement in the proposed model. The results indicate that including downstream task information in the contrastive SSL framework leads to a substantial reduction in MAE. For example, in test case 1, the MAE decreases from 0.1214 (without downstream information) to 0.0716 (with downstream information), showing significant performance gains.

Our proposed model exhibits the lowest error rates across all bearings. For instance, it achieves an error rate of 0.0716 for Bearing 1, which is substantially lower than the benchmark model output (0.1941) and even more pronounced

### 4.3. Considering downstream information in SSL pretext task design

when compared to the SAE+GRU’s result (0.4036). The proposed model’s highest error rate is 0.1221 for Bearing 9, still significantly lower than the benchmark’s highest of 0.6966 for the same bearing.



**Figure 4.20:** Box plots of the impact of different amounts of labelled data on the prediction results of the model.

Additionally, Fig. 4.20 presents the boxplots of prediction errors for specific models across different numbers of labeled datasets. The triangular arrows in the boxplot represent the mean value of the absolute error of prediction, which is shown on the right side of the figure. The whiskers in the box plot extend to the highest and lowest values within the range that are not considered outliers. Outliers are displayed as individual points positioned outside the whiskers. In Fig. 4.20, we observe that the SSL model consistently outperforms the benchmark model across various levels of data availability. Specifically, the SSL model maintains a stable error distribution, even when the number of labeled datasets is reduced from six to three. In contrast, the benchmark model experiences a significant increase in prediction error under the same reduction in labeled data, highlighting the SSL model’s robustness.

Quantitatively, with six labeled datasets, the SSL model achieves a median prediction error of merely 561.1 seconds, while the benchmark model’s median error is significantly higher, at 12980 seconds. This substantial difference in performance is also reflected in the relative compactness of the SSL model’s error distribution boxes within the box plots, indicating not only lower median errors but also a tighter error range across varying labeled dataset sizes.

However, it should be noted that the prediction capabilities of both models experience a marked decline when available labeled datasets fall below three. This threshold signifies the critical point beyond which the models, reliant on data-driven mechanisms, are unable to glean sufficient information to make ac-

curate RUL predictions. This underscores the necessity for a minimum dataset size to ensure reliable model performance and suggests an area for further research into how SSL can be made more robust against significant data scarcity.

## 4.4 Summary

The contributions presented in this chapter advanced the field of SSL and its application in PHM by addressing key challenges related to sequential information mining, downstream task alignment, and dynamic knowledge representation. The proposed methods laid the foundation for developing more robust and adaptable prognostic models in the presence of limited labeled data and diverse operating conditions. The key contributions of this chapter are summarized as follows:

1. **Development of a contrastive learning strategy for sequential information mining.** A novel SSL approach was proposed that capitalized on distinguishing between sequential orders to capture essential degradation trend information. This model not only refined the understanding of degradation trends but also enhanced RUL prediction accuracy. By maximizing the differences in feature outputs for data processed in various sequential orders, the model achieved enhanced performance. The stability and effectiveness of this SSL strategy in RUL prediction with limited labels were confirmed (See details in Appendix D.6), highlighting the benefits of freezing the pre-trained feature extractor for consistent feature processing.
2. **Integration of downstream task information into the SSL framework.** A consistency condition was introduced based on a constant failure threshold prediction for failure time predictions as an intermediary step. This approach ensured that the learned features aligned with the requirements of the downstream task, improving the relevance of feature representations for degradation processes and boosting overall SSL model performance. The impact of this consistency condition on both upstream and downstream tasks was explored (See details in Appendix D.6), demonstrating a noteworthy phenomenon: the upstream pretext task learned what to predict, while the downstream task further aligned the already reasonable distribution of results with real-world scenarios.
3. **Case study of SSL theory applied to bearing prognostics.** A case study employed a contrastive SSL approach using two CNN-LSTM models on the PRONOSTIA Bearing Dataset. Initially, a **Siamese CNN-LSTM** architecture was proposed, featuring step-by-step SSL pre-training and

fine-tuning phases. A novel objective function (Eq. (4.2)) optimized SSL performance, and comparative analysis confirmed that this approach outperformed SOTA models by achieving lower mean absolute error (MAE) values on all tested bearings (Table 4.2). This validation underscored the robustness and generalization capabilities of the **enhanced SSL-based residual CNN-LSTM model**, emphasizing its potential in scenarios with sparse labeled data.

4. **Comprehensive evaluation of the proposed models.** Comprehensive evaluations of the proposed models were carried out, focusing on prediction error metrics, uncertainty quantification, computational cost, and the impact of labeled data availability. Further testing on a milling process dataset confirmed the model’s superior performance in RUL prediction. While the enhanced performance through SSL came with a higher computational cost due to an increase in FLOPs, this trade-off was justified. The additional computational effort was crucial for effectively leveraging information from unlabeled data and ensuring precise and reliable RUL predictions, which are vital for PHM applications.

The next chapter builds on these foundational developments, transitioning from specialized feature learning to the practical application of these strategies in developing a generalized PHM model. Chapter 5 focuses on enhancing computational efficiency, managing variable-length sequences, and overcoming integration challenges across diverse industrial systems. With the groundwork laid for “Directional Mining” of meaningful degraded representations, it delves into leveraging and mining higher-level information—specifically, physics knowledge, from unlabelled data through advanced PIML techniques to further refine and optimize PHM models.

# Improving existing PIML and SSL paradigms to build a common PHM model across scenarios

---

## Contents

---

<b>5.1</b>	<b>Introduction</b> . . . . .	<b>110</b>
<b>5.2</b>	<b>Extend PIML to active mining of knowledge</b> . . . . .	<b>111</b>
5.2.1	Gated neuron theory for liquid NN . . . . .	112
5.2.2	Discovering physics on unlabeled data . . . . .	114
5.2.3	Validation of the proposed methodology for dynamic identification in robotic arms . . . . .	115
5.2.3.1	Test E2NN performance on real data . . . . .	116
5.2.3.2	Investigation on the E2NN's robustness . . . . .	117
<b>5.3</b>	<b>E2E data-driven model for cross-scenario time series processing</b> . . . . .	<b>118</b>
5.3.1	Gated selective state spaces mechanism . . . . .	119
5.3.2	End-to-End (E2E) CNN-SSM model . . . . .	121
5.3.3	Performance evaluation of E2E CNN-SSM model on different datasets . . . . .	123
5.3.3.1	Bearing prediction results . . . . .	123
5.3.3.2	Battery prediction results . . . . .	125
<b>5.4</b>	<b>A generic model incorporating the advanced PIML and SSL</b> . . . . .	<b>126</b>
5.4.1	Liquid physics-informed CNN-SSM model . . . . .	128
5.4.2	Hybrid learning strategy . . . . .	129
5.4.2.1	Supervised information generation . . . . .	129
5.4.2.2	Simultaneous supervised and self-supervised learning . . . . .	130
5.4.3	Investigating the performance of liquid PI CNN-SSM model . . . . .	131
5.4.4	Interpretable outcomes of knowledge representation after Hybrid learning . . . . .	139
<b>5.5</b>	<b>Conclusion</b> . . . . .	<b>143</b>

---

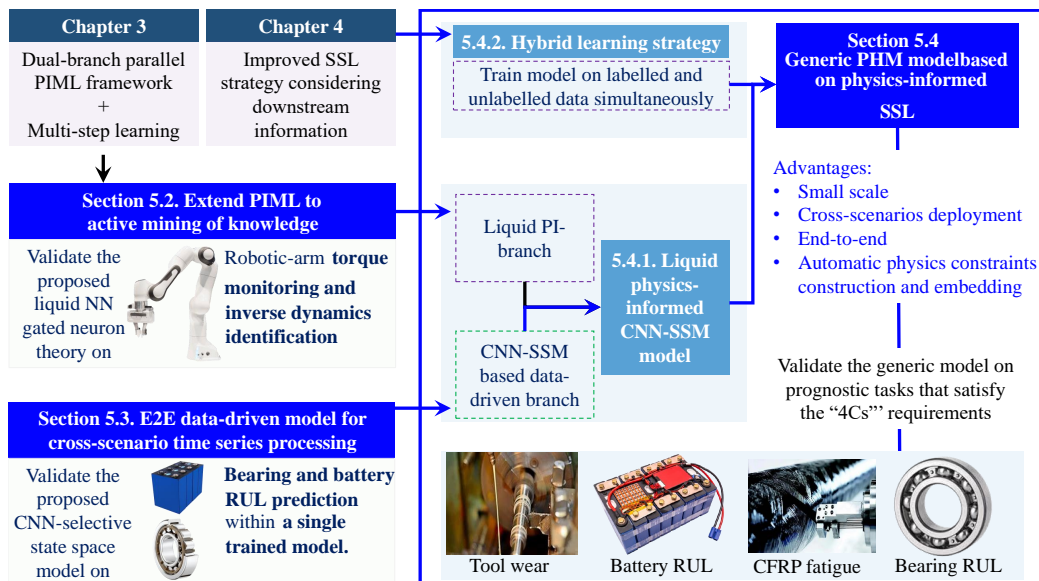
Part of the work in this chapter is based on the following publication:

J3 **WeiKun DENG**, Fabio ARDIANI, Khanh T.P.NGUYEN, Mourad BENOUS-SAAD, Kamal MEDJAHHER (2024). Physics informed machine learning model for inverse dynamics in robotic manipulators. *Applied Soft Computing*, 111877.

C2 **WeiKun DENG**, Khanh T.P.NGUYEN, Kamal MEDJAHHER (2022), Physics Informed Self Supervised Learning For Fault Diagnosticss and Prognostics in the Context of Sparse and Noisy Data. *PHM Society European Conference* 7 (1), 574-576.

## 5.1 Introduction

This chapter builds on insights from previous chapters to enhance PIML and SSL methodologies. Our objective is to develop a generic PHM model that is lightweight, computationally efficient, and hardware-friendly. Additionally, the model is designed with self-adaptive capabilities, making it versatile for a wide range of deployment scenarios.



**Figure 5.1:** Research framework for building the generic PHM model across scenarios based on the combination of improved PIML and SSL paradigms.

However, existing PIML methods still face significant challenges, particularly in model design and training alignment. Despite the advantages of the generic PIML architecture, such as reducing dependence on experts, the development of custom physical operators often requires trial and error. Moreover, integrating



prior physics knowledge with data-driven models remains a complex process, making PIML a passive learning method when it comes to model selection. Similarly, SSL methods encounter difficulties in designing pretext tasks and choosing appropriate models. The diversity of objectives complicates the creation of pretext tasks, necessitating iterative refinement for both pretext and downstream tasks. This experimental and customized approach limits the expressive and adaptive capabilities of the models, often leading to empirical and case-specific customization. Furthermore, the application of physics knowledge varies across different devices, measurable quantities, and timescales, affecting how model inputs and outputs are processed. The degraded representations of unlabeled data in various scenarios also show significant variation. Therefore, in the following 4 sections, we present three key contributions that shape our generic PHM model.

- We extend PIML by introducing autonomous operator discovery and symbolic relation exploration, which we term “active knowledge discovery” in Section 5.2. This advancement shifts PIML from a passive to an active learning approach, enabling the model to uncover and utilize underlying physical principles more effectively.
- We develop a lightweight, end-to-end backbone model that can efficiently process inputs across a wide range of measurable quantities, varying context lengths, and different timescales in Section 5.3. This model is designed to be flexible, allowing it to adapt seamlessly to different prediction targets without sacrificing computational efficiency or hardware compatibility.
- We enhance the multi-step training process by implementing a refined SSL strategy called “Hybrid Learning” in Section 5.4. This approach integrates downstream information and physics discovery tasks, improving representation learning for both labeled and unlabeled data. This hybrid strategy ensures more robust and accurate model performance across diverse scenarios.

These contributions collectively address the limitations of existing PIML and SSL methods, paving the way for a more versatile, efficient, and adaptive PHM model that is validated across different real-world applications.

## 5.2 Extend PIML to active mining of knowledge

This section introduces a new concept by transforming the tradition knowledge-embedded representation to active discovery, as illustrated in Fig. 5.2.

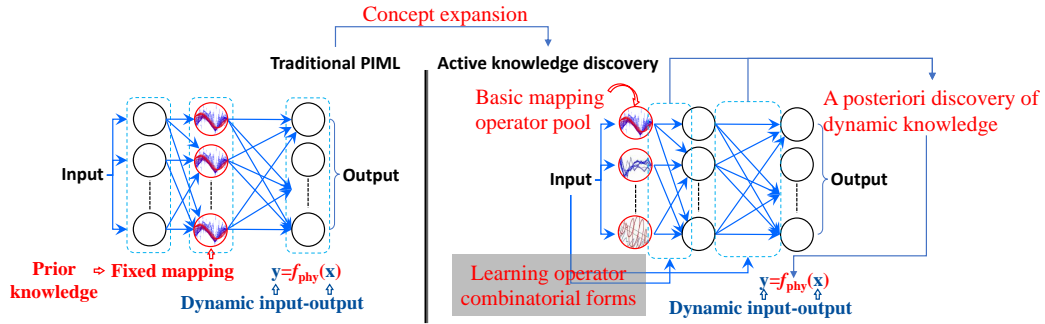


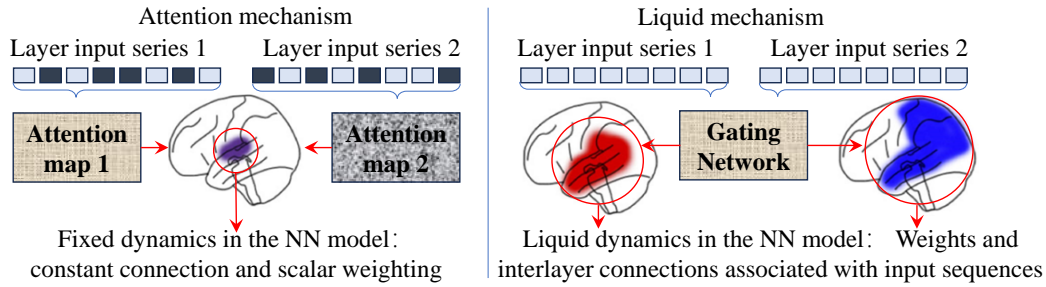
Figure 5.2: From knowledge embedded representation to active discovery.

tradition PIML relies on fixed operator arithmetic and rigid enforcement of physical constraints on ML data flows by static learned weights and biases which restricts flexibility. The nature of active knowledge discovery change this by applying **morphological discovery**, similar to how water adapts to the shape of its container. The primary concept of morphological discovery can be summarized as “Employ nonlinear operators as activation functions to analytically transform hidden layer inputs. Subsequently, determine the optimal relationship between inputs and nonlinear operators to construct the underlying physical dynamics.” We refer to this as a “Liquid” connection, wherein inputs from various scenarios serve as the foundation for modifying operator relationships, especially in “Sparse data” environments. This “Liquid” connection is based on the theory of gated neurons proposed in Section 5.2.1. The entire **morphological discovery** process is then demonstrated and validated through a case study involving the discovery of robotic arm dynamical regularities in Section 5.2.2 and 5.2.3.

### 5.2.1 Gated neuron theory for liquid NN

This approach partly contradicts the principles of cognitive science, which advocate for the flexible application of learned knowledge. In practice, models should adapt and create variations in learned knowledge, akin to how water conforms to the shape of different containers. This flexibility is not solely rooted in attention mechanisms, as illustrated in Fig. 5.3.

In attention-based mechanisms, attention maps highlight specific regions of the input using varying shades of color, allowing the model to focus on relevant information. However, this attention mechanism does not directly modify the network’s internal structure. In contrast, the liquid-like adaptive mechanism we aim to develop in this section involves adjusting the layer weights and connections within the neural network based on the input. This process creates different activated connectivity pathways, much like how the human brain forms distinct



**Figure 5.3:** Differences between attention mechanism and liquid NN dynamics.

connections when processing various types of data.

Building on the mimetic theory discussed in Chapter 3, this section applies gated neuron theory to establish a dynamic knowledge representation mechanism. This mechanism serves as a foundation for pre-training and allows the network to dynamically adjust its internal structure in response to different inputs from different scenarios. To illustrate this approach, we use a dynamics modeling representation problem for a robotic arm in Section 5.2.3. This example demonstrates how the methodology can be applied to dynamically characterize a system by fitting predicted moments from joint angular velocities, angular accelerations, and angular displacements.

Generally, neural networks contain multiple neurons per layer, structurally identical but functionally distinct due to unique weights and biases. This diversity enables them to learn various features from input data, forming a distributed representation. In a trained network, neuron mappings between layers remain fixed, ensuring consistent input-output predictions, and making the model reliable and efficient.

This study argues that the “Liquid” mechanism involves establishing adaptive feature combinations within layers, creating input-controlled inter-layer connections, ensuring information flow controllability, and selectively transferring features between layers. Gated neurons, as shown in Fig. 5.4, are proposed as the basic units executing these dynamics.

In Fig. 5.4, inputs at different stages are shown. The raw input represents the original data, such as sensor readings or pixel values. The layer input is the processed information from previous layers. The state input retains historical context from previous steps, capturing temporal dynamics. The gate value  $g$  regulates the influence of two feedforward layers,  $f f_1$  and  $f f_2$ , on the neuron’s state. This gating mechanism is a dynamic filter, determining how much past (state input) and present (layer input) information influences the current state. The output is a weighted combination,  $f f_1 \times (1 - g) + g \times f f_2$ , balancing immediate and historical data. This enhances the model’s understanding of temporal

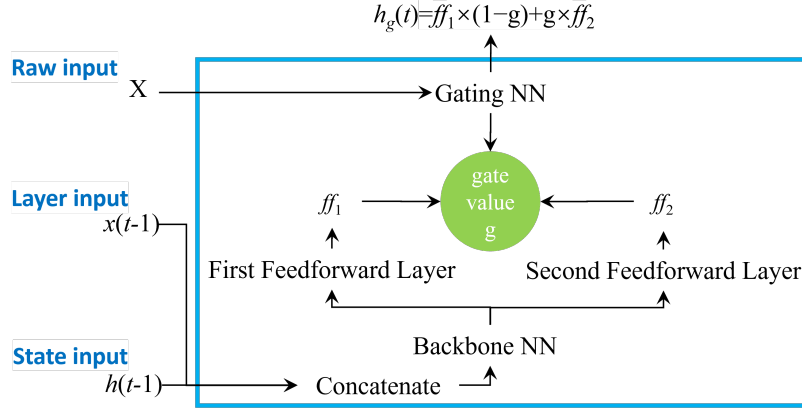


Figure 5.4: Liquid neuron design.

sequences and dependencies, improving predictions and decision-making. The proposed gate neuron can be integrated into layers like liquid-Conv1d or liquid-RNN cells, providing different properties for diverse tasks.

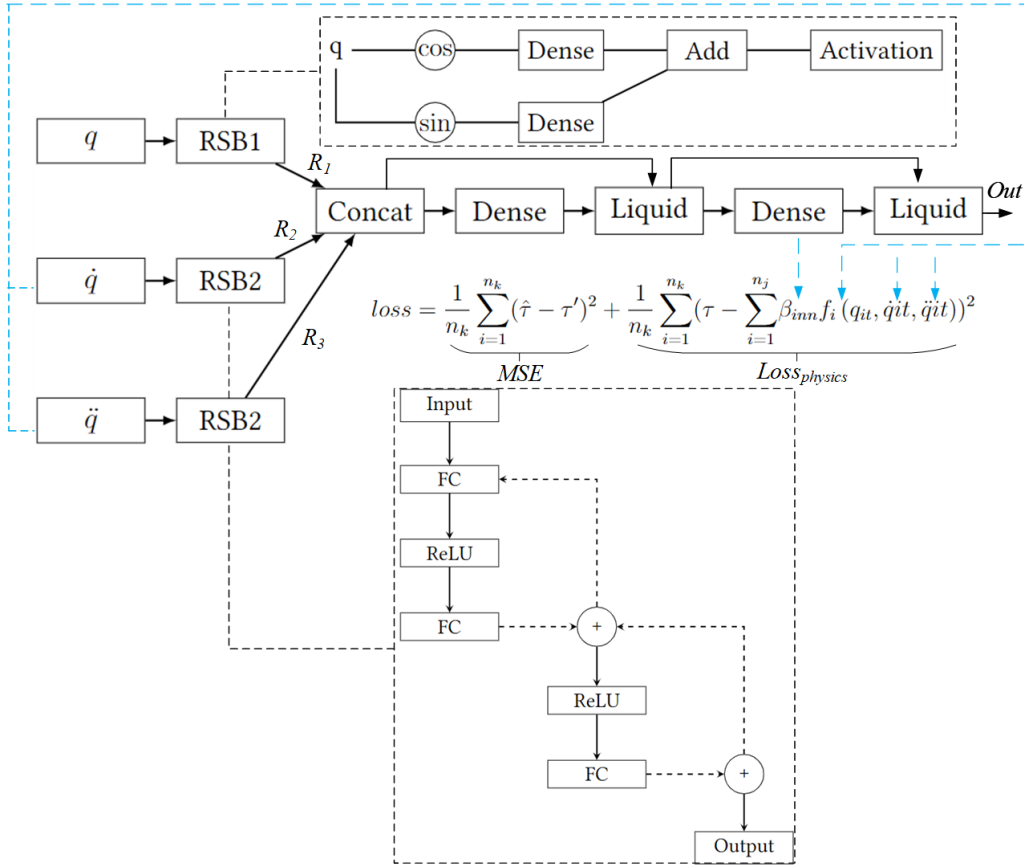
### 5.2.2 Discovering physics on unlabeled data

This section introduces a novel approach to physics discovery using an Equation Embedded Neural Network (E2NN), which is enhanced by the gated neuron theory. In E2NN, inverse dynamics equations are used to construct neural layers, allowing the network to encode physical knowledge directly through its activation functions and interconnections. The liquid-like mechanism within E2NN enables dynamic adaptation of interlayer connections based on the input data, which enhances the model’s real-time flexibility and performance.

To further strengthen this approach, we integrate the E2NN module with a deep residual shrinkage network (DRSN), resulting in the E2NN-ResNet model. This integration combines the physics-based structure of E2NN with the powerful feature extraction capabilities of DRSN, as illustrated in Fig.5.5. This proposed new model structure aims to improve both the accuracy and adaptability of the network in real-world applications. For comparison, the conventional DRSN model is given in Appendix E.1.

In detail, the E2NN enhances the DRSN model by innovatively modifying the activation function and interconnections within the residual blocks, incorporating unique blocks utilizing trigonometric functions, which are absent in conventional DRSN models. This integration of PI principles into ANN architectures significantly improves the model’s capability to mimic inverse dynamics processes under physical constraints. The model’s structure includes Residual

Shrinkage Blocks and a Liquid Layer. Specifically, the first Residual Shrinkage Block (“RSB1”) calculates the cosine of  $q$  (shortcut =  $\text{tf.cos}(q)$ ) and maps it to the filters via a dense layer, while the second block (“RSB2”) processes  $\dot{q}$  and  $\ddot{q}$ . A “Concat” layer merges the three outputs from the residual shrinkage blocks, and their products ( $R1 \times R1$ ,  $R1 \times R2$ ) are concatenated. The Liquid Layer then assimilates these components and approximates them to  $\tau$  in a mathematical form.



**Figure 5.5:** E2NN: Deep Residual Shrinkage Network with embedded equations.

### 5.2.3 Validation of the proposed methodology for dynamic identification in robotic arms

In robotic modeling, system modeling with limited joint data challenges both tradition PBMs and ML techniques. PBMs struggle with uncertainties, variable conditions, diverse configurations, and incomplete parameters. ML methods face issues with physical consistency, interpretability, and extensive data needs. This section introduces a novel approach: E2NN, enhanced by the proposed gated

neuron theory. E2NN uses inverse dynamics equations to construct specialized neural layers, with activation functions and interconnections as composition operators, encoding physical knowledge. The Liquid mechanism allows dynamic adaptation to changing inputs and motion equations, enhancing flexibility and performance.

The metrics used to evaluate the proposed methods are MSE, Polygon Area Difference, Fréchet Distance, and Time cost. These metrics' definitions are detailed in Appendix E.2.

### 5.2.3.1 Test E2NN performance on real data

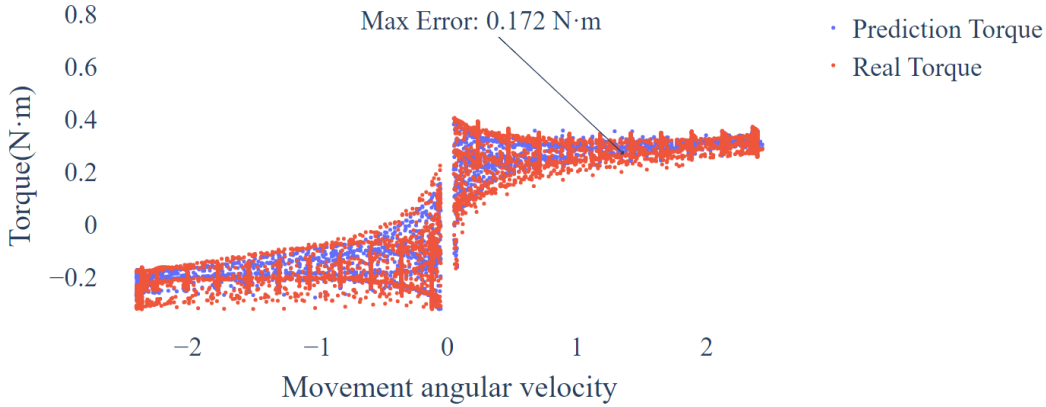
This investigation meticulously evaluates and compares various ML methodologies for robot manipulator torque estimation. We selected methods suited for small sample sizes and complex data structures typical in robotics. These include classical algorithms like K-Nearest Neighbors (KNN) and Support Vector Machine (SVM), known for their efficacy with smaller datasets. We also employ Deep Multilayer Perceptron (MLP) and DRSN for their capabilities in handling high-dimensional data, and XGBRegressor for its proficiency in regression tasks. Additionally, Nonlinear Regression with Lasso Regularization is used, treating each sub-term of the torque equation as an individual operator. Finally, the E2NN-enhanced DRSN, as depicted in Fig. 5.5, stands as a testament to the integration of PIML into this multifaceted comparative study. Their performance is given in Table 5.1.

**Table 5.1:** Validation on real-world data.

Method	Metric			Maximum error (N·m)
	MSE	Area Difference	Fréchet Distance	
Deep MLP	0.00272	3.998	0.249	0.510
SVM	0.575	9.791	1.513	2.263
XGBRegressor	0.00247	3.703	0.223	0.391
KNN	0.00442	4.029	0.209	0.528
DRSN	0.00314	4.835	0.160	0.374
Physics estimation	0.00542	6.519	0.256	0.572
E2NN	<b>0.00103</b>	<b>1.248</b>	<b>0.173</b>	<b>0.172</b>

According to Table. 5.1, it appears that KNN, with an MSE of 0.00442, Area Difference of 4.029, and Fréchet Distance of 0.209, would be a strong contender in this application. However, a deeper analysis reveals while KNN does outperform SVM (which has a significantly higher MSE of 0.575 and an Area Difference of 9.791), it falls short when compared to more advanced methods like Deep MLP and DRSN. The performance gap is particularly noticeable in the context of complex dynamics modeling for robotic arms, where Deep MLPs and DRSNs excel

due to their multi-level data representation capabilities. These models, with their advanced feature extraction and noise tolerance abilities, are especially adept at handling the intricate interplay between various input and output variables, such as angular displacement and torque.



**Figure 5.6:** Prediction results on real robot manipulators.

Considering Fig. 5.6, which presents scatter plots comparing the predicted torque and real torque as a function of movement angular velocity, we observe that the E2NN model significantly outperforms all others. It achieves a MSE of 0.00103, making it the most accurate model. The E2NN not only provides a better overall fit to the trajectory, but its joint torque predictions exhibit minimal deviations, with only a few minor errors and the absence of any large, severe deviations.

### 5.2.3.2 Investigation on the E2NN’s robustness

This section employs data generated under the “Friction” working condition to evaluate the robustness of the E2NN. This process involves applying the benchmark model and the E2NN, which have been trained on the same dataset, directly to the new test without any additional training. The size of the new test set is 8996 samples. The prediction results of the two models are presented in Table 5.2. During the steady-state motion of the robot manipulators in the angular velocity range of -2 to 2, it can be observed that the benchmark DRSN model produces large outlier points and extremely unstable predicted curves. The maximum error of DRSN model is 5.3, which is higher than that of the E2NN, and there are significant outliers in the slewing process around -3 and 3. The benchmark model shows that the prediction results deviate significantly from the observation in the

**Table 5.2:** Comparison of the robustness of different methods across various metrics when applied to new data.

Method	Metrics			Response time	Parameters
	MSE	Area Difference	Fréchet Distance		
DRSN	1.1	14.4	8.2	$1.2 \times 10^{-4}$	66753
E2NN	0.3	1.6	1.6	$8.8 \times 10^{-5}$	56223

enlarged view of the entire steady-state movement formation. In addition, the E2NN can fit the actual trajectory with promised trend tracking.

More validation on the benefits and the exploration of E2NN are provided in Appendix E.3 using both simulated and real-world data from a 7-DOF KUKA robotic manipulator. In joints' dynamic response simulations, E2NN outperformed the benchmark DRSN model, achieving lower MSE (0.5 vs 0.6), Area Difference (8.3 vs 42.9), and Fréchet Distance (14.3 vs 56.7). The E2NN's capability to identify inverse dynamics parameters was evaluated by comparing the weights of its embedded operator's neural network layer with the known  $\beta$  values from simulated data. The average accuracy of parameter estimation, measured as MAE, was approximately 0.05433. The reconstructed trajectories using E2NN's weights demonstrated high torque fitting accuracy, reaching 97.1%. These results show that E2NN could effectively simulate the actual behavior of the robot manipulator, bridging the gap between theoretical modeling and practical applications.

### 5.3 E2E data-driven model for cross-scenario time series processing

As discussed in previous chapters, the performance of the generic parallel PIML framework hinges on its data-driven branches. This section focuses on constructing an optimal data-driven branch model suitable for various datasets. Prior studies have examined models such as Temporal Convolutional Neural Networks (1.2 MB), CNN-LSTM (3.2 MB), and Dilated CNN (78.3 MB), each demonstrating varying degrees of effectiveness and efficiency. However, these models lack the capability to handle long sequences and varying length inputs.

Traditional RNNs offer a solution by considering only the previous hidden state and current input, avoiding the need to recalculate hidden states. This enables fast inference and scalable context length but impedes parallel training and can lead to information loss over time. Recent advancements emphasize Transformer models, known for their capacity to capture long-term dependencies and parallelize computations, rendering them promising for time series processing. While



Transformer models excel in complex data within a context window, they face challenges with fixed context windows, which limit their ability to capture dispersed information and impose significant computational costs, particularly for long sequences in high-frequency time series monitoring (see Appendix E.4). In PHM applications, where computational resources are constrained, achieving computational efficiency while managing long-range dependencies remains crucial.

Therefore, we explored models like the selective state space (SSM)-based calculation units in combining RNN and Transformer benefits, allowing parallel training, information retention, and linear time complexity growth with sequence length. Therefore, **this section develops a generic time series processing model based on a gated SSM**. This model is suited for various sample lengths, measurement characteristics, time scales, and device monitoring scenarios.

### 5.3.1 Gated selective state spaces mechanism

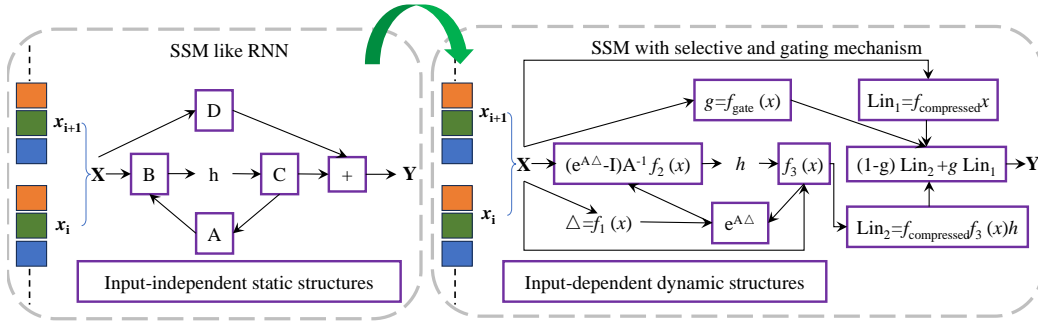
**The model proposed in this section introduces inputs-related model hidden status selective mechanisms based on the parallelization transformation of RNNs.** We detailed the derivation progression from continuous state space representation to the design of SSM models for machine learning, and the incorporation mathematical basis of gated selectivity for discrete ML parallel computing units in Appendix E.5.

$$\begin{aligned}
y_n &= Ch_n \\
&= C(\bar{A}h_{n-1} + \bar{B}u_n) \\
&= C(\bar{A}^{n-1}(\bar{A}h_0 + \bar{B}u_1) + \bar{A}^{n-2}\bar{B}u_2 + \cdots + \bar{B}u_n) \\
&= C\left(\bar{A}^n h_0 + \sum_{k=1}^n \bar{A}^{n-k} \bar{B}u_k\right) \\
&= C\bar{A}^n h_0 + \sum_{k=1}^n C\bar{A}^{n-k} \bar{B}u_k
\end{aligned} \tag{5.1}$$

The hidden states update process of RNN is given in Eq.(5.1), where  $\mathcal{A}$ , the State Transition Matrix, governs how the current state evolves into the next state;  $\mathcal{B}$ , the Input Matrix, dictates the influence of external inputs on the system state;  $\mathcal{C}$ , the Output Matrix, links the internal states  $h$  to observable outputs; and  $\mathcal{D}$ , the Feedforward Matrix, represents the direct impact of inputs on outputs. The problem with RNNs is that the matrix  $A, B, C$  does not vary with the input, and targeted inference on the inputs is not possible.

The selective mechanism proposed in [180] vary its attention this issue by highlighting that the tradeoff between efficiency and effectiveness in sequential models depends on state compression. Moreover, applying selective mechanism in state space model is computational fridndly which has a small, fixed while effective hidden states, retaining all necessary contextual information.

During training,  $\Delta$  is a time-varying parameter that reflects the discretization scale of the state space model at different time steps. It plays an important role in the state update process and influences the state discretization and computation. It is generated from the input through the projection module and can be dynamically adapted to the characteristics of the input data, thus forming the selection mechanism, as shown in Fig. 5.7. It adjusts the parameters  $\bar{A}_t$ ,  $\bar{B}_t$ ,  $\bar{C}_t$ , and  $\Delta_t$  based on the current input  $x_t$  and previous hidden state  $h_{t-1}$ . This dynamic adjustment helps the model focus on relevant features and dependencies at each time step.



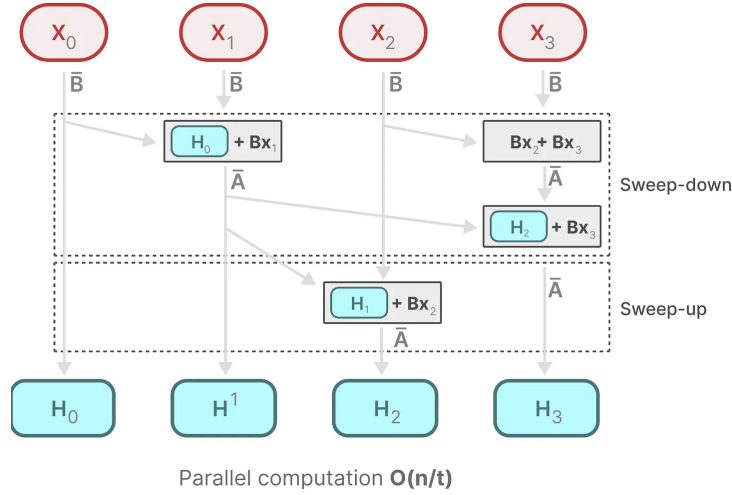
**Figure 5.7:** Selectivity mechanism to construct input-dependent state matrices.

$(\bar{A}, \bar{B}, \bar{C}, \bar{D})$  are state matrices at SSM.

To address selective long-term dependencies, we introduce a gated skip connection mechanism. This retains part of the input information, similar to residual connections, optimizing relevant information retention for task representation. As shown in Fig. 5.7,  $f_{gate}$  and  $f_1$  use activation functions in the range  $(0,1)$ , while  $f_2$ ,  $f_3$ , and  $f_{compressed}$  are linear layers for information compression.

The parallelization of the computation of the above processes, in particular for the optimization of the core hidden states, is implemented using a hardware-aware parallel scanning algorithm, as shown in Fig. 5.8.

As summarized in [182], the Sweep-up and Sweep-down phases can be performed by hardware with different computational speeds. During Sweep-up, data is processed and reduced in parallel chunks, using fast-access SRAM in GPUs. The Sweep-down phase combines and finalizes results, efficiently handled by HBM (High Bandwidth Memory) for larger data movements. This completes



**Figure 5.8:** Parallel scanning algorithm for accelerated computation to update hidden state [181].

the design of the gated SSM unit.

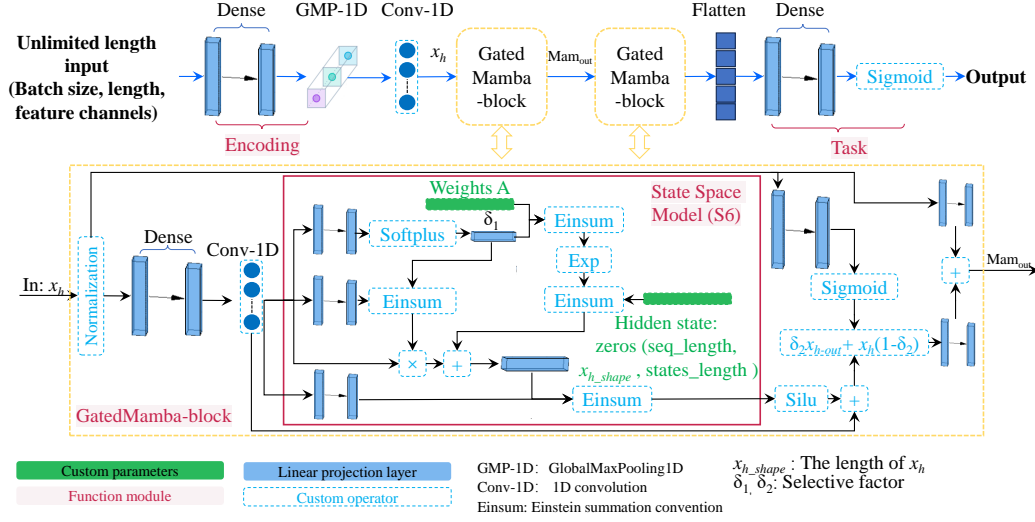
### 5.3.2 End-to-End (E2E) CNN-SSM model

Based on the SSM units designed in Section 5.3.1, we constructed a **new generation of end-to-end time-series data processing model for varying length sequences, directly handling raw data from various degradation scenarios**. Normally, the basic three-dimensional structure of the input time-series data is defined as (number of samples, sample length, and monitoring channels), corresponding to the total number of monitoring windows, the size of each window, and the number of physical quantities being monitored. However, the time series, collected during discrete monitoring processes, have varying sampling rates. For instance, rotating machinery requires sampling at least 2.5 times the maximum fault frequency, while battery temperature and voltage-current, considered slow variables, may be sampled at 1 Hz. High-frequency time-series data from long-term monitoring often have low information density, necessitating different lengths of monitoring windows to ensure sufficient information content.

The variability in sample numbers and lengths poses significant challenges for tradition deep learning models in PHM. These models struggle to compress and extract information from varying length sequences, requiring samples to be segmented using a sliding window approach and then padded. This ensures consistent input dimensions, with the variable dimension being the number of samples, allowing for batch processing of different sizes.

Current DL models lack the capability to handle diverse lengths and sampling

### 5.3. E2E data-driven model for cross-scenario time series processing



**Figure 5.9:** End-to-End CNN-SSM model architecture for processing variable input length sequences.

frequencies in real-world data, necessitating preprocessing steps like sliding window segmentation and padding. These steps can introduce noise, reduce feature extraction effectiveness, and increase computational burden due to redundant data points.

Furthermore, traditional models often have difficulty capturing long-range temporal dependencies in sequences of varying lengths. They use “Padding” to fill the sequence which disrupts temporal continuity, diminishing the model’s ability to learn sequential patterns crucial for accurate prognostics and health management. This limitation affects the model’s adaptability to different operational scenarios and fault conditions, undermining the reliability and robustness of predictive maintenance strategies.

The newest end-to-end model for processing varying length inputs is based on a CNN-LSTM-like stacked module, as verified in Chapter 3 and 4. This architecture is reused to construct the stacked CNN-SSM module. Additionally, the “S6” model from Mamba [180] is integrated for the hidden layer state update. The final architecture is given in Fig. 5.9.

The proposed model efficiently processes varying length sequences by integrating a sophisticated state space model within a gated block structure. The data flow begins with an encoding stage, where the unlimited-length input is passed through dense, GMP-1D, and Conv-1D layers. These layers are employed to reduce the dimensionality of the input data and extract relevant features, preparing

the data for subsequent processing. The encoded data then undergoes normalization, ensuring a consistent input distribution for the model, essential for stable training and efficient convergence.

The core architecture integrates a State Space Model (S6) to update hidden states. The S6 model uses operations like Einsum, Softplus, Exp, and Silu to compute state transitions. Custom weights  $A$  adjust these transitions via matrix calculations. Hidden states, initialized to zero, are updated at each time step based on the current input and previous hidden state, efficiently capturing and propagating information. The processed data then passes through linear projection layers and a Sigmoid activation function to produce the final output.

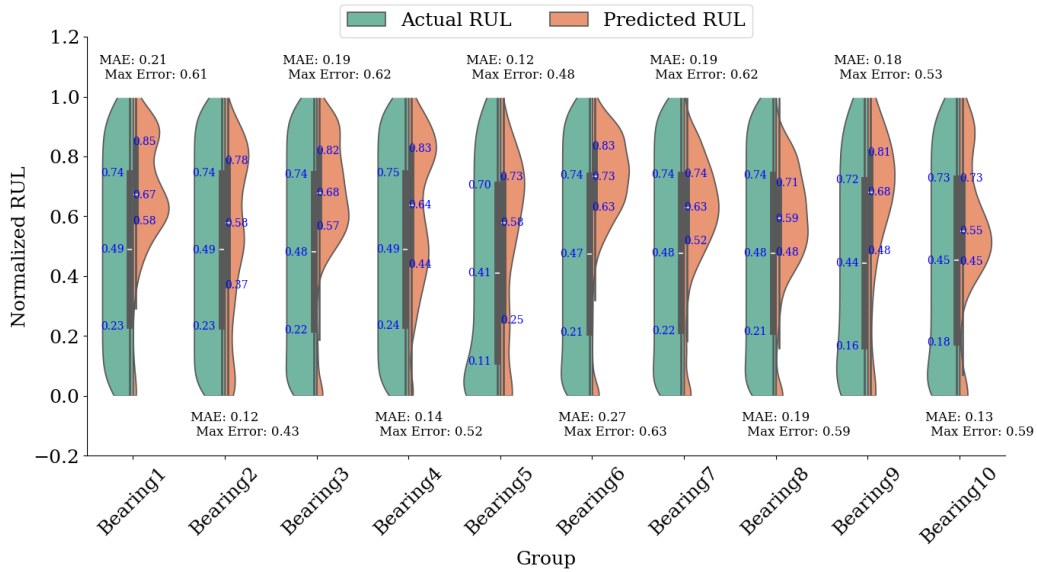
#### 5.3.3 Performance evaluation of E2E CNN-SSM model on different datasets

The proposed CNN-SSM model supports a maximum input shape of (4096, 4) to optimize computational resources. For the bearing dataset, inputs expanded from single-direction vibration signals to x-y dual-direction vibration signals. This expansion includes both vector amplitude and phase, with a raw data length of 2560. For the battery dataset, inputs expanded from 3 to 4 channels (voltage, current, operation time, and temperature) with a raw data length of 2500. Despite the larger input size, the CNN-SSM model has fewer parameters, resulting in a model size of 227 KB. Thanks to this, it can reduce the training time and shows efficiency and scalability compared to other models presented in the previous chapters. We show quantitative prediction results below and provide a further comparative analysis of qualitative trend predictions in Appendix E.6.

##### 5.3.3.1 Bearing prediction results

Fig. 5.10 presents the violin diagram to compare the predicted and true bearing RUL. Violin plots are powerful data visualization tools that combine the benefits of box plots and density plots to display data distributions. They provide a detailed view of data by showing density and variability, with the plot's width at any given value representing data density. Symmetric in shape, each violin includes a central box plot that highlights the median, first, and third quartiles, with whiskers indicating the data range and outliers clearly marked. In the context of RUL prediction, violin plots facilitate model performance evaluation by visually comparing predicted and actual RUL distributions, identifying discrepancies, and aiding model optimization. They also reveal data trends, variability, and outliers, which are essential for analysis and anomaly detection. Additionally, violin plots enable the comparison of multiple data groups, such as different devices or bearings, within a single plot.

### 5.3. E2E data-driven model for cross-scenario time series processing



**Figure 5.10:** Violin diagram for prediction error analysis of bearing RUL.

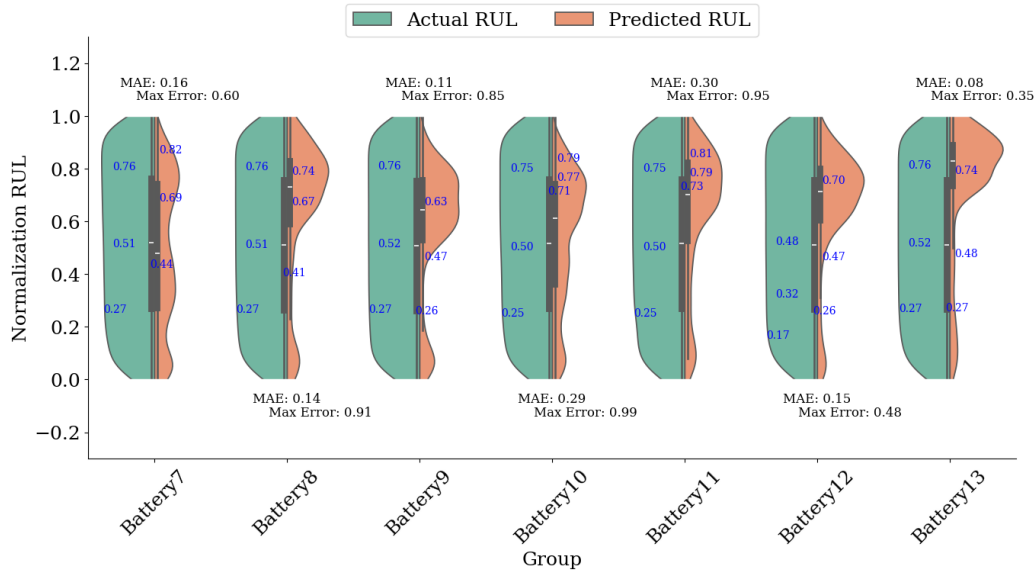
Fig. 5.10 presents violin plots comparing predicted and actual RUL values. Violin plots, which merge box plots and density plots, provide a detailed view of data distributions, showing density, variability, median, quartiles, and outliers.

Each violin in Fig. 5.10 represents a bearing group, with the width indicating data density and the central box plot showing median, quartiles, and potential outliers. The violin plots of actual and predicted RUL values are roughly similar, indicating the model's fair predictive ability. The distributions for most bearing groups, such as Bearing 1 and Bearing 5, align closely around central values, showing good predictive capability. The medians for several groups, like Bearing 3 and Bearing 6, are nearly identical, reflecting accuracy in predicting central tendencies. Moderate interquartile ranges, like those in Bearing 3 and Bearing 7, suggest relatively low prediction errors, with few outliers indicating stable performance, as seen in Bearing 5 and Bearing 10.

However, groups like Bearing 2 and Bearing 8 show higher predicted RUL values, indicating optimistic errors. Further optimization is needed for groups like Bearing 4 and Bearing 9, which exhibit anomalies and prediction deviations, suggesting the model tends to overestimate RUL in certain scenarios. Overall, while the model performs well, improvements are needed for better accuracy in extreme-value regions.

### 5.3.3.2 Battery prediction results

Fig. 5.11 shows that the predicted and actual battery RUL values are highly consistent across most battery groups, indicating good model accuracy. The distributions are similar, reflecting the model's ability to capture the actual decay process.



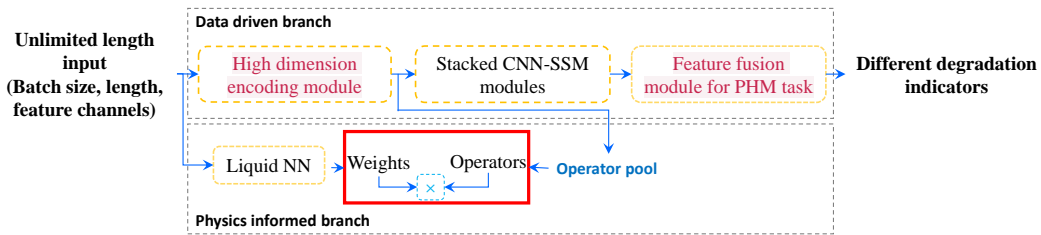
**Figure 5.11:** Violin diagram for prediction error analysis of battery RUL.

In Battery 1 and Battery 5, the predicted and actual medians overlap at 0.76, demonstrating high accuracy. Battery 2 has a higher maximum error (0.82), but the predicted median (0.77) is close to the actual (0.76). Battery 3 shows a predicted median (0.77) near the actual (0.73), with a maximum error of 0.54. Battery 4 has slight overestimation at higher RUL values, with both medians at 0.76 and a maximum error of 0.68. Battery 6 exhibits the best performance, with a low MAE of 0.12 and closely aligned medians (predicted 0.74, actual 0.76). These results show that while the model generally provides accurate predictions, there is room for improvement, such as Battery 2 and Battery 4.

The longer observation scale (2500s) for batteries, compared to bearings (0.1s), contributes to superior performance. Additionally, cleaner data from batteries, unconnected to other components, enhances model accuracy. This comparison highlights the importance of data relevance and the absence of external interferences in optimizing model performance.

## 5.4 A generic model incorporating the advanced PIML and SSL

Based on the previous sections' results, we propose a generic cross-scenario PHM model with active knowledge discovery capability. This model employs an improved multi-step training strategy: "Pretraining, physics-alignment learning, joint training" incorporating self-supervised learning during the physics-alignment phase. The model architecture is illustrated in Fig. 5.12, and the learning strategy is given in Fig. 5.13.



**Figure 5.12:** Generic end-to-end cross-scenario time series processing model for prognostic tasks.

In Fig. 5.12, the data-driven branch employs modules to encode and compress inputs into stacked CNN-SSM modules, generating feature fusions for the task of predicting degradation metrics. This branch leverages CNN-SSM modules to discover relevant patterns and representations from high-dimensional input data, which are crucial for the PHM task.

The physical information branch uses a liquid neural network to generate dynamic weights and operators, creating diverse knowledge representations that complement the data-driven branch. This branch includes a pool of basic linear and nonlinear transformation operators. By dynamically combining these operators as activation functions, the system achieves powerful approximation capabilities, allowing it to activate and express knowledge flexibly. The CNN-SSM modules in the data-driven branch guide the learning of the liquid neural networks by providing informational features and representations. Conversely, the physical information branch regularizes and directs the CNN-SSM modules to learn physically consistent patterns, ensuring coherence between data-driven insights and physical principles. The detailed structure is demonstrated by a lightweight liquid physics-informed CNN-SSM presented in Section 5.4.1.

The training method shown in Fig. 5.13 builds on the multi-step training method of PIML. In "Step 1", the data-driven branch of the model is pre-trained using input data. During this phase, the data-driven branch captures relevant



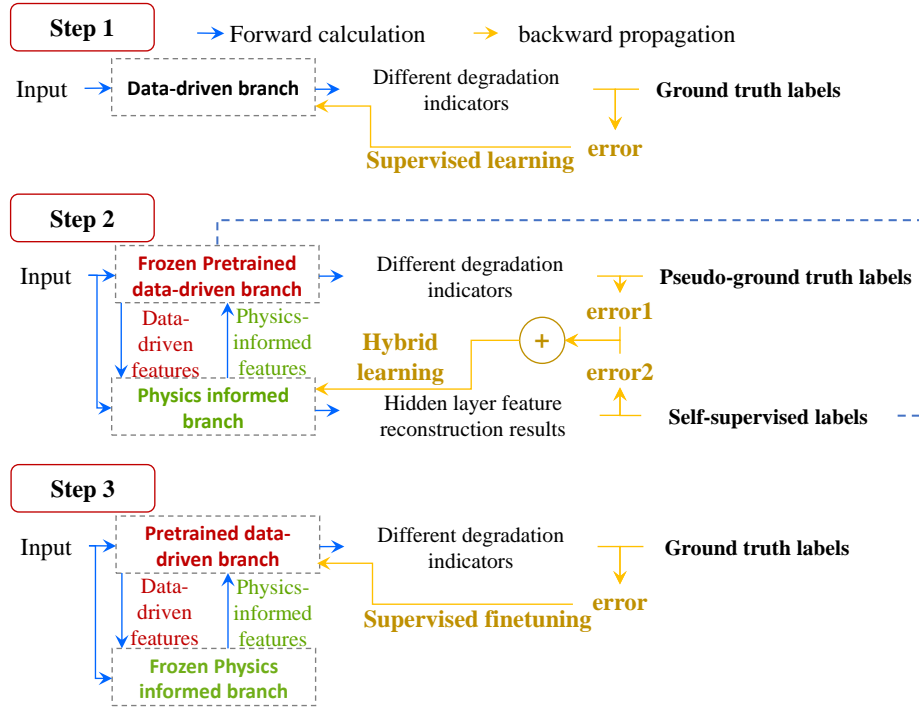


Figure 5.13: A multi-step hybrid learning strategy for generic model training.

features and representations through forward computation and backward propagation. These learned representations are then used to predict various degradation metrics, serving as true labels for supervised learning. This process ensures the model effectively learns from the input data and enhances its predictive capabilities.

“Step 2” introduces a novel learning approach, called physical alignment learning phase. Its crucial aspect is the application of self-supervised learning. During this step, the model is trained using a frozen pre-trained data-driven branch, which provides the necessary data-driven features. Additionally, a PI branch is introduced to take these data-driven features as input. Here, the encoded features before CNN-SSM processing serve as reconstruction targets, eliminating the need for external labels. The PI branch uses a codec-like architecture, with the encoder’s hidden layers containing rich degenerate representations [183]. The encoder incorporates an operator regression structure for autonomous discovery of physics knowledge, while the decoder remains data-driven. The model benefits from a hybrid learning approach, using both supervised ground truth labels and self-supervised reconstruction targets, enhancing data utilization efficiency. Details on generating pseudo-ground truth and self-supervised labels are in Section 5.4.2.

“Step 3” involves a joint training phase, where the pre-trained data-driven branch is unfrozen, and the physical information branch is frozen to preserve the knowledge learned from unlabeled data not present in the training set species. Subsequently, the data-driven branch is fine-tuned to deepen the application of the learned knowledge and enhance the fusion and matching of features from both branches. This fine-tuning process enables the model to make accurate predictions by leveraging both data-driven and physical information knowledge. By integrating these learning approaches, the model can achieve improved performance and robustness.

Ultimately, this modeling and training methodology will train a generic model for cross-scenario degradation prediction on the PHM dataset, meeting the 4C requirements. This will complete the prognostic task in Section 5.4.3, including predictions for bearing and battery RUL, tool wear, and composite fatigue cycle counts.

### 5.4.1 Liquid physics-informed CNN-SSM model

Based on the proposed theory and the validated CNN-SSM and liquid PIML structures discussed earlier in this chapter, our research innovatively establishes a liquid physics-informed CNN-SSM model, illustrated in Fig. 5.14. It is a comprehensive DL model architecture that incorporates both data-driven and knowledge-driven approaches. The model accepts an input sequence of unlimited length (batch size, length, feature channels). The design and data flow details of this generic model are clarified as follows:

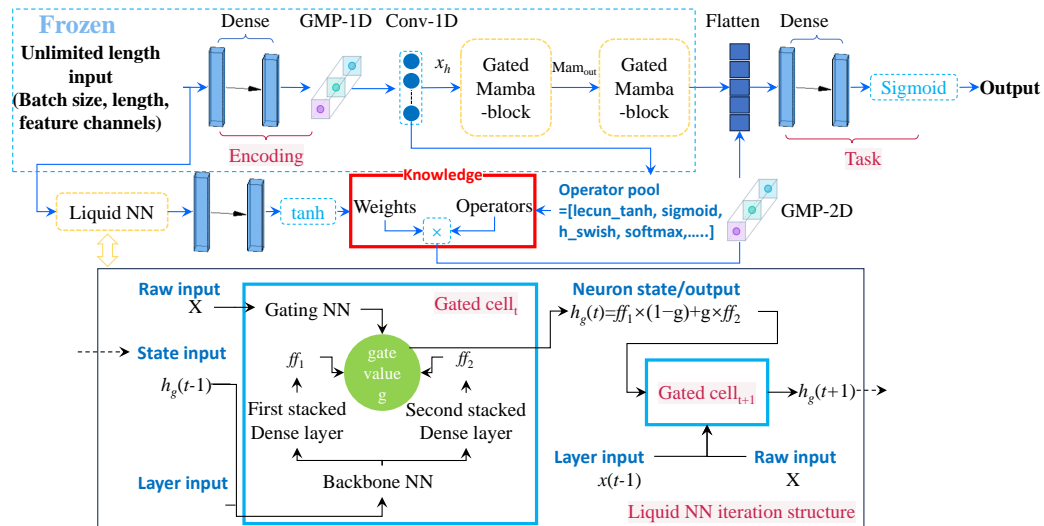


Figure 5.14: Auto-discovered knowledge informs CNN-SSM in liquid form.

The initial stage involves encoding, where the sequence is transformed by Dense, GMP-ID (Global Max Pooling ID), and Conv-ID (Convolution ID) layers into a compressed representation, enabling variant-input length processing. This encoded representation is subsequently refined through two consecutive Gated Mamba blocks, which utilize learned gates to selectively pass information. The output from the second Gated Mamba block is then flattened and passed through dense layers, culminating in a sigmoid activation function that produces the task-specific final output.

The model also includes a Liquid Neural Network (Liquid NN) component as the PI branch. In each iteration, the Liquid NN takes the previous layer input  $x(t - 1)$  and state input  $h(t - 1)$ , using a Gating NN to generate a gate value  $g$  that modulates information flow. The Backbone NN, comprising two dense layers, processes  $x(t - 1)$  to produce  $f2$ , which is combined with  $h(t - 1)$  in a Gated Cell using the function  $h(t) = f1 \times (1 - g) + g \times f2$ . A GMP-2D (Global Max Pooling 2D) layer in the knowledge component provides a global summary of prominent features. The core of the model dynamically expresses knowledge through input-dependent weights, generating various forms of knowledge by combining operators differently. This informs the data-driven branch by adding the PI feature to the final output decision module. This integration enhances the model's capability to handle complex tasks and capture intricate patterns.

To further enhance the model's capability, a GMP-2D (Global Max Pooling 2D) layer is included in the operator pool of the knowledge component. This layer performs a max pooling operation over the spatial dimensions of the input, providing a global summary of the most prominent features. Through these mechanisms, the model achieves a robust integration of data-driven encoding, iterative refinement via Liquid Neural Networks, and knowledge-driven augmentation. This enhances its ability to handle complex tasks and capture intricate patterns in the data.

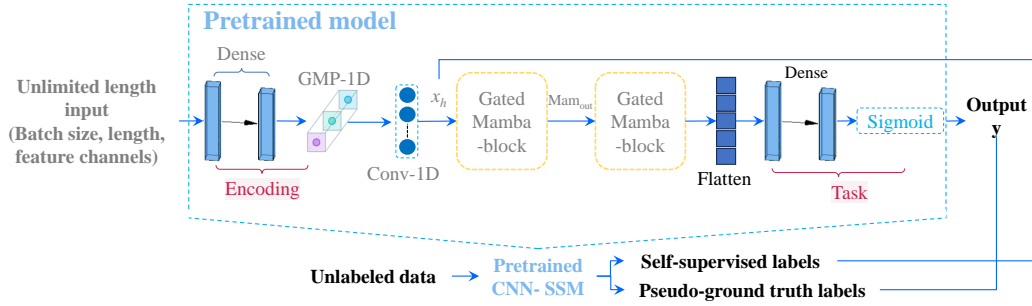
Overall, the core design of the model aims to minimize the need for extensive scaling by creating rich, input-dependent computational dynamics. This approach forms a more expressive base module compared to traditional models, enabling the model to handle diverse and complex data more effectively.

## 5.4.2 Hybrid learning strategy

### 5.4.2.1 Supervised information generation

The supervised information used in the SSL is generated by the methodology shown in Fig. 5.15.

By using  $y$  as pseudo-ground truth labels, the model can align its predictions with the actual task objectives, ensuring consistency between the self-supervised



**Figure 5.15:** Generating labels for unlabelled data for supervised learning.

learning and the supervised learning goals. The encoded features  $x_h$  from the pre-trained CNN-SSM modules in the data-driven branch contain rich information about the input data. By using  $x_h$  as self-supervised labels, the model learns to reconstruct these features, thereby capturing the underlying patterns and representations of the data.

Generally, the codec-like approach uses the input data itself as the reconstruction objective, learning to reconstruct the high-level representations. For long sequence inputs, reconstructing the hidden layer output  $x_h$  is more computationally efficient than reconstructing the raw data. In addition, the hidden layer output  $x_h$  from the pretrained CNN-SSM modules represents high-level features and patterns. By reconstructing these representations, the PI branch aligns itself with the informative abstractions captured by the pretrained CNN-SSM modules.

#### 5.4.2.2 Simultaneous supervised and self-supervised learning

In “Step 2” hybrid learning, the proposed model expands the single-output structure to a dual-output architecture. During training, both the labeled and unlabeled data are used. The loss function consists of two parts as shown in Eq. (5.2). Both loss and val\_loss contain the supervised degradation indicator prediction loss and the self-supervised reconstruction loss. The term  $(2 - (\text{loss}_y + \text{loss}_{x_h}))$  is used to weight the validation loss based on the training loss, allowing the model to dynamically adjust its learning.

The sum\_loss design offers significant advantages over tradition validation loss minimization, especially when using both labeled and unlabeled data. It dynamically balances supervised and self-supervised learning, adjusting the model’s focus based on training loss. When training loss is high, sum\_loss enhances feature representation learning. When training loss is low, it improves performance on labeled data. This approach mitigates overfitting by weighting validation loss more heavily when needed, guiding the model toward better generalization. Additionally, incorporating self-supervised tasks encourages the learning of robust,

semantically meaningful features, crucial for generalization on unlabeled data. The `sum_loss` also accelerates convergence by effectively using unlabeled data from the start, unlike traditional methods. Moreover, it provides a more flexible early stopping criterion by considering both training and validation losses, ensuring superior generalization performance.

$$\text{sum\_loss} = \text{loss}_y + \text{loss}_{x_h} + (2 - (\text{loss}_y + \text{loss}_{x_h})) \times (\text{val\_loss}_y + \text{val\_loss}_{x_h}) \quad (5.2)$$

Where the suffix  $x_h$  represents the supervised validation loss on labeled data, aiding the model in learning meaningful hidden representations. The suffix  $y$  measures the model’s performance on pseudo-labeled data for the main prediction task. The prefix *val* is the validation loss.

The addition of the hidden feature reconstruction decoder makes the model structure heavy. To address this, in “Step 3” Joint training, the model is adjusted to have a single output (degradation indicators) without the decoder part, making it lightweight again. This is possible because the self-supervised reconstruction task has already helped the model learn meaningful hidden representations during the previous training stage.

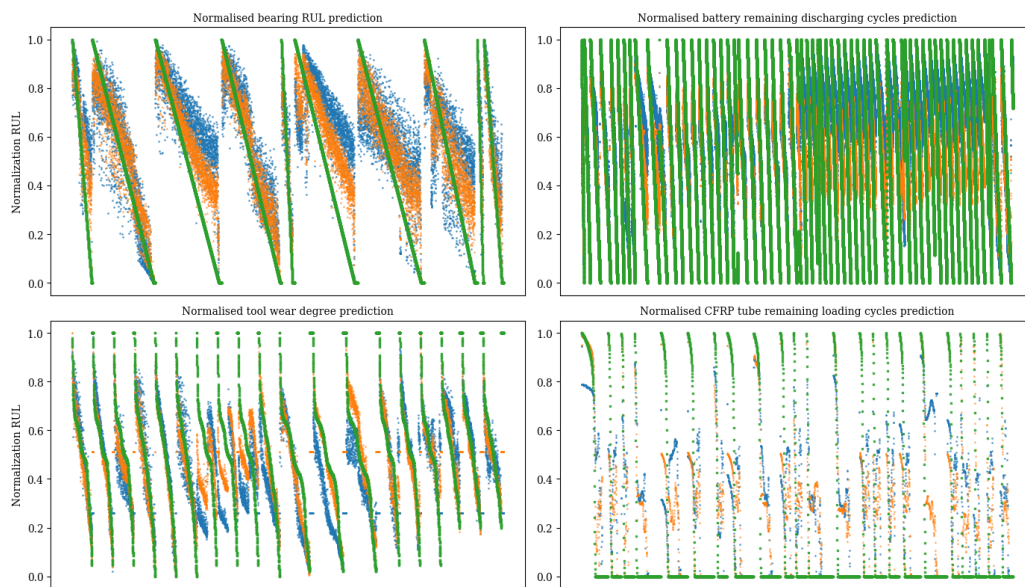
### 5.4.3 Investigating the performance of liquid PI CNN-SSM model

This section investigates the performance of the Liquid physics-informed CNN-SSM model using a mixed cross-scenario dataset that fulfills the “4C” requirements. Table 5.3 provides an overview of the dataset’s characteristics. The mixed dataset’s significance and composition, highlighting its cross-scenario nature and the advanced methodologies used in the analysis, are introduced. For each dataset, no data preprocessing is done other than the normalisation of its prediction targets.

The model’s training process is optimized by dynamically monitoring and adjusting key parameters. Monitors are set for several tasks: One halts training if there is no improvement in training loss for a specified number of epochs, preventing overfitting and saving resources. Another saves the model’s weights whenever the custom loss metric improves, preserving the best version of the model. A learning rate monitor reduces the rate if the custom loss metric shows no improvement for a set number of epochs, helping to fine-tune the learning process and avoid plateauing. Additionally, a custom `sum_loss` monitor recalculates the custom loss metric at the end of each epoch by combining training loss and weighted validation loss, ensuring validation performance is considered. These measures create a robust system for effective training, improving generalization, and preventing issues like overfitting and stagnation.

## 5.4. A generic model incorporating the advanced PIML and SSL

---



**Figure 5.16:** Prediction results of the same CNN-SSM with Physics-informed CNN-SSM on multiple datasets.

**Note that,** Green: Ground truth value, Blue: CNN-SSM results, and Yellow: PI-CNN-SSM results.

#### 5.4. A generic model incorporating the advanced PIML and SSL

**Table 5.3: Cross-scenarios dataset introduction.**

<b>Cross-scenarios content</b>	<b>Data characteristics</b> (All on time series signal, end-to-end model, without feature extraction)
Different studied systems	<b>Prognostia dataset</b> (Bearing aging [178]), <b>MIT dataset</b> (Battery aging [175]), <b>Tool wear</b> (Milling process tool wear [184]), <b>Composite material</b> (Fatigue data for carbon fibre-reinforced polymers [185])
Diverse sampling frequencies	<b>Bearing</b> (25.60 kHz), <b>battery</b> (1 Hz), <b>Tool wear</b> (500 Hz, 25 kHz), <b>Composite material</b> (150-200 Hz)
Multiple physics measurements	<b>Bearing</b> (Vibration directions), <b>battery</b> (Voltage, Current, Temperature, Charging time), <b>Tool wear</b> (Position control, tool position, motor torque, cut force), <b>Composite material</b> (Axial position, Force, Rotation, Torque)
Varying sample lengths	<b>Bearing</b> (2560), <b>battery</b> (2500), <b>Tool wear</b> (512), <b>Composite material</b> (1024)
Diverse prediction targets	<b>Bearing</b> (Remaining useful life), <b>battery</b> (Remaining discharging cycles), <b>Tool wear</b> (Wear degree), <b>Composite material</b> (Remaining cycles)
Suitable input length and maximum number of channels	Supports 4K inputs (Max 4096 points), 4 channels
Hardware friendly	Mamba module (Linear computational cost) + Liquid physics branch (Input-based weights), processing 23.9 GB data with 450 KB model size, supports parallel computation

**Table 5.4: Updated prediction accuracy of different models.**

Methods	Bearing degradation	Battery aging	Tool wear	Composite fatigue
CNN-LSTM	0.789	-	-	-
CNN-LSTM (SSL)	<b>0.927</b>			
CNN-SSM	0.830	0.79	0.85	0.77
SEI-DCN	-	<b>0.91</b>	-	-
PI CNN-SSM(Hybrid learning)	0.900	0.87	<b>0.91</b>	<b>0.86</b>

The results in Table 5.4 and Fig. 5.16 compare model performances for bearing degradation, battery aging, tool wear, and composite fatigue. The table shows error rates, highlighting the CNN-LSTM model's improvement with SSL, achieving the lowest error rate of 0.073 in bearing degradation. The proposed CNN-SSM model has moderate performance with error rates from 0.15 to 0.23. The Physics-

informed CNN-SSM (PI CNN-SSM) model excels with the lowest error rates in tool wear (0.09) and composite fatigue (0.14). The SEI-DCN model performs well in battery aging with an error rate of 0.09.

Considering Fig. 5.16, the green lines present the ground truth values, the blue ones indicate the CNN-SSM results, and the yellow ones show PI-CNN-SSM results. For bearing RUL prediction, both models closely follow the ground truth, but the PI CNN-SSM model's predictions align more tightly, indicating superior performance. In battery discharging cycles, while both models exhibit good fits, the PI CNN-SSM consistently provides more accurate predictions. Similar trends are observed in tool wear and CFRP tube remaining loading cycles, where the PI CNN-SSM model's predictions are more precise compared to the CNN-SSM model. These visual results corroborate the quantitative findings, emphasizing the PI CNN-SSM model's effectiveness, particularly due to the incorporation of physical knowledge.

However, while the PI CNN-SSM model often outperforms the CNN-SSM model, it does not consistently do so across all datasets. This inconsistency suggests that the model's performance could be further improved by incorporating operators with greater approximation expressiveness.

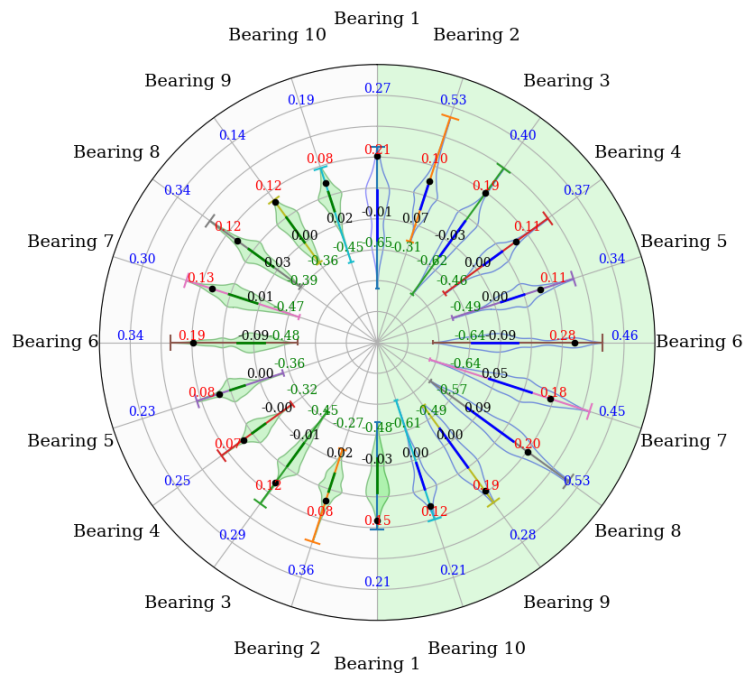
We further demonstrate the change in model performance before and after incorporating autonomously discovered physical knowledge on unlabeled data in different scenarios using polar violin plots. These plots combine kernel density estimation with box-and-line plots to display data distributions and probability densities. The violin-shaped curve represents the data distribution probability, with wider sections indicating higher data density. The middle box plot shows the five-number summary: the maximum and minimum of non-outliers, the upper and lower quartiles, and the median. Figures 5.17, 5.18, 5.19, and 5.20 illustrate the results: the left half shows Physics-informed CNN-SSM after hybrid learning and joint training, while the right half shows CNN-SSM with supervised learning only.

Fig. 5.17, 5.18, 5.19, and 5.20 compare the prediction errors between the Physics-Informed CNN-SSM model and a generic data-driven model.

We can see from the violin plots that significant differences exist in the error distributions between the two models across all 10 bearings. Compared to the data-driven model, the PIML model generally exhibits smaller error ranges, as indicated by the more compact violins. This suggests that incorporating physics knowledge into the machine learning model leads to more consistent and accurate RUL predictions. Looking at the specific error metrics, the PIML model consistently outperforms the data-driven model. The MAE, represented by the red lines, is lower for the PIML model in all cases. The 3rd quartile of errors (black lines) is also significantly lower for the PIML model, indicating that 75% of the PIML model's predictions have smaller errors than the corresponding percentile



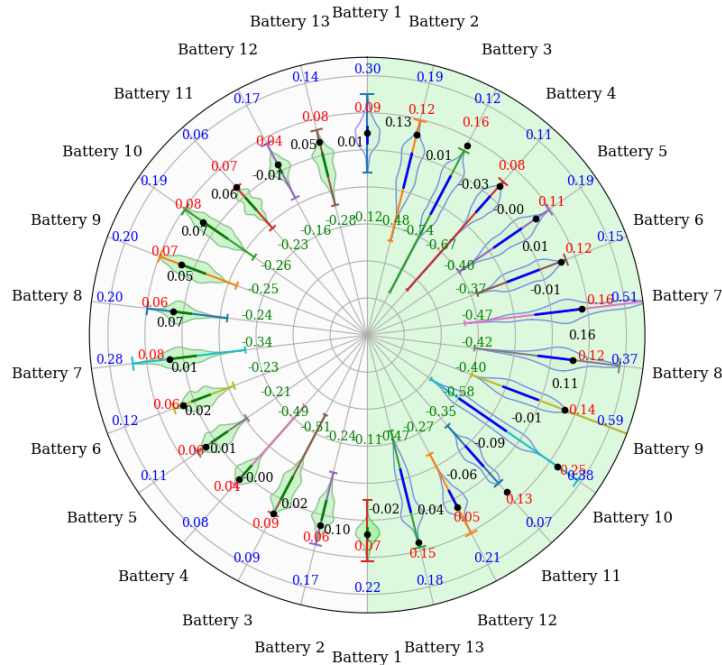
5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.17:** Performance difference in bearing RUL predictions before and after physics learning.

In order along the radius: green - Min error value, red - MAE, black - 3rd quartile, blue - Max error value, Polar axis range  $[-1, 0.8]$ . The left semicircle is the result of CNN-SSM, and there semicircle is the result of Liquid PI CNN-SSM.

of the data-driven model.

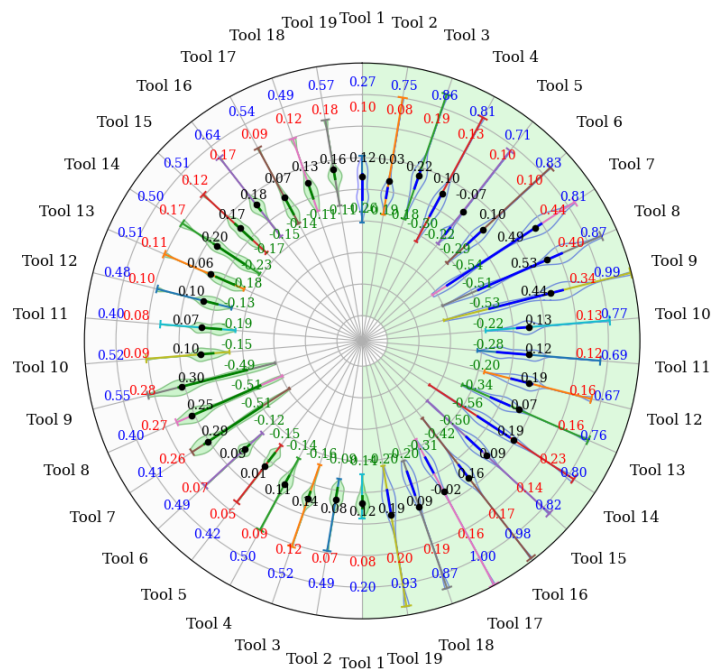


**Figure 5.18:** Performance difference in battery RUL predictions before and after physics learning. Polar axis range  $[-1, 0.5]$ .

The position of the error distributions relative to zero is also noteworthy. For the data-driven model, the violins are mostly centered above zero, with the 3rd quartile line (black) often above zero. This indicates a bias towards conservative predictions, where the model frequently underestimates the remaining useful life. In contrast, the PIML model's error distributions are more centered around zero, suggesting less bias and more balanced predictions.

It's important to note that the prediction error is calculated as the true RUL minus the predicted RUL. Therefore, positive errors indicate conservative predictions (underestimation of RUL), while negative errors indicate hazardous predictions (overestimation of RUL). The PIML model's more balanced error distribution, with the majority of the 3rd quartile lines close to zero, demonstrates its ability to make more accurate and less biased predictions compared to the purely data-driven approach.

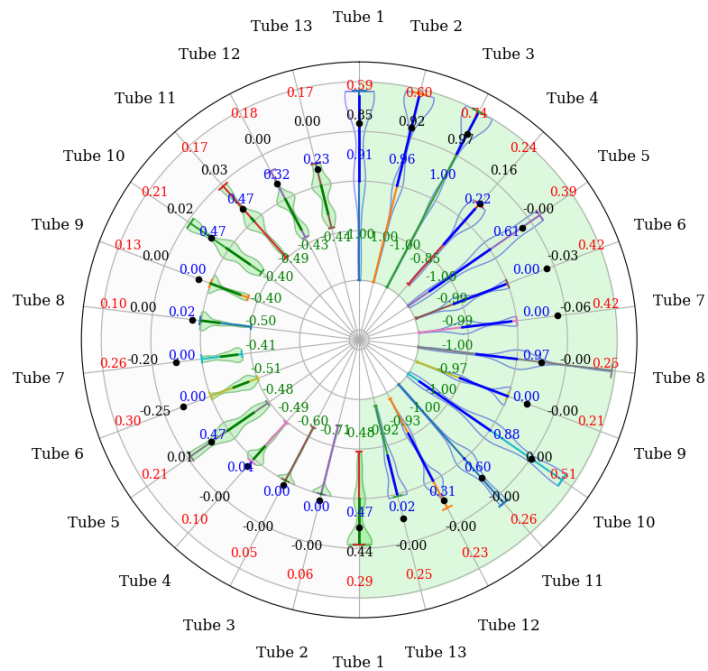
## 5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.19:** Performance difference in tool wear predictions before and after physics learning.

**Note that,** Polar axis range  $[-1, 0.5]$ .

## 5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.20:** Performance difference in CFRP tubes remaining loading cycle predictions before and after physics learning.

**Note that,** Polar axis range  $[-1, 0.5]$ .

#### 5.4.4 Interpretable outcomes of knowledge representation after Hybrid learning

In addition to predicting results, we examined the interpretable knowledge learned by the model within hidden layers. By analyzing the output weights of specific “Liquid NN” layers, we investigate how these weights combine different operators under various input situations. This analysis demonstrates the model’s knowledge representation process and enhances understanding of its internal mechanisms. Using Bearing 1, Bearing 2, and Battery 1 cases, we loaded the weights of the PI CNN-SSM training model, created sub-models to freeze specific layer weights, applied different activation functions, and visualized the weight changes in Fig. 5.21, 5.22, 5.23, and Appendix E.6.

The adjustment of operator weights with degradation is common, showing the model’s dependence on specific embedded features throughout the process. This dependence is both operator-specific and channel-specific. For example, in Fig. 5.21, 5.22, and 5.23, the predictions of  $2\frac{x^5}{15}$  on different bearings in different directions are consistently consistently show higher activation in channels 20 to 32. This indicates that operator combination weights vary for unmonitored physical quantities of the same test object, and knowledge representation patterns differ for the same monitored quantities across different objects.

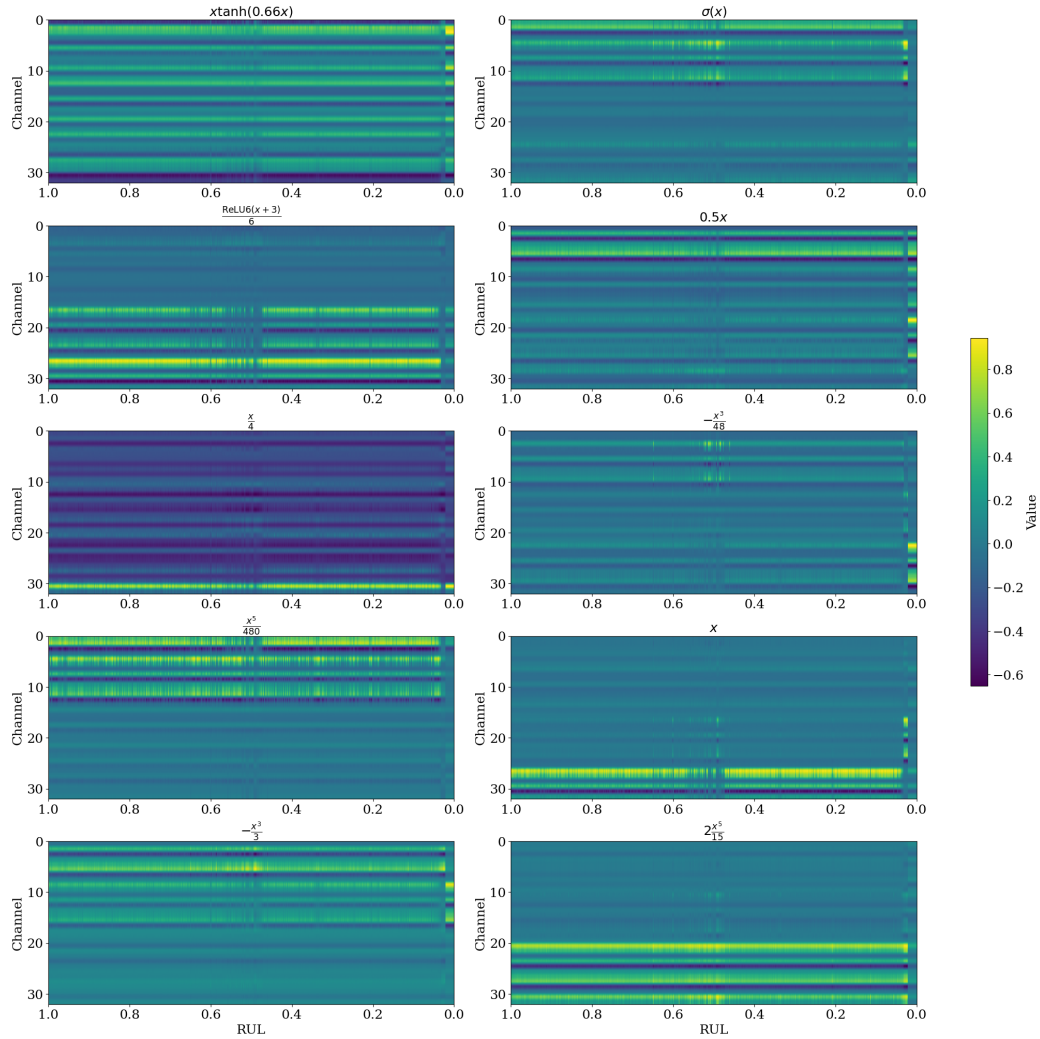
The attention of the model is shifted between different operators, e.g., in the near-failure phase, we observe a significant activation of the “0.5x” operator, which reflects the learning and adaptation of the model to the changing degradation patterns in the data. In addition,  $\frac{\text{ReLU}_6(x+3)}{6}$  shows good generality and smoothing properties, receiving long-term attention across different signals and devices. For bearing degradation, operators with higher-order powers of the input data receive the main attention, while for batteries, the focus is primarily on lower-order powers.

In addition, we find that the abrupt shifts in attention seen in bearings are absent in battery predictions, as shown in Fig. 5.21, 5.22. For bearings, attention jumps across channels and operators, while for batteries, certain channels consistently receive attention. We believe this discrepancy arises because bearing measurement signals contain more noise. As bearings approach failure, the fault signals become much stronger than the noise, causing attention shifts similar to the sudden amplitude increases in the time series signals near failure.

All of this supports the chapter’s initial point that **knowledge should be dynamic**. tradition PIML paradigms embed physical knowledge in fixed representations, limiting their application and validation. This static approach relies heavily on trial and error and empirical methods, underscoring the need for more flexible, dynamic models.

Furthermore, the diverse operator combinations across channels on individ-

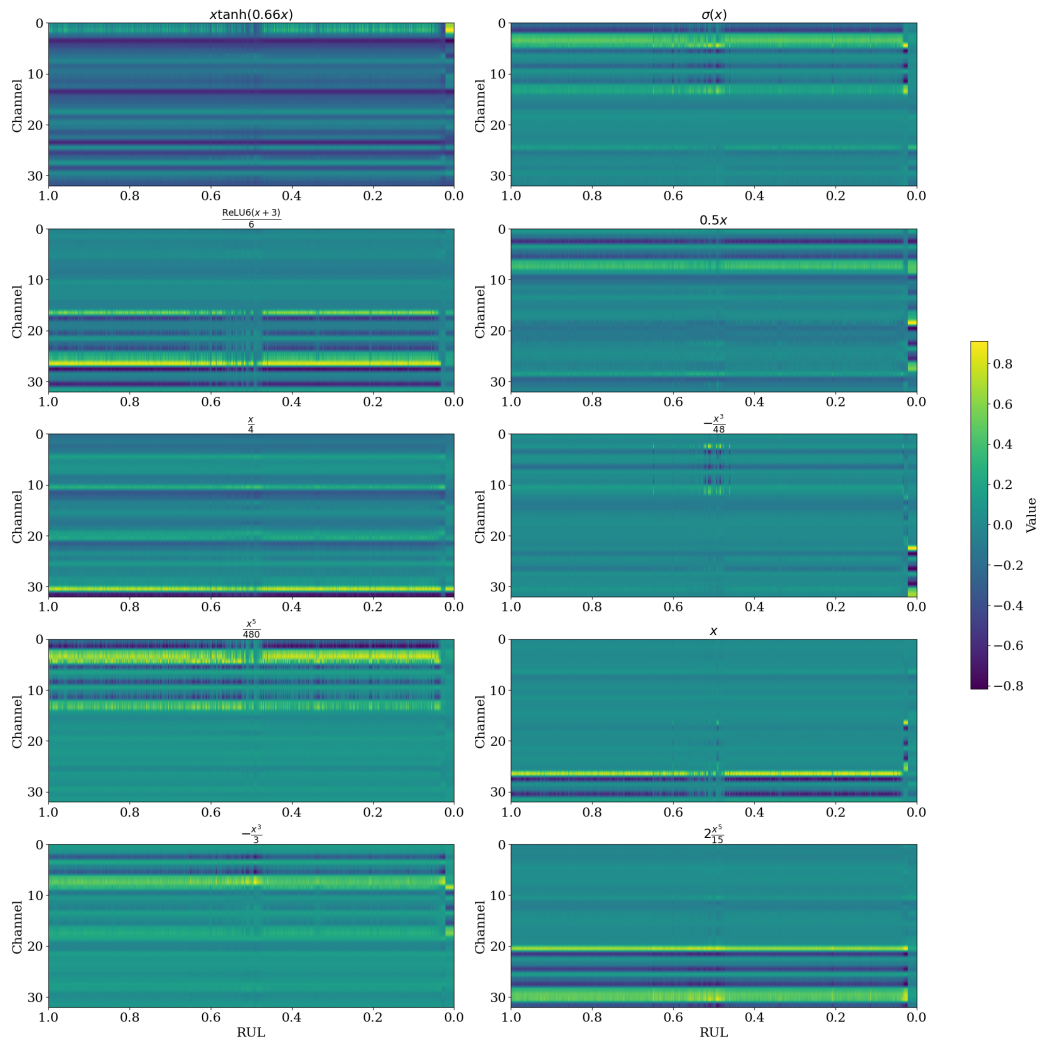
## 5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.21:** Operator weight changes in the actively discovered knowledge about the horizontal vibration during 1st bearing's degradation.

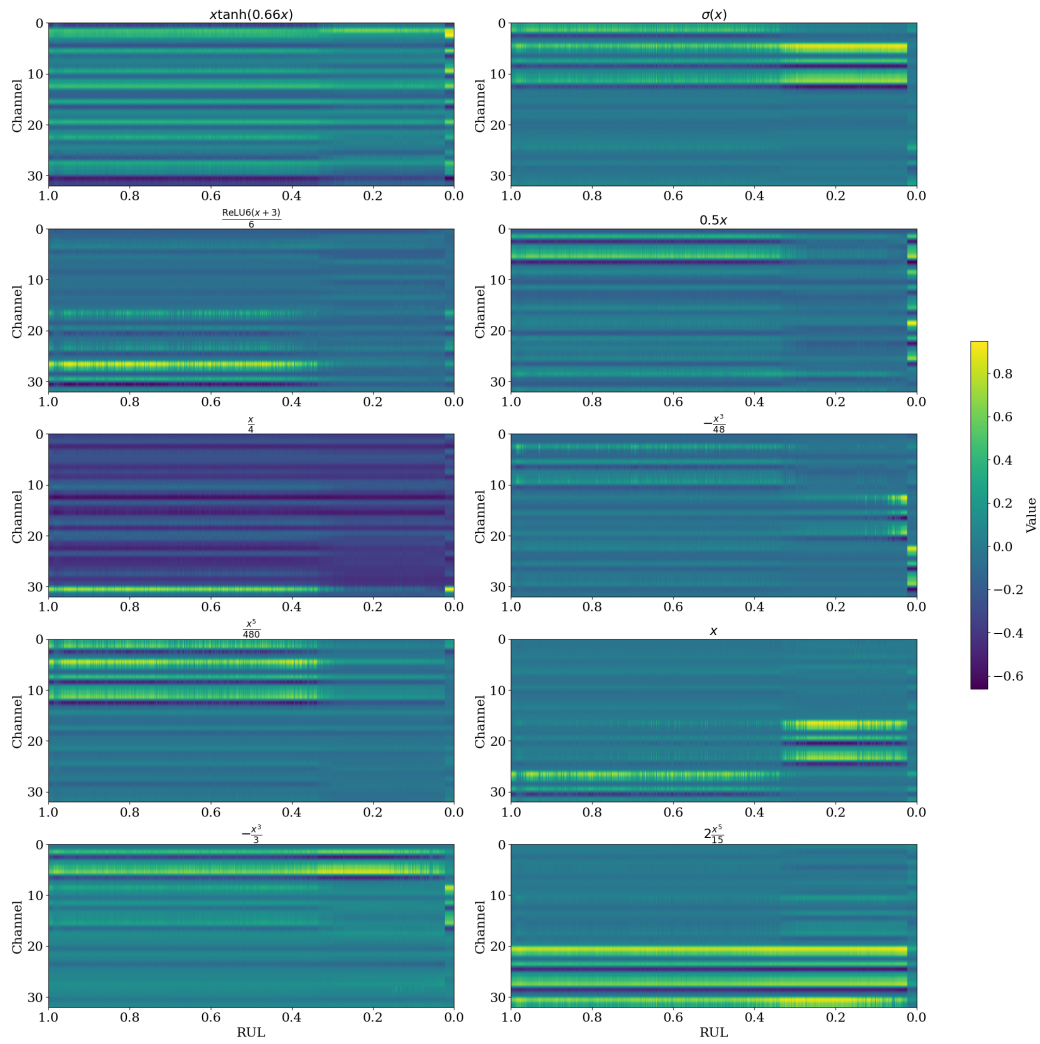
**Note that,** the vertical axis represents the feature channel after initial data embedding and operator processing in the high-dimensional space. The horizontal axis represents the remaining lifetime decay process, with colors indicating the weights assigned to the operators.

## 5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.22:** Operator weight changes in the actively discovered knowledge about the vertical vibration during 1st bearing's degradation.

## 5.4. A generic model incorporating the advanced PIML and SSL



**Figure 5.23:** Operator weight changes when actively discovering knowledge from horizontal vibration signals during 2nd bearing's degradation.



ual weight maps indicate that tradition PIML struggles to rely solely on the non-linear fitting capabilities of machine learning. This often necessitates stacking complex, large-parameter deep structures, which hampers interpretability and generalization. Combining different operators through channel weights offers a cost-effective approach to multivariate and diverse knowledge discovery in AI for science. This method enhances the generalization of hidden layer representations, aligning them more closely with known physics.

## 5.5 Conclusion

This chapter presented a comprehensive approach to developing a generic, end-to-end cross-scenario physics-informed SSL model with active knowledge discovery capabilities. It breaks through tradition PIML and SSL boundaries and makes the following major contributions:

- Extending PIML to active knowledge discovery using gated neuron theory and liquid neural networks. This allows for dynamic and flexible knowledge representation, adapting to different input data. A physics discovery-oriented unlabeled mining approach using E2NN-enhanced is developed by applying gated neuron theory. E2NN uses inverse dynamics equations to construct specialized neural layers, encoding physical knowledge through activation functions and interconnections.
- Proposing an end-to-end data-driven model called CNN-SSM for cross-scenario time series processing. This model efficiently handles varying length sequences and supports a maximum input shape of (4096, 4) to optimize computational resources.
- Introducing a multi-step hybrid learning strategy that combines supervised and self-supervised learning. This strategy enhances data utilization efficiency and improves the model’s performance by leveraging both labeled and unlabeled data with the downstream task matching knowledge discovery as part of the SSL learning task.
- Validating the proposed liquid physics-informed CNN-SSM model on a cross-scenario dataset that fulfills the “4C” requirements. The model demonstrated superior performance compared to purely data-driven approaches.
- Analyzing the interpretable knowledge learned by the model in the hidden layers reveals the dynamics of combining weights of different operators in various input situations. This analysis enhances the understanding of the model’s internal mechanisms and knowledge representation.

By eliminating the need for a priori knowledge and domain-specific expertise, and enabling knowledge discovery from unsupervised data, this approach addresses the issue of scarce knowledge in PHM. We aim to connect the discovered knowledge to real-world problems and develop more expressive knowledge operators. Additionally, by training models on both real and simulated data, we can leverage the ability of these generic models to accept raw sequences of variable lengths without the need for extensive data processing and augmentation, effectively addressing the issue of limited data.

# Conclusion and future work

---

## Contents

---

<b>6.1</b>	<b>Summary of our contributions</b>	<b>145</b>
<b>6.2</b>	<b>Discussion of our key findings</b>	<b>147</b>
6.2.1	Insights and challenges in PIML and SSL for PHM applications	148
6.2.2	Building generic PHM model by thinking outside the box	149
6.2.2.1	A generic architectural approach to PIML modeling	149
6.2.2.2	SSL pretext task design considering downstream PHM tasks and information	149
6.2.2.3	Breaking the old PIML and SSL paradigms to solve the problem of sparse data and scarce knowledge	150
<b>6.3</b>	<b>Limitations and Future work</b>	<b>151</b>
6.3.1	Limitation of the existing research	151
6.3.2	Future perspectives	152

---

Our thesis offers multiple insights to answer the research question: How to build a generic PHM model under sparse data and scarce physics knowledge conditions. We have presented the novel theory, algorithms, and training strategies of PIML and SSL, along with their applications to various PHM problems. This chapter has 3 sections. We summarized our contributions in Section 6.1, discussed their insights in Section 6.2, and recommended future research directions in Section 6.3.

## 6.1 Summary of our contributions

Our research contributes to the field in three key ways: enhancing existing methodologies, introducing and validating new approaches, and accelerating practical implementation.

For the existing PIML strategy, our contributions are:

- **Mimetic theory proposal:** We proposed “Mimetic Theory”, which replicates physically-driven data flows by customizing ML operators and inter-operator connections based on physical knowledge. This forms a standard methodology for constructing PIML (Section 3.2).
- **Constraint projection strategy:** We developed a constraint projection strategy to address the flaws of existing PIML methods lacking safeguard mechanisms for embedding poor physical knowledge. This strategy ensures physical consistency and performance gains during the learning process, validated through sequential feedback training in reinforcement learning (Section 3.3).
- **Generic PIML model architecture:** We proposed a generic PIML model architecture integrating constraint projection and response sequential learning (Section 3.4). This architecture features both physically-driven and data-driven parallel branches, presenting a new perspective on embedded knowledge and data-driven models. The “Pretraining, Physics Alignment, Joint Training” strategy guarantees PIML model to have the better optimization during the learning process, using the original data-driven model’s performance as a lower bound and the quality of embedded knowledge as an upper bound to solve the problem of embedding bad knowledge.

For the existing SSL strategy, our contributions focus on:

- **Contrastive learning for RUL prediction:** We leveraged degradation characteristics in time-series data, using the differences in positive and negative order of samples along the temporal direction as a contrastive feature extraction strategy. This method underpins the effective application of SSL in RUL prediction (Section 4.2).
- **Upstream-downstream task alignment:** We addressed the mismatch between upstream and downstream tasks in SSL within PHM by integrating downstream task-related models into upstream pretext tasks. This enhances self-supervised training objectives with downstream task variants, ensuring that features learned upstream are functionally relevant and applicable to PHM tasks (Section 4.3).

Beyond optimizing existing paradigms from structural simulation, computer vision, and natural language processing, we further extend PIML and SSL in Chapter 5:

- **Active knowledge discovery:** We advanced PIML by incorporating active knowledge discovery using gated neuron theory and liquid neural networks. This enhancement allows for dynamic and adaptable knowledge

representation tailored to varying input data, reducing reliance on expert experience and making knowledge acquisition and PI structuring more efficient.

- **Hybrid learning strategy:** We proposed a multi-step hybrid learning strategy that combines supervised and self-supervised learning. This approach significantly improves data utilization and model performance by leveraging both labeled and unlabeled data. It aligns knowledge discovery with SSL task requirements, standardizing the learning process and enabling a seamless flow of information between labeled and unlabeled data.

Finally, our proposed methodology and model extend beyond theoretical and laboratory validation to practical applications. We aim for our model to serve as a prototype or basic architecture for a generalized PHM model. To ensure practical applicability, we considered the high computation efficiency and fast reasoning requirements essential for PHM deployment. Our model was validated in scenarios meeting the “4Cs” requirements: cross-machine, cross-monitoring timescales, cross-physical measurements, and cross-prediction targets. This comprehensive validation demonstrates the model’s robustness, adaptability, and potential for real-world implementation in diverse PHM tasks.

- **Efficient computational resources:** We developed the CNN-Selective State Space Model, which efficiently processes variable-length time series with a maximum context input shape of (4096, 4). It operates effectively on onboard computing power, maintaining a compact model size while handling huge data.
- **Easy implementation and versatility:** Our liquid physics-informed CNN-SSM model serves as a generic solution for cross-scenario datasets that meet the “4C” requirements. It delivers superior performance, approaching state-of-the-art levels of customized PIML models, and offers clear interpretability of physical knowledge within its hidden space.
- **Scalable knowledge discovery:** This approach facilitates knowledge discovery from unsupervised data without requiring prior domain expertise, addressing knowledge scarcity in PHM. It processes raw data sequences of variable lengths without the need for preprocessing or augmentation, simplifying implementation and enhancing data utilization.

## 6.2 Discussion of our key findings

This section aims to delve into the critical insights and challenges identified in our research. We will discuss the implications of our findings, highlight the ad-

vancements made, and address the ongoing challenges.

### 6.2.1 Insights and challenges in PIML and SSL for PHM applications

In Chapter 2, we conducted both qualitative and quantitative analyses of the PIML and SSL approaches. Our research proposes a hybrid model development roadmap with PIML and SSL as key approaches for handling sparse data and scarce knowledge in PHM. We identified that PIML is predominantly applied in material damage, aerospace, production equipment, and power grids, leveraging extensive existing mathematical and physical modeling research. Conversely, SSL finds applications in fault diagnostics and surface defect detection, with increasing utilization of 2D image-based data.

**Our literature review found that the existing studies focus on architecture development and different application cases without addressing methodology foundation and deployment issues.** Particularly, PIML research has primarily integrated domain knowledge with advanced algorithms to enhance robustness and accuracy. These studies aim to solve complex problems governed by partial differential equations and tackle issues like scalability, training stability, and discontinuities. Concurrently, SSL research has emphasized improving model generalization in low-data scenarios, innovating architectures, and refining loss functions to better capture self-supervised representations. **Both fields continue to face challenges such as computational complexity, data quality sensitivity, and the integration of complex physical laws and multi-physics phenomena.**

Our research also revealed that PIML applications across different fields often use similar mathematical formulas as the embedded knowledge. However, these applications show both common and unique variations in their overall structure, depending on the specific task. Furthermore, few studies address the impact of poor-quality degradation knowledge in PIML, suggesting a need for a more generic architectural approach and a clearer perspective on the relationship between embedded physical knowledge and ML models.

For SSL, there is a disconnect between upstream pretext tasks and downstream PHM tasks, highlighting the necessity for integrating downstream tasks into upstream training. Additionally, PIML and SSL are highly complementary, with PIML benefiting from physical knowledge and SSL capable of learning fixed knowledge by pretext tasks. However, many PHM studies fail to explore this synergy, often relying on existing knowledge rather than discovering it through SSL strategies. These insights underscore the need for innovative approaches to enhance the robustness, generalization, and practical application of PIML and SSL

in PHM.

## **6.2.2 Building generic PHM model by thinking outside the box**

### **6.2.2.1 A generic architectural approach to PIML modeling**

Chapter 3 demonstrated that various PIML architectures can be unified by introducing physics-based constraints at different stages of the ML pipeline. These constraints, integrated as specific input-output relationships, significantly impact model behavior, loss landscapes, and gradient calculations. Mimetic theory, which employs customized layers and connections to replicate physical structures and behaviors, emerged as a key approach. This theory was applied in the RFEMNN, which showed strong performance in diagnosing rotor compound faults by learning physical relationships in the rotor system matrices.

However, traditional PIML models sometimes deviate from physical principles due to the unique optimization landscape created by explicit gradients and higher-order derivatives. To address this, constraint projection theory was introduced, reformulating the optimization problem to include physical constraints. This method maintains model adherence to physical principles through a sequential learning process, enhancing the model's ability to generalize to unseen fault scenarios. The RFEMNN, fine-tuned with reinforcement learning, demonstrated exceptional diagnostics performance in zero-shot, one-shot, and few-shot settings.

The research identified the need for a generic PIML architectural solution. A dual-branch architecture was proposed, consisting of a data-driven branch and a physics-informed branch, integrated through a three-step training approach: data-driven pre-training, physics-informed alignment, and joint training. This framework ensures that the model retains the performance of data-driven models while incorporating physical knowledge. In battery RUL prediction, this architecture significantly improved prediction accuracy and generalization, achieving an average absolute error of 15 cycles with only 4 cycles of input data, highlighting its effectiveness and potential for broader application.

### **6.2.2.2 SSL pretext task design considering downstream PHM tasks and information**

Chapter 4 explored the challenges and solutions for learning from unlabeled data due to the difficulty and laborious nature of acquiring high-quality labeled data. A novel SSL framework was introduced to enhance feature learning and improve

model performance in PHM tasks by leveraging sequential information mining and aligning pretext tasks with downstream requirements.

Key findings include the successful application of a contrastive learning strategy to capture degradation trends by distinguishing features from different sequential directions. This approach, using a Siamese CNN-LSTM architecture guided by a custom contrastive loss function, effectively learns robust degradation feature representations. Incorporating failure time predictions as an intermediary step ensures that the learned features align with downstream RUL prediction tasks. Validation through a case study on bearing RUL prediction showed the framework’s superior performance, particularly when labeled data is scarce. The model demonstrated stable error distribution and outperformed state-of-the-art methods in median prediction error and error range. The obtained results highlight the critical role of considering downstream task information in pretext task design and the impact of model architecture on overall performance. It also underscores the necessity of identifying the most expressive parts of the pre-trained Siamese CNN-LSTM model for effective transfer to downstream tasks.

These findings highlight the potential for developing more adaptable and generalized prognostics models that can handle diverse operating conditions and work effectively with unlabeled data. Such a pre-training task design and model structure design strategy pave the way for future research in the field of SSL for PHM applications.

### **6.2.2.3 Breaking the old PIML and SSL paradigms to solve the problem of sparse data and scarce knowledge**

Chapter 5 applied the improved PIML and SSL, that we developed before to real-world scenarios, highlighting several key findings. First, the integration of gated neuron theory and liquid neural network layer connections into PIML has shown significant enhancements in dynamic and flexible knowledge representation, which is crucial for supporting self-supervised tasks effectively. Second, the development of an enhanced E2NN model for torque monitoring in robotic manipulators illustrated the effectiveness of active knowledge discovery. This approach aligns with the generic PIML model, proposed in Chapter 3, demonstrating the ability to discover inverse dynamics knowledge and improve diagnostics accuracy.

Additionally, the hybrid learning paradigm proposed in this chapter, which incorporates SSL into the “Pretraining, Physics Alignment, Joint Training” strategy, has proven to significantly improve model performance. By leveraging both labeled and unlabeled data, this approach enhances data utilization and aligns learning tasks with the physical knowledge required for RUL prediction.



Besides advancements in the concept expansions of PIML and SSL Chapter 5 also addressed the computation requirements during deployment. The hardware-efficient lightweight CNN-SSM effectively combined data-driven and physics-informed branches, capturing high-dimensional patterns and dynamically generating weights and operators to handle complex tasks and intricate data patterns. A key finding is its efficient use of computational resources, with parallel computing capabilities and linear scaling of computational complexity enabling large dataset processing without a significant increase in resource requirements. Furthermore, the proposed model is compact, processing large amounts of degenerate data (e.g., 23.9 GB) with a relatively small model size of only 450 KB. This allows reducing memory needs and speeds up inference, making it highly suitable for real-world deployment.

Finally, the liquid Physics-Informed CNN-SSM model has been validated in various degradation scenarios, meeting the “4Cs” criteria and achieving state-of-the-art prediction performance. It consistently outperforms the generic data-driven CNN-SSM model, demonstrating lower error rates in bearing degradation (0.10), battery aging (0.13), tool wear (0.09), and composite fatigue (0.14). Visualization through polar violin plots highlighted the model’s superior performance, showing smaller error ranges, lower mean absolute errors, and less biased predictions. These findings underscore the effectiveness of integrating physical knowledge into machine learning models, enhancing their accuracy and generalization capabilities across diverse real-world applications.

## 6.3 Limitations and Future work

Building on the previous key findings, this section addresses the limitations and outlines potential directions for future work. While our results demonstrate the transformative potential of models with enhanced internal dynamics and expressive capacity in the PHM field, several challenges remain.

### 6.3.1 Limitation of the existing research

While the existing research shows a lot of exciting potential, there are still some shortcomings that need to be improved.

1. **The operator pool needs optimization.** Currently, only one universal operator for  $h_{swish}$  has been identified, effective across various failure scenarios. This is attributed to the smoothness of  $h_{swish}$  and its strong universal function approximation in neural networks. However, the remaining operators are limited in their expressive power.

2. **The lack of an end-to-end information filtering mechanism.** This drawback affects operator weight changes during knowledge discovery about the current signal and at various stages of degradation. The weight change graph shows discontinuous changes resembling water mist, indicating noise interference. This section suggests that masking-induced signal noise, especially in high-frequency cyclic time-series vibrations, highlighted the absence of an effective information-filtering mechanism. The interference is not solely due to noise but also from irrelevant information affecting some operators.
3. **The current learning strategy is not efficient enough.** During the hybrid learning phase, the model doesn't make full use of the labeled data. Currently, the model only considers the validation loss on labeled data after training with pseudo-labeled data, which is a passive approach. This method lacks the active feedback mechanism found in reinforcement learning, where both strategy and value losses are used. A more proactive approach would involve using both types of data to generate feedback and guide the learning process.
4. **The current model cannot yet cross task types.** It does not achieve abnormal detection, fault diagnostics, and prognostics in a grand unification, and similarly lacks the multi-modal ability to time arbitrary format input grand unification.

### 6.3.2 Future perspectives

We posit that a model with greater internal dynamics and expressive capacity is more likely to transform the PHM field, similar to how transformer-based GPT has revolutionized NLP. Our bold prediction is that *“the expressive power of the basic computational unit determines the model’s intelligence and generalization”*. This vision in PHM requires extensive reflection and further development, with significant potential for future work in the following key areas:

- **Scaling law application.** Expanding the model’s scale and training it on more extensive computational resources could enhance its capacity to represent complex internal dynamics, following the scaling law principle in AI development.
- **Pre-trained datasets.** Developing pre-trained datasets encompassing diverse monitoring time series and modalities across the industry is crucial. Although proprietary data poses challenges, simulated and synthetic data can effectively facilitate knowledge learning.

- **Synthetic data exploration.** Leveraging a vast corpus of synthetic knowledge to train PIML and SSL models, and developing a Liquid CNN-SSM trained on cleanly degraded simulated data, represents a promising approach.
- **Cross-modal knowledge dynamics.** Aligning diverse data forms, including two-dimensional images and higher-dimensional video data, to capture cross-modal knowledge dynamics is vital for the development of generalized PHM models.
- **Integration of discovered Knowledge.** Integrating newly discovered knowledge with existing human knowledge obtained from experiments, observations, and simulations can achieve consistent expressions and insights, influencing the design of next-generation products and optimizing operation and maintenance strategies.

These future directions highlight the potential for further refining and expanding the approaches presented in this thesis, paving the way for more advanced and generalized PHM solutions.

# Bibliography

- [1] Nam Kim, Dawn An, and Joo-Ho Choi. *Prognostics and Health Management of Engineering Systems*. Jan. 2017. ISBN: 978-3-319-44740-7. DOI: [10.1007/978-3-319-44742-1](https://doi.org/10.1007/978-3-319-44742-1) (cit. on p. 7).
- [2] Fernando Brügge. “Predictive Maintenance Market: 5 Highlights for 2024 and Beyond”. In: *IoT Analytics* (Nov. 2023). URL: <https://iot-analytics.com/predictive-maintenance-market/> (cit. on p. 7).
- [3] Michael J Scott et al. “A systematic literature review of predictive maintenance for defence fixed-wing aircraft sustainment and operations”. In: *Sensors* 22.18 (2022), p. 7070 (cit. on p. 7).
- [4] Peng Zhang et al. “Marine systems and equipment prognostics and health management: a systematic review from health condition monitoring to maintenance strategy”. In: *Machines* 10.2 (2022), p. 72 (cit. on p. 7).
- [5] Vepa Atamuradov et al. “Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation”. In: *International Journal of Prognostics and Health Management* 8.060 (2017), pp. 1–31 (cit. on pp. 7, 8).
- [6] Khanh TP Nguyen, Kamal Medjaher, and Do T Tran. “A review of artificial intelligence methods for engineering prognostics and health management with implementation guidelines”. In: *Artificial Intelligence Review* (2022), pp. 1–51 (cit. on p. 8).
- [7] Enrico Zio. “Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice”. In: *Reliability Engineering & System Safety* 218 (2022), p. 108119 (cit. on p. 8).
- [8] Christos Skliros et al. “A review of model based and data driven methods targeting hardware systems diagnostics”. In: (2018) (cit. on p. 8).
- [9] Chuanjiang Li et al. “Small data challenges for intelligent prognostics and health management: a review”. In: *Artificial Intelligence Review* 57.8 (2024), pp. 1–52 (cit. on p. 9).
- [10] Behnoush Rezaeianjouybari and Yi Shang. “Deep learning for prognostics and health management: State of the art, challenges, and opportunities”. In: *Measurement* 163 (2020), p. 107929 (cit. on p. 9).
- [11] Samir Khan et al. “A review of physics-based learning for system health management”. In: *Annual Reviews in Control* 57 (2024), p. 100932 (cit. on p. 9).
- [12] Yigit A Yucesan, Arinan Dourado, and Felipe AC Viana. “A survey of modeling for prognosis and health management of industrial equipment”. In: *Advanced Engineering Informatics* 50 (2021), p. 101404 (cit. on p. 9).

- 
- [13] Liangwei Zhang et al. “A review on deep learning applications in prognostics and health management”. In: *Ieee Access* 7 (2019), pp. 162415–162438 (cit. on p. 9).
- [14] Erdem Altuntac et al. “Bottom-up neurogenic-inspired computational model”. In: *2023 IEEE BioSensors Conference (BioSensors)*. IEEE, 2023, pp. 1–4 (cit. on p. 9).
- [15] Chaomei Chen. *CiteSpace: a practical guide for mapping scientific literature*. Nova Science Publishers Hauppauge, NY, 2016 (cit. on p. 19).
- [16] Manuel Arias Chao et al. “Hybrid deep fault detection and isolation: Combining deep neural networks and system performance models”. In: *arXiv preprint arXiv:1908.01529* (2019) (cit. on pp. 20, 26, 185).
- [17] Wangpeng He et al. “A data-driven group-sparse feature extraction method for fault detection of wind turbine transmission system”. In: *Measurement Science and Technology* 31.7 (2020), p. 074008 (cit. on p. 20).
- [18] Weikun Deng et al. “Physics-informed lightweight Temporal Convolution Networks for fault prognostics associated to bearing stiffness degradation”. In: *PHM Society European Conference*. Vol. 7. 1. 2022, pp. 118–125 (cit. on pp. 20, 32, 185).
- [19] Donal P Finegan et al. “The Application of Data-Driven Methods and Physics-Based Learning for Improving Battery Safety”. In: *Joule* (2020) (cit. on pp. 20, 192).
- [20] Jinjiang Wang et al. “An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples”. In: *Renewable Energy* 145 (2020), pp. 642–650 (cit. on pp. 20, 28, 33, 185).
- [21] Jingren Zhou, Xin Hong, and Peiquan Jin. “Information fusion for multi-source material data: progress and challenges”. In: *Applied Sciences* 9.17 (2019), p. 3473 (cit. on p. 25).
- [22] Michael D Sangid. “Coupling in situ experiments and modeling—Opportunities for data fusion, machine learning, and discovery of emergent behavior”. In: *Current Opinion in Solid State and Materials Science* 24.1 (2020), p. 100797 (cit. on p. 25).
- [23] Marzia Sepe et al. “A physics-informed machine learning framework for predictive maintenance applied to turbomachinery assets”. In: (2021) (cit. on pp. 25, 185).
- [24] Alberto Diez-Olivan et al. “Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0”. In: *Information Fusion* 50 (2019), pp. 92–111 (cit. on p. 25).
- [25] Michael G Kapteyn, David J Knezevic, and Karen Willcox. “Toward predictive digital twins via component-based reduced-order models and interpretable machine learning”. In: *AIAA Scitech 2020 Forum*. 2020, p. 0418 (cit. on pp. 25, 28).

- 
- [26] TG Ritto and FA Rochinha. “Digital twin, physics-based model, and machine learning applied to damage detection in structures”. In: *Mechanical Systems and Signal Processing* 155 (2021), p. 107614 (cit. on pp. 25, 28).
- [27] Min Xia et al. “Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning”. In: *Reliability Engineering & System Safety* 215 (2021), p. 107938 (cit. on pp. 25, 28, 185).
- [28] Rohit Srikonda, Ankur Rastogi, and Haavard Oestensen. “Increasing Facility Uptime Using Machine Learning and Physics-Based Hybrid Analytics in a Dynamic Digital Twin”. In: *Offshore Technology Conference. OnePetro*. 2020 (cit. on p. 25).
- [29] Jeremy Shen et al. “Machine Fault Classification using Hamiltonian Neural Networks”. In: *arXiv preprint arXiv:2301.02243* (2023) (cit. on p. 25).
- [30] Zhonghai Ma et al. “Physics-Informed Machine Learning for Degradation Modeling of an Electro-Hydrostatic Actuator System”. In: *Reliability Engineering & System Safety* 229 (2023), p. 108898 (cit. on pp. 26, 186).
- [31] Karthik Reddy Lyathakula and Fuh-Gwo Yuan. “A Probabilistic Fatigue Life Prediction for Adhesively Bonded Joints via ANNs-based Hybrid Model”. In: *International Journal of Fatigue* (2021), p. 106352 (cit. on pp. 26, 185).
- [32] Yi Gao et al. “Physics-based Deep Learning for Probabilistic Fracture Analysis of Composite Materials”. In: *AIAA Scitech 2020 Forum*. 2020, p. 1860 (cit. on p. 26).
- [33] Urko Leturiondo et al. “Synthetic data generation in hybrid modelling of rolling element bearings”. In: *Insight-Non-Destructive Testing and Condition Monitoring* 57.7 (2015), pp. 395–400 (cit. on p. 26).
- [34] Vincentius Ewald et al. “Perception modelling by invariant representation of deep learning for automated structural diagnostic in aircraft maintenance: A study case using DeepSHM”. In: *Mechanical Systems and Signal Processing* 165 (2022), p. 108153 (cit. on pp. 26, 185).
- [35] J Marcus. “Quantifying Uncertainty in Physics-Informed Variational Autoencoders for Anomaly Detection”. In: *Impact and Opportunities of Artificial Intelligence Techniques in the Steel Industry: Ongoing Applications, Perspectives and Future Trends* 1338 (2021), p. 28 (cit. on p. 26).
- [36] Susu Xu and Hae Young Noh. “Knowledge transfer between buildings for seismic damage diagnosis through adversarial learning”. In: *arXiv preprint arXiv:2002.09513* (2020) (cit. on pp. 26, 28, 38).
- [37] Kari Lovise Ness et al. “Towards a generic physics-based machine learning model for geometry invariant thermal history prediction in additive manufacturing”. In: *Journal of Materials Processing Technology* 302 (2022), p. 117472 (cit. on pp. 26, 28).

- [38] Kai Zhou, Edward Diehl, and Jiong Tang. “Deep convolutional generative adversarial network with semi-supervised learning enabled physics elucidation for extended gear fault diagnosis under data limitations”. In: *Mechanical Systems and Signal Processing* 185 (2023), p. 109772 (cit. on pp. 27, 185).
- [39] Fabian Perez-Sanjines et al. “Fleet-based early fault detection of wind turbine gearboxes using physics-informed deep learning based on cyclic spectral coherence”. In: *Mechanical Systems and Signal Processing* 185 (2023), p. 109760 (cit. on pp. 27, 28).
- [40] Pin Li et al. “A novel scalable method for machine degradation assessment using deep convolutional neural network”. In: *Measurement* 151 (2020), p. 107106 (cit. on pp. 27, 28, 186).
- [41] Yunhan Kim, Kyumin Na, and Byeng D Youn. “A health-adaptive time-scale representation (HTSR) embedded convolutional neural network for gearbox fault diagnostics”. In: *Mechanical Systems and Signal Processing* 167 (2022), p. 108575 (cit. on pp. 27, 185).
- [42] Daisy H Green et al. “Physics-Informed Feature Space Evaluation for Diagnostic Power Monitoring”. In: *IEEE Transactions on Industrial Informatics* (2022) (cit. on p. 27).
- [43] Furkan Guc and YangQuan Chen. “Fault Cause Assignment with Physics Informed Transfer Learning”. In: *IFAC-PapersOnLine* 54.20 (2021), pp. 53–58 (cit. on pp. 27, 185).
- [44] Yuntian Chen et al. “Theory-guided hard constraint projection (HCP): A knowledge-based data-driven scientific machine learning method”. In: *Journal of Computational Physics* 445 (2021), p. 110624 (cit. on p. 28).
- [45] Omer F Eker, Fatih Camci, and Ian K Jennions. “A new hybrid prognostic methodology”. In: *International Journal of Prognostics and Health Management* 10.2 (2019) (cit. on pp. 28, 186).
- [46] Yan Pan et al. “Knowledge-based data augmentation of small samples for oil condition prediction”. In: *Reliability Engineering & System Safety* 217 (2022), p. 108114 (cit. on p. 29).
- [47] Linxia Liao and Felix Köttig. “A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction”. In: *Applied Soft Computing* 44 (2016), pp. 191–199 (cit. on pp. 29, 185).
- [48] Abderrahim Boushaba et al. “Comparative Study between Physics-Informed CNN and PCA in Induction Motor Broken Bars MCSA Detection”. In: *Sensors* 22.23 (2022), p. 9494 (cit. on pp. 29, 185).
- [49] Cheolkyun Jeong et al. “A Physics Model Embedded Hybrid Deep Neural Network for Drillstring Washout Detection”. In: *IADC/SPE International Drilling Conference and Exhibition*. OnePetro. 2020 (cit. on p. 29).

- [50] Minghang Zhao et al. “Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes”. In: *IEEE Transactions on Industrial Electronics* 65.5 (2017), pp. 4290–4300 (cit. on pp. 29, 32).
- [51] Mohammadkazem Sadoughi and Chao Hu. “Physics-based convolutional neural network for fault diagnosis of rolling element bearings”. In: *IEEE Sensors Journal* 19.11 (2019), pp. 4181–4192 (cit. on p. 29).
- [52] Renato G Nascimento et al. “Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis”. In: *Journal of Power Sources* 513 (2021), p. 230526 (cit. on pp. 29, 31, 185).
- [53] Houman Hanachi et al. “Hybrid data-driven physics-based model fusion framework for tool wear prediction”. In: *The International Journal of Advanced Manufacturing Technology* 101.9 (2019), pp. 2861–2872 (cit. on pp. 29, 186).
- [54] Hongyu Sun et al. “Microcrack defect quantification using a focusing high-order SH guided wave EMAT: the physics-informed deep neural network GuwNet”. In: *IEEE Transactions on Industrial Informatics* (2021) (cit. on pp. 30, 185).
- [55] Hongyu Sun et al. “Development of a Physics-Informed Doubly Fed Cross-Residual Deep Neural Network for High-Precision Magnetic Flux Leakage Defect Size Estimation”. In: *IEEE Transactions on Industrial Informatics* 18.3 (2021), pp. 1629–1640 (cit. on pp. 30, 185).
- [56] Sandeep Das et al. “A data-driven physics-informed method for prognosis of infrastructure systems: Theory and application to crack prediction”. In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 6.2 (2020), p. 04020013 (cit. on pp. 30, 38, 186).
- [57] Ramakrishna Tipireddy and Alexandre Tartakovsky. “Physics-informed machine learning method for forecasting and uncertainty quantification of partially observed and unobserved states in power grids”. In: *arXiv preprint arXiv:1806.10990* (2018) (cit. on pp. 30, 32, 38).
- [58] Somdatta Goswami et al. “A physics-informed variational DeepONet for predicting the crack path in brittle materials”. In: *arXiv preprint arXiv:2108.06905* (2021) (cit. on pp. 30, 186).
- [59] Somdatta Goswami et al. “A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials”. In: *Computer Methods in Applied Mechanics and Engineering* 391 (2022), p. 114587 (cit. on pp. 30, 185).
- [60] Khemraj Shukla et al. “Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks”. In: *Journal of Nondestructive Evaluation* 39.3 (2020), pp. 1–20 (cit. on pp. 30, 32, 185).
- [61] Reza Sepasdar, Anuj Karpatne, and Maryam Shakiba. “A data-driven approach to full-field damage and failure pattern prediction in microstructure-dependent composites using deep learning”. In: *arXiv preprint arXiv:2104.04485* (2021) (cit. on p. 30).



- [62] Arinan Dourado and Felipe AC Viana. “Physics-informed neural networks for corrosion-fatigue prognosis”. In: *Proceedings of the Annual Conference of the PHM Society*. Vol. 11. 1. 2019 (cit. on p. 30).
- [63] Felipe AC Viana et al. “Estimating model inadequacy in ordinary differential equations with physics-informed neural networks”. In: *Computers & Structures* 245 (2021), p. 106458 (cit. on pp. 30, 186).
- [64] Renato Giorgiani Nascimento and Felipe AC Viana. “Cumulative damage modeling with recurrent neural networks”. In: *AIAA Journal* 58.12 (2020), pp. 5459–5471 (cit. on pp. 30, 186).
- [65] Renato Giorgiani do Nascimento et al. “Usage-based Lifing of Lithium-Ion Battery with Hybrid Physics-Informed Neural Networks”. In: *AIAA AVIATION 2021 FORUM*. 2021, p. 3046 (cit. on pp. 30, 32, 185).
- [66] Soheil Sadeghi Eshkevari et al. “DynNet: Physics-based neural architecture design for nonlinear structural response modeling and prediction”. In: *Engineering Structures* 229 (2021), p. 111582 (cit. on pp. 30, 32, 185).
- [67] Renato G Nascimento, Kajetan Fricke, and Felipe AC Viana. “A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network”. In: *Engineering Applications of Artificial Intelligence* 96 (2020), p. 103996 (cit. on pp. 30, 32).
- [68] Houpu Yao, Yi Gao, and Yongming Liu. “FEA-Net: A physics-guided data-driven model for efficient mechanical response prediction”. In: *Computer Methods in Applied Mechanics and Engineering* 363 (2020), p. 112892 (cit. on p. 30).
- [69] Jaydeep Karandikar, Tony Schmitz, and Scott Smith. “Physics-guided logistic classification for tool life modeling and process parameter optimization in machining”. In: *Journal of Manufacturing Systems* 59 (2021), pp. 522–534 (cit. on pp. 31, 32, 184, 186).
- [70] Zhilu Lai et al. “Structural identification with physics-informed neural ordinary differential equations”. In: *Journal of Sound and Vibration* 508 (2021), p. 116196 (cit. on pp. 31, 32).
- [71] Marco Star and Kristoffer McKee. “Remaining Useful Life Estimation Using Neural Ordinary Differential Equations”. In: *International Journal of Prognostics and Health Management* 12.2 (2021) (cit. on pp. 31, 32, 185).
- [72] Alexander Bills et al. “Universal battery performance and degradation model for electric aircraft”. In: *arXiv preprint arXiv:2008.01527* (2020) (cit. on pp. 31, 38).
- [73] Ahmed Samir Zamzam and Nicholas D Sidiropoulos. “Physics-aware neural networks for distribution system state estimation”. In: *IEEE Transactions on Power Systems* 35.6 (2020), pp. 4347–4356 (cit. on pp. 31, 32).
- [74] Wenlong Liao et al. “Fault diagnosis of power transformers using graph convolutional network”. In: *CSEE Journal of Power and Energy Systems* 7.2 (2020), pp. 241–249 (cit. on pp. 31, 185).

- [75] Kunjin Chen et al. “Fault location in power distribution systems via deep graph convolutional networks”. In: *IEEE Journal on Selected Areas in Communications* 38.1 (2019), pp. 119–131 (cit. on p. 31).
- [76] Vassilis N Ioannidis, Antonio G Marques, and Georgios B Giannakis. “A recurrent graph neural network for multi-relational data”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8157–8161 (cit. on p. 31).
- [77] Laurent Pagnier and Michael Chertkov. “Physics-informed graphical neural network for parameter & state estimations in power systems”. In: *arXiv preprint arXiv:2102.06349* (2021) (cit. on p. 31).
- [78] Yigit A Yucesan et al. “Adjusting a torsional vibration damper model with physics-informed neural networks”. In: *Mechanical Systems and Signal Processing* 154 (2021), p. 107552 (cit. on p. 31).
- [79] Elisabeth Baseman et al. “Physics-informed machine learning for DRAM error modeling”. In: *2018 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2018, pp. 1–6 (cit. on pp. 31, 33, 37, 185).
- [80] Bingchang Hou et al. “Interpretable online updated weights: Optimized square envelope spectrum for machine condition monitoring and fault diagnosis”. In: *Mechanical Systems and Signal Processing* 169 (2022), p. 108779 (cit. on pp. 31, 185).
- [81] Peter Weiderer, Ana Maria Tomé, and Elmar W Lang. “A NMF-based extraction of physically meaningful components from sensory data of metal casting processes”. In: *Journal of Manufacturing Systems* 54 (2020), pp. 62–73 (cit. on pp. 32, 33).
- [82] Aditya Chakravarty, Siddharth Misra, and Chandra S Rai. “Visualization of hydraulic fracture using physics-informed clustering to process ultrasonic shear waves”. In: *International Journal of Rock Mechanics and Mining Sciences* 137 (2021), p. 104568 (cit. on p. 32).
- [83] Nandar Hlaing et al. “Inspection and maintenance planning for offshore wind structural components: integrating fatigue failure criteria with Bayesian networks and Markov decision processes”. In: *Structure and Infrastructure Engineering* (2022), pp. 1–19 (cit. on pp. 32, 186).
- [84] Yigit A Yucesan and Felipe AC Viana. “A physics-informed neural network for wind turbine main bearing fatigue”. In: *International Journal of Prognostics and Health Management* 11.1 (2020) (cit. on pp. 32, 186).
- [85] Erik Blasch et al. “Data fusion methods for materials awareness”. In: *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX*. Vol. 11423. International Society for Optics and Photonics, 2020, 114230K (cit. on p. 32).

- [86] Wenting Li and Deepjyoti Deka. “Physics-Informed Graph Learning for Robust Fault Location in Distribution Systems”. In: *arXiv e-prints* (2021), arXiv-2107 (cit. on pp. 32, 34, 37, 38, 185).
- [87] Ion Matei et al. “Controlling Draft Interactions Between Quadcopter Unmanned Aerial Vehicles with Physics-aware Modeling”. In: *Journal of Intelligent & Robotic Systems* 101.1 (2021), pp. 1–21 (cit. on p. 32).
- [88] Sergio Cofre-Martel, Enrique Lopez Droguett, and Mohammad Modarres. “Uncovering the Underlying Physics of Degrading System Behavior Through a Deep Neural Network Framework: The Case of Remaining Useful Life Prognosis”. In: *arXiv preprint arXiv:2006.09288* (2020) (cit. on pp. 34, 36, 185).
- [89] Alban Odot, Ryadh Haferssas, and Stéphane Cotin. “DeepPhysics: a physics aware deep learning framework for real-time simulation”. In: *arXiv preprint arXiv:2109.09491* (2021) (cit. on pp. 34, 36).
- [90] Carlos JG Rojas, Marco L Bitterncourt, and José L Boldrini. “Parameter identification for a damage model using a physics informed neural network”. In: *arXiv preprint arXiv:2107.08781* (2021) (cit. on pp. 34, 36, 185).
- [91] Tim von Hahn and Chris K Mechefske. “Knowledge Informed Machine Learning using a Weibull-based Loss Function”. In: *arXiv preprint arXiv:2201.01769* (2022) (cit. on pp. 34, 185, 186).
- [92] Ying Wang et al. “Sensor fault detection of vehicle suspension systems based on transmissibility operators and Neyman–Pearson test”. In: *Reliability Engineering & System Safety* (2023), p. 109067 (cit. on p. 34).
- [93] Yang Xu et al. “Attribute-based structural damage identification by few-shot meta learning with inter-class knowledge transfer”. In: *Structural Health Monitoring* 20.4 (2021), pp. 1494–1517 (cit. on p. 34).
- [94] Brittny Freeman et al. “Physics-informed turbulence intensity infusion: A new hybrid approach for marine current turbine rotor blade fault detection”. In: *Ocean Engineering* 254 (2022), p. 111299 (cit. on pp. 34, 185).
- [95] Harsha Vardhan Tetali, K Supreet Alguri, and Joel B Harley. “Wave Physics Informed Dictionary Learning In One Dimension”. In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2019, pp. 1–6 (cit. on pp. 35, 185).
- [96] Matthew Russell and Peng Wang. “Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring”. In: *Mechanical Systems and Signal Processing* 168 (2022), p. 108709 (cit. on pp. 35, 185).
- [97] Hamed Bolandi et al. “Physics Informed Neural Network for Dynamic Stress Prediction”. In: *arXiv preprint arXiv:2211.16190* (2022) (cit. on pp. 35, 36, 185).

- [98] Sheng Shen et al. “A physics-informed deep learning approach for bearing fault detection”. In: *Engineering Applications of Artificial Intelligence* 103 (2021), p. 104295 (cit. on pp. 35, 36, 38, 185).
- [99] Zhiming Zhang and Chao Sun. “Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating”. In: *Structural Health Monitoring* 20.4 (2021), pp. 1675–1688 (cit. on pp. 35, 36, 185).
- [100] Laura Schröder et al. “Using Transfer Learning to Build Physics-Informed Machine Learning Models for Improved Wind Farm Monitoring”. In: *Energies* 15.2 (2022), p. 558 (cit. on pp. 35, 185).
- [101] Xusheng Yang et al. “Gas path fault diagnosis for gas turbine group based on deep transfer learning”. In: *Measurement* 181 (2021), p. 109631 (cit. on pp. 35, 185).
- [102] Somdatta Goswami et al. “Transfer learning enhanced physics informed neural network for phase-field modeling of fracture”. In: *Theoretical and Applied Fracture Mechanics* 106 (2020), p. 102447 (cit. on p. 35).
- [103] Laura von Rueden et al. “Informed Machine Learning—A Taxonomy and Survey of Integrating Knowledge into Learning Systems”. In: *arXiv preprint arXiv:1903.12394* (2019) (cit. on p. 36).
- [104] Fuh-Gwo Yuan et al. “Machine learning for structural health monitoring: challenges and opportunities”. In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020*. Vol. 11379. International Society for Optics and Photonics. 2020, p. 1137903 (cit. on p. 36).
- [105] Jochen Stiasny, George S Misyris, and Spyros Chatzivasileiadis. “Physics-informed neural networks for non-linear system identification for power system dynamics”. In: *2021 IEEE Madrid PowerTech*. IEEE. 2021, pp. 1–6 (cit. on p. 37).
- [106] Yikang Zhang et al. “Dynet: Dynamic convolution for accelerating convolutional neural networks”. In: *arXiv preprint arXiv:2004.10694* (2020) (cit. on p. 38).
- [107] Xinyu Liu et al. “Tower Masking MIM: A Self-supervised Pretraining Method for Power Line Inspection”. In: *IEEE Transactions on Industrial Informatics* (2023) (cit. on p. 40).
- [108] Qinyuan Luo et al. “Multi-mode non-Gaussian variational autoencoder network with missing sources for anomaly detection of complex electromechanical equipment”. In: *ISA transactions* 134 (2023), pp. 144–158 (cit. on p. 40).
- [109] Tian Wang et al. “Data-driven prognostic method based on self-supervised learning approaches for fault detection”. In: *Journal of Intelligent Manufacturing* 31 (2020), pp. 1611–1619 (cit. on p. 40).
- [110] Yiwei Fu and Feng Xue. “Mad: Self-supervised masked anomaly detection task for multivariate time series”. In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2022, pp. 1–8 (cit. on p. 40).

- [111] Jagath Sri Lal Senanayaka, Kjell G Robbersmyr, et al. “A robust method for detection and classification of permanent magnet synchronous motor faults: Deep autoencoders and data fusion approach”. In: *Journal of Physics: Conference Series*. Vol. 1037. 3. IOP Publishing, 2018, p. 032029 (cit. on p. 40).
- [112] Anass Akrim et al. “Self-Supervised Learning for data scarcity in a fatigue damage prognostic problem”. In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105837 (cit. on pp. 40, 43).
- [113] Yifei Ding et al. “Domain generalization via adversarial out-domain augmentation for remaining useful life prediction of bearings under unseen conditions”. In: *Knowledge-Based Systems* 261 (2023), p. 110199 (cit. on p. 40).
- [114] Zhenjun Yu et al. “Feature extraction based on self-supervised learning for RUL prediction”. In: *Journal of Computing and Information Science in Engineering* (2023), pp. 1–13 (cit. on p. 41).
- [115] Haoren Guo et al. “Masked self-supervision for remaining useful lifetime prediction in machine tools”. In: *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*. IEEE, 2022, pp. 353–358 (cit. on p. 41).
- [116] Juan Xu et al. “Hard Negative Samples Contrastive Learning for Remaining Useful-Life Prediction of Bearings”. In: *Lubricants* 10.5 (2022), p. 102 (cit. on pp. 41, 93, 105, 246, 248).
- [117] Tim Von Hahn and Chris K Mechefske. “Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder”. In: *International Journal of Hydromechatronics* 4.1 (2021), pp. 69–98 (cit. on p. 41).
- [118] Xiang Zhang et al. “Self-supervised contrastive pre-training for time series via time-frequency consistency”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 3988–4003 (cit. on p. 41).
- [119] Shilin Sun, Tianyang Wang, and Fulei Chu. “Adaptive and Self-supervised Condition Monitoring Method for Machinery”. In: *2022 Global Reliability and Prognostics and Health Management (PHM-Yantai)*. IEEE, 2022, pp. 1–7 (cit. on p. 41).
- [120] Zhe Yang et al. “A novel fault detection method for rotating machinery based on self-supervised contrastive representations”. In: *Computers in Industry* 147 (2023), p. 103878 (cit. on p. 41).
- [121] Wenzhen Xie et al. “A unified out-of-distribution detection framework for trustworthy prognostics and health management in renewable energy systems”. In: *Engineering Applications of Artificial Intelligence* 125 (2023), p. 106707 (cit. on p. 41).
- [122] Yifei Ding et al. “Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings”. In: *Reliability Engineering & System Safety* 218 (2022), p. 108126 (cit. on p. 41).

- 
- [123] Yuanfei Wang et al. “Source-free domain adaptation network for rolling bearing fault diagnosis”. In: *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE. 2023, pp. 1691–1696 (cit. on p. 42).
- [124] Zitong Yan and Hongmei Liu. “SMoCo: A powerful and efficient method based on self-supervised learning for fault diagnosis of aero-engine bearing under limited data”. In: *Mathematics* 10.15 (2022), p. 2796 (cit. on p. 42).
- [125] Weikai Lu et al. “Self-supervised domain adaptation for cross-domain fault diagnosis”. In: *International Journal of Intelligent Systems* 37.12 (2022), pp. 10903–10923 (cit. on p. 42).
- [126] Da Peng et al. “Few-label learning for fault diagnosis based on contrastive representations”. In: *2022 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*. IEEE. 2022, pp. 1–5 (cit. on p. 42).
- [127] Meirong Wei et al. “Fault diagnosis of rotating machinery based on improved self-supervised learning method and very few labeled samples”. In: *Sensors* 22.1 (2021), p. 192 (cit. on p. 42).
- [128] Zitong Yan et al. “A Universal Feature Extractor Based on Self-Supervised Pre-Training for Fault Diagnosis of Rotating Machinery under Limited Data”. In: *Aerospace* 10.8 (2023), p. 681 (cit. on p. 42).
- [129] Jichao Zhuang et al. “Health assessment of rotating equipment with unseen conditions using adversarial domain generalization toward self-supervised regularization learning”. In: *IEEE/ASME Transactions on Mechatronics* 27.6 (2022), pp. 4675–4685 (cit. on p. 42).
- [130] Weicheng Wang et al. “One-stage self-supervised momentum contrastive learning network for open-set cross-domain fault diagnosis”. In: *Knowledge-Based Systems* (2023), p. 110692 (cit. on p. 42).
- [131] Tilman Krokotsch, Mirko Knaak, and Clemens Gühmann. “Improving semi-supervised learning for remaining useful lifetime estimation through self-supervision”. In: *arXiv preprint arXiv:2108.08721* (2021) (cit. on pp. 43, 93).
- [132] Yilin Wang et al. “Self-supervised Health Representation Decomposition based on contrast learning”. In: *Reliability Engineering & System Safety* (2023), p. 109455 (cit. on p. 43).
- [133] Ismael Ben-Yelun et al. “Self-learning locally-optimal hypertuning using maximum entropy, and comparison of machine learning approaches for estimating fatigue life in composite materials of the aerospace industry”. In: *Engineering Structures* 283 (2023), p. 115829 (cit. on p. 43).
- [134] Yang Li, Feiyun Xu, and Chi-Guhn Lee. “Self-supervised metalearning generative adversarial network for few-shot fault diagnosis of hoisting system with limited data”. In: *IEEE Transactions on Industrial Informatics* 19.3 (2022), pp. 2474–2484 (cit. on pp. 43, 44).

- 
- [135] Wentao Mao et al. “Self-supervised deep domain-adversarial regression adaptation for online remaining useful life prediction of rolling bearing under unknown working condition”. In: *IEEE Transactions on Industrial Informatics* 19.2 (2022), pp. 1227–1237 (cit. on p. 43).
- [136] Wentao Mao et al. “Self-supervised Deep Tensor Domain-Adversarial Regression Adaptation for Online Remaining Useful Life Prediction Across Machines”. In: *IEEE Transactions on Instrumentation and Measurement* (2023) (cit. on pp. 43, 44).
- [137] Xiao Liu et al. “Self-supervised learning: Generative or contrastive”. In: *IEEE transactions on knowledge and data engineering* 35.1 (2021), pp. 857–876 (cit. on p. 44).
- [138] Mohamed Ragab et al. “Contrastive adversarial domain adaptation for machine remaining useful life prediction”. In: *IEEE Transactions on Industrial Informatics* 17.8 (2020), pp. 5239–5249 (cit. on p. 44).
- [139] Davide Chicco. “Siamese neural networks: An overview”. In: *Artificial neural networks* (2021), pp. 73–94 (cit. on p. 44).
- [140] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018) (cit. on p. 44).
- [141] Benjamin Moseley. *Physics-Informed Machine Learning : from Concepts to Real-World Applications*. 2022 (cit. on p. 45).
- [142] Pu Ren. *Embedding Physics into Deep Learning for Modeling Spatiotemporal Systems*. 2022 (cit. on p. 45).
- [143] Yang Yu. *Data-Physics Driven Reduced Order Homogenization*. 2023 (cit. on p. 45).
- [144] Nan Wang. *Modeling of Water Waves and Sediment Transport Using Physics-Based and Machine Learning Methods*. 2022 (cit. on p. 45).
- [145] Olga Fuks. *Physics Informed Machine Learning and Uncertainty Propagation for Multiphase Transport in Porous Media*. 2020 (cit. on p. 45).
- [146] Gaétan Raynaud. *Study of Physics-Informed Neural Networks to Solve Fluid-Structure Problems for Turbine-Like Phenomena*. 2021 (cit. on p. 45).
- [147] Maryam Aliakbari. *Physics Informed Neural Networks to Solve Forward and Inverse Fluid Flow and Heat Transfer Problems*. 2023 (cit. on p. 45).
- [148] Seyedeh P. Dajkhosh. *Structured Illumination Microscope Image Reconstruction Using Unrolled Physics-Informed Generative Adversarial Network (UPIGAN)*. 2023 (cit. on p. 45).
- [149] Collin Russell. *Applying Physics-Informed Neural Networks to the Solution of Parametric, Nonlinear Heat Conduction Problems*. 2023 (cit. on p. 45).
- [150] Matthew Seman. *Physics Informed Machine Learning Models for PDEs with Applications to Laser Bioeffects*. 2023 (cit. on p. 45).

- 
- [151] Ali Kamali. *Physics-Informed Deep Learning for Quasi-Static Elasticity Imaging: Mechanical Characterization of Heterogeneous Domains in Two and Three Dimensions*. 2023 (cit. on p. 45).
- [152] Phat K. Huynh. *Knowledge Integration in Domain-Informed Machine Learning and Multi-Scale Modeling of Nonlinear Dynamics in Complex Systems*. 2023 (cit. on p. 45).
- [153] Xiaolong He. *Thermodynamically Consistent Physics-Informed Data-Driven Computing and Reduced-Order Modeling of Nonlinear Materials*. 2022 (cit. on p. 45).
- [154] Suraj Pawar. *Physics-Guided Machine Learning for Turbulence Closure and Reduced-Order Modeling*. 2022 (cit. on p. 45).
- [155] Mohammad M. Rahman. *Physics-Aware Deep Learning for Radar-Based Cyber-Physical Human Systems*. 2023 (cit. on p. 45).
- [156] Sifan Wang. *Physics-Informed Machine Learning: Theory, Algorithms and Applications*. 2023 (cit. on p. 45).
- [157] Berkcan Kapusuzoglu. *Physics-Informed Machine Learning for Uncertainty Quantification and Optimization*. 2022 (cit. on p. 45).
- [158] Peter Y. Lu. *Interpretable Physics-Informed Machine Learning Methods for Scientific Modeling and Data Analysis*. 2023 (cit. on p. 45).
- [159] Zhuoran Qiao. *Physics-Informed Neural Approaches for Multiscale Molecular Modeling and Design*. 2023 (cit. on p. 45).
- [160] Erik Wallin. *Semi-Supervised Learning with Self-Supervision for Closed and Open Sets*. 2023 (cit. on pp. 45, 199, 200, 204).
- [161] Ran Tao. *Towards Learning With Limited Supervision: Efficient Few-Shot and Semi-supervised Classification for Vision Tasks*. 2023 (cit. on pp. 45, 200, 204).
- [162] Vipin R. Pillai. *Learning Explainable Models Using Self-Supervised Learning*. 2023 (cit. on pp. 45, 201, 204).
- [163] Jordan R. Stomps. *Gamma Spectroscopy Data Augmentation for Self-supervised Machine Learning Applications to Nuclear Nonproliferation on Measured Data With Limited Ground-Truth*. 2023 (cit. on pp. 46, 200, 205).
- [164] Colorado J. Reed. *Data and Label Efficient Representation Learning*. 2023 (cit. on pp. 46, 201, 205).
- [165] Xiang Deng. *Breaking Down Data Barriers: Knowledge Discovery and Natural Language Interfaces for Heterogeneous Data*. 2023 (cit. on pp. 46, 200, 205).
- [166] Donghuan Wang, Hong Xiao, and Shengqin Huang. *Automatic Defect Recognition and Localization for Aeroengine Turbine Blades Based on Deep Learning*. 2023 (cit. on pp. 46, 190, 199, 200, 204, 205).
- [167] Rebeen A. Hamad. *Sequential Learning and Shared Representation for Sensor-Based Human Activity Recognition*. 2022 (cit. on pp. 46, 199, 201, 204, 205).



- [168] Nikunj Saunshi. *Towards Understanding Self-Supervised Representation Learning*. 2022 (cit. on pp. 46, 199, 203–205).
- [169] Zhangyue Shi et al. *An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing*. Apr. 2023 (cit. on pp. 46, 189, 201, 205).
- [170] Andrew Howard et al. “Searching for mobilenetv3”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1314–1324 (cit. on p. 53).
- [171] Ralph E Blake. “Basic vibration theory”. In: *Shock and vibration handbook 1* (1961), pp. 2–8 (cit. on pp. 53, 206).
- [172] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 55).
- [173] Xiaodong Liang, Mohammad Zawad Ali, and Huaguang Zhang. “Induction motors fault diagnosis using finite element method: A review”. In: *IEEE Transactions on Industry Applications* 56.2 (2019), pp. 1205–1217 (cit. on p. 61).
- [174] Zhiqiang Shen et al. “Partial is better than all: Revisiting fine-tuning strategy for few-shot learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 11. 2021, pp. 9594–9602 (cit. on p. 70).
- [175] Peter M Attia et al. “Closed-loop optimization of fast-charging protocols for batteries with machine learning”. In: *Nature* 578.7795 (2020), pp. 397–402 (cit. on pp. 76–78, 133, 225).
- [176] Joonki Hong et al. “Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning”. In: *Applied energy* 278 (2020), p. 115646 (cit. on pp. 77, 78, 227, 230).
- [177] Kristen A Severson et al. “Data-driven prediction of battery cycle life before capacity degradation”. In: *Nature Energy* 4.5 (2019), pp. 383–391 (cit. on pp. 78, 227).
- [178] Patrick Nectoux et al. “PRONOSTIA: An experimental platform for bearings accelerated degradation tests.” In: *IEEE International Conference on Prognostics and Health Management, PHM’12*. IEEE Catalog Number: CPF12PHM-CDR. 2012, pp. 1–8 (cit. on pp. 90, 133, 238).
- [179] Kaiming He et al. “Identity mappings in deep residual networks”. In: *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer. 2016, pp. 630–645 (cit. on p. 101).
- [180] Albert Gu and Tri Dao. “Mamba: Linear-time sequence modeling with selective state spaces”. In: *arXiv preprint arXiv:2312.00752* (2023) (cit. on pp. 120, 122).
- [181] Chenpaopao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. Published on Chenpaopao blog. Provides a detailed explanation of the Mamba model, which enhances sequence modeling by introducing a selective state space mechanism. June 2024. URL: <http://139.9.1.231/index.php/2024/06/27/mamba/> (cit. on p. 121).

- [182] Rui Xu et al. “A survey on vision mamba: Models, applications and challenges”. In: *arXiv preprint arXiv:2404.18861* (2024) (cit. on p. 120).
- [183] 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 (cit. on p. 127).
- [184] Berend Denkena, Heinrich Klemme, and Tobias H Stiehl. “Multivariate time series data of milling processes with varying tool wear and machine tools”. In: *Data in Brief* 50 (2023), p. 109574 (cit. on pp. 133, 253).
- [185] Marc Moeller, Jochen Blaurock, and Gerhard Ziegmann. “Fatigue dataset for carbon fibre-reinforced polymers under uni- and multiaxial loads with varying biaxiality and proportional stress ratios”. In: *Data in Brief* 45 (2022), p. 108757 (cit. on p. 133).
- [186] Wannas De Groote, Sofie Van Hoecke, and Guillaume Crevecoeur. “Prediction of follower jumps in cam-follower mechanisms: The benefit of using physics-inspired features in recurrent neural networks”. In: *Mechanical Systems and Signal Processing* 166 (2022), p. 108453 (cit. on p. 185).
- [187] Yuejian Chen et al. “Physics-Informed LSTM hyperparameters selection for gearbox fault detection”. In: *Mechanical Systems and Signal Processing* 171 (2022), p. 108907 (cit. on p. 185).
- [188] Shenghan Guo et al. “Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm”. In: *Journal of Manufacturing Systems* 62 (2022), pp. 145–163 (cit. on p. 185).
- [189] Kerman López de Calle et al. “Dynamic condition monitoring method based on dimensionality reduction techniques for data-limited industrial environments”. In: *Computers in Industry* 112 (2019), p. 103114 (cit. on p. 185).
- [190] Asif Khan, Hyunho Hwang, and Heung Soo Kim. “Synthetic Data Augmentation and Deep Learning for the Fault Diagnosis of Rotating Machines”. In: *Mathematics* 9.18 (2021), p. 2336 (cit. on p. 185).
- [191] Susu Xu and Hae Young Noh. “PhyMDAN: Physics-informed knowledge transfer between buildings for seismic damage diagnosis through adversarial learning”. In: *Mechanical Systems and Signal Processing* 151 (2021), p. 107374 (cit. on p. 185).
- [192] Arinan D Dourado and Felipe Viana. “Physics-informed neural networks for bias compensation in corrosion-fatigue”. In: *AIAA Scitech 2020 Forum*. 2020, p. 1149 (cit. on p. 185).
- [193] Chaojie Hu et al. “Damage localization in pressure vessel by guided waves based on convolution neural network approach”. In: *Journal of Pressure Vessel Technology* 142.6 (2020) (cit. on p. 185).
- [194] SZ Feng et al. “A phase field and deep-learning based approach for accurate prediction of structural residual useful life”. In: *Computer Methods in Applied Mechanics and Engineering* 383 (2021), p. 113885 (cit. on p. 185).

- 
- [195] Manuel Arias Chao et al. “Fusing physics-based and deep learning models for prognostics”. In: *Reliability Engineering & System Safety* 217 (2022), p. 107961 (cit. on p. 185).
- [196] Pier Carlo Berri, Matteo DL Dalla Vedova, and Laura Mainini. “Computational framework for real-time diagnostics and prognostics of aircraft actuation systems”. In: *Computers in Industry* 132 (2021), p. 103523 (cit. on p. 185).
- [197] Haroun El Mir and Suresh Perinpanayagam. “Certification Approach for Physics Informed Machine Learning and its Application in Landing Gear Life Assessment”. In: *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–6 (cit. on p. 185).
- [198] Simona Pepe et al. “Neural Ordinary Differential Equations and Recurrent Neural Networks for Predicting the State of Health of Batteries”. In: (2021) (cit. on p. 185).
- [199] Jinjiang Wang et al. “Physics guided neural network for machining tool wear prediction”. In: *Journal of Manufacturing Systems* 57 (2020), pp. 298–310 (cit. on p. 186).
- [200] Arinan De Piemonte Dourado and Felipe Viana. “Ensemble of Hybrid Neural Networks to Compensate for Epistemic Uncertainties: A Case Study in System Prognosis”. In: (2021) (cit. on p. 186).
- [201] Cong Tien Nguyen, Selda Oterkus, and Erkan Oterkus. “A physics-guided machine learning model for two-dimensional structures based on ordinary state-based peridynamics”. In: *Theoretical and Applied Fracture Mechanics* 112 (2021), p. 102872 (cit. on p. 186).
- [202] Simon Pfingstl and Markus Zimmermann. “On integrating prior knowledge into Gaussian processes for prognostic health monitoring”. In: *Mechanical Systems and Signal Processing* 171 (2022), p. 108917 (cit. on p. 186).
- [203] Jie Chen and Yongming Liu. “Fatigue property prediction of additively manufactured Ti-6Al-4V using probabilistic physics-guided learning”. In: *Additive Manufacturing* 39 (2021), p. 101876 (cit. on p. 186).
- [204] Insung Baek and Seoung B. Kim. *Contrastive deep clustering for detecting new defect patterns in wafer bin maps*. Feb. 2024 (cit. on p. 186).
- [205] Minghao Piao and Cheng H. Jin. *CNN and ensemble learning based wafer map failure pattern recognition based on local property based features*. Dec. 2023 (cit. on p. 186).
- [206] Jaegwang Shin and Suan Lee. *Robust and Lightweight Deep Learning Model for Industrial Fault Diagnosis in Low-Quality and Noisy Data*. 2023 (cit. on p. 186).
- [207] Iljoo Jeong et al. *Wafer map failure pattern classification using geometric transformation-invariant convolutional neural network*. 2023 (cit. on p. 186).

- [208] Youngju Kim, Hoyeop Lee, and Chang O. Kim. *A variational autoencoder for a semiconductor fault detection model robust to process drift due to incomplete maintenance*. Feb. 2023 (cit. on p. 186).
- [209] Tobias Schlosser et al. *Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks*. Apr. 2022 (cit. on p. 186).
- [210] Tao Peng et al. *Industrial Product Surface Anomaly Detection with Realistic Synthetic Anomalies Based on Defect Map Prediction*. 2024 (cit. on p. 187).
- [211] Zhiyan Zhong, Hongxin Wang, and Dan Xiang. *Small Defect Detection Based on Local Structure Similarity for Magnetic Tile Surface*. 2023 (cit. on p. 187).
- [212] Zhiqian Jiang et al. *FR-PatchCore: An Industrial Anomaly Detection Method for Improving Generalization*. 2024 (cit. on p. 187).
- [213] Peng Hou et al. *A Bearing Fault Diagnosis Method Based on Dilated Convolution and Multi-Head Self-Attention Mechanism*. 2023 (cit. on p. 187).
- [214] Hüseyin Üzen et al. *Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection*. May 2023 (cit. on p. 187).
- [215] Hanxin Zhang, Qian Sun, and Ke Xu. *A Self-Supervised Model Based on CutPaste-Mix for Ductile Cast Iron Pipe Surface Defect Classification*. 2023 (cit. on pp. 187, 190).
- [216] Jiaying Yang et al. *PreCaCycleGAN: Perceptual Capsule Cyclic Generative Adversarial Network for Industrial Defective Sample Augmentation*. 2023 (cit. on pp. 187, 202).
- [217] Yu He, Xin Wen, and Jing Xu. *A Semi-Supervised Inspection Approach of Textured Surface Defects under Limited Labeled Samples*. 2022 (cit. on p. 187).
- [218] Ihar Volkau et al. *The Impact of a Number of Samples on Unsupervised Feature Extraction, Based on Deep Learning for Detection Defects in Printed Circuit Boards*. 2022 (cit. on p. 187).
- [219] Ji Qiu et al. *Enhancing Anomaly Detection Models for Industrial Applications through SVM-Based False Positive Classification*. 2023 (cit. on p. 187).
- [220] Tianjian Li et al. *Defect Detection Algorithm for Battery Cell Casings Based on Dual-Coordinate Attention and Small Object Loss Feedback*. 2024 (cit. on p. 187).
- [221] Hosameldin O. A. Ahmed and Asoke K. Nandi. *Convolutional-Transformer Model with Long-Range Temporal Dependencies for Bearing Fault Diagnosis Using Vibration Signals*. 2023 (cit. on p. 187).
- [222] Zhenyu Liu et al. *Few-shot defect recognition of metal surfaces via attention-embedding and self-supervised learning*. Dec. 2023 (cit. on p. 187).
- [223] Peng Hou et al. *A Bearing Fault Diagnosis Method Based on Dilated Convolution and Multi-Head Self-Attention Mechanism*. 2023 (cit. on p. 187).

- 
- [224] Shouquan Che et al. *Multiscale Time-Frequency Sparse Transformer Based on Partly Interpretable Method for Bearing Fault Diagnosis*. 2023 (cit. on p. 187).
- [225] Qing Zhang et al. *Visualized Stacked Denoising Auto-Encoder Model for Extracting and Evaluating the State Features of Rolling Bearings*. 2022 (cit. on pp. 187, 190).
- [226] Jianpeng Ma, Chengwei Li, and Guangzhu Zhang. *Rolling Bearing Fault Diagnosis Based on Deep Learning and Autoencoder Information Fusion*. 2022 (cit. on p. 187).
- [227] Xiaoyu Li et al. *A Self-Supervised Bearing Failure Detection Method with STFT and Point-wise CNN Feature Extraction*. Jan. 2023 (cit. on pp. 187, 190).
- [228] Yunji Yang et al. *A limited labeled data bearing fault diagnosis method based on self-supervised learning*. Apr. 2024 (cit. on p. 187).
- [229] Zelin Zhang et al. *Surface defect detection method for discarded mechanical parts under heavy rust coverage*. 2024 (cit. on p. 187).
- [230] Hsu Chia-Yu and Liu Wei-Chen. *Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing*. Mar. 2021 (cit. on p. 187).
- [231] Dejene M. Sime et al. *Deep learning-based automated steel surface defect segmentation: a comparative experimental study*. Jan. 2024 (cit. on p. 187).
- [232] Gongfa Chen et al. *Crack Detection Based on Generative Adversarial Networks and Deep Learning*. Apr. 2022 (cit. on pp. 187, 199, 202, 204).
- [233] Zian Yu et al. *Semi-supervised auxiliary learning for surface defect detection and segmentation of injection-molded products from small image datasets*. Apr. 2024 (cit. on p. 187).
- [234] Cheng Wei et al. *Multi-stage unsupervised fabric defect detection based on DCGAN*. Dec. 2023 (cit. on p. 187).
- [235] Haobin Wen, Long Zhang, and Jyoti K. Sinha. *Adaptive Band Extraction Based on Low Rank Approximated Nonnegative Tucker Decomposition for Anti-Friction Bearing Faults Diagnosis Using Measured Vibration Data*. 2022 (cit. on p. 187).
- [236] Jiachen Kuang et al. *Domain-Adaptive Prototype-Recalibrated Network with Transductive Learning Paradigm for Intelligent Fault Diagnosis under Various Limited Data Conditions*. 2022 (cit. on p. 187).
- [237] Tengyi Peng et al. *Fault Feature Extractor Based on Bootstrap Your Own Latent and Data Augmentation Algorithm for Unlabeled Vibration Signals*. 2022 (cit. on p. 187).
- [238] Wei Zhang et al. *Federated learning for machinery fault diagnosis with dynamic validation and self-supervision*. Feb. 2021 (cit. on pp. 187, 189).
- [239] Xiaobo Liu, Hantao Guo, and Yibing Liu. *One-Shot Fault Diagnosis of Wind Turbines Based on Meta-Analogical Momentum Contrast Learning*. 2022 (cit. on p. 187).

- [240] Youming Wang, Gongqing Cao, and Jiali Han. *A Combination of Dilated Self-Attention Capsule Networks and Bidirectional Long- and Short-Term Memory Networks for Vibration Signal Denoising*. 2022 (cit. on pp. 187, 190).
- [241] Phong B. Dao. *On Cointegration Analysis for Condition Monitoring and Fault Detection of Wind Turbines Using SCADA Data*. 2023 (cit. on p. 188).
- [242] David Gonzalez-Jimenez et al. *Machine Learning-Based Fault Detection and Diagnosis of Faulty Power Connections of Induction Machines*. 2021 (cit. on p. 188).
- [243] Zhengyang Fan, Wanru Li, and Chang Kuo-Chu. *A Two-Stage Attention-Based Hierarchical Transformer for Turbofan Engine Remaining Useful Life Prediction*. 2024 (cit. on pp. 188, 189).
- [244] Hongqing Zheng et al. *Remaining Useful Life Prediction of a Planetary Gearbox Based on Meta Representation Learning and Adaptive Fractional Generalized Pareto Motion*. 2024 (cit. on pp. 188, 189).
- [245] Shulian Xie et al. *Data-Driven Predictive Maintenance Policy Based on Dynamic Probability Distribution Prediction of Remaining Useful Life*. 2023 (cit. on p. 188).
- [246] Jose Maestro-Prieto et al. *Semi-supervised diagnosis of wind-turbine gearbox misalignment and imbalance faults*. Mar. 2024 (cit. on p. 188).
- [247] Yilin Wang et al. *Unsupervised Learning with Generative Adversarial Network for Automatic Tire Defect Detection from X-ray Images*. 2021 (cit. on p. 188).
- [248] Moinak Pyne, Benjamin J. Yurkovich, and Stephen Yurkovich. *Synthetic Battery Data Generation and Validation for Capacity Estimation*. 2023 (cit. on p. 188).
- [249] Sajjad Ahmad, Ahmad Zahoor, and Jong-Myon Kim. *A Centrifugal Pump Fault Diagnosis Framework Based on Supervised Contrastive Learning*. 2022 (cit. on p. 188).
- [250] Abdellatif Elmouatamid et al. *Air Conditioning Systems Fault Detection and Diagnosis-Based Sensing and Data-Driven Approaches*. 2023 (cit. on p. 188).
- [251] Yong Shi et al. *Fault Detection in Nuclear Power Plants using Deep Learning based Image Classification with Imaged Time-series Data*. Feb. 2022 (cit. on p. 188).
- [252] Nick Torenvliet and John Zelek. *Evaluating Diffusion Models for the Automation of Ultrasonic Nondestructive Evaluation Data Analysis*. 2024 (cit. on p. 188).
- [253] Chaofan Zhou et al. *A graph-based two-stage classification network for mobile screen defect inspection*. Feb. 2023 (cit. on pp. 188, 190).
- [254] Nhat-Quang Dang et al. *Supervised Contrastive Learning for Fault Diagnosis Based on Phase-Resolved Partial Discharge in Gas-Insulated Switchgear*. 2024 (cit. on p. 188).
- [255] Li Qi et al. *Two-view LSTM variational auto-encoder for fault detection and diagnosis in multivariable manufacturing processes*. Oct. 2023 (cit. on p. 188).

- 
- [256] Wentao Mao et al. *Self-Supervised Deep Tensor Domain-Adversarial Regression Adaptation for Online Remaining Useful Life Prediction Across Machines*. 2023 (cit. on p. 189).
- [257] Hossein Hassani et al. *Generative Adversarial Network-Based Scheme for Diagnosing Faults in Cyber-Physical Power Systems*. 2021 (cit. on p. 189).
- [258] Imene Mitiche et al. *Data-Driven Anomaly Detection in High-Voltage Transformer Bushings with LSTM Auto-Encoder*. 2021 (cit. on p. 189).
- [259] Dou Jianming et al. *An unsupervised online monitoring method for tool wear using a sparse auto-encoder*. Jan. 2020 (cit. on p. 189).
- [260] Wei Sun et al. *Markov Transition Field Enhanced Deep Domain Adaptation Network for Milling Tool Condition Monitoring*. 2022 (cit. on p. 189).
- [261] Ahmed Elsheikh et al. *Failure time prediction using adaptive logical analysis of survival curves and multiple machining signals*. Feb. 2020 (cit. on p. 189).
- [262] Iulian Lupea and Mihaela Lupea. *Detecting Helical Gearbox Defects from Raw Vibration Signal Using Convolutional Neural Networks*. 2023 (cit. on p. 189).
- [263] Elisavet Karapalidou et al. *Implementation of a Sequence-to-Sequence Stacked Sparse Long Short-Term Memory Autoencoder for Anomaly Detection on Multivariate Time-series Data of Industrial Blower Ball Bearing Units*. 2023 (cit. on p. 189).
- [264] Peng Gao et al. *Method for monitoring and controlling penetration of complex groove welding based on online multi-modal data*. Mar. 2024 (cit. on p. 190).
- [265] Hanshen Chen, Minghai Yao, and Qinlong Gu. *Pothole detection using location-aware convolutional neural networks*. Apr. 2020 (cit. on p. 190).
- [266] Yao Xu et al. *Consistency-based Semi-supervised Learning Framework for Power Line Insulators Detection*. Apr. 2022 (cit. on p. 190).
- [267] Rong Luo et al. *RBD-Net: robust breakage detection algorithm for industrial leather*. Aug. 2023 (cit. on p. 190).
- [268] J. W. Chen, W. J. Lin, and C. L. Hung. *A Novel Denoising Autoencoder Method for Surface Defect Detection of Screw Products*. Oct. 2023 (cit. on p. 190).
- [269] Ye Zheng and Li Cui. *Defect detection on new samples with siamese defect-aware attention network*. Feb. 2023 (cit. on p. 190).
- [270] Georgios Chliveros, Stylianos V. Kontomaris, and Apostolos Letsios. *Automatic Identification of Corrosion in Marine Vessels Using Decision-Tree Imaging Hierarchies*. 2023 (cit. on p. 190).
- [271] Liang Ma and Fanwu Meng. *Anomaly Detection in the Production Process of Stamping Progressive Dies Using the Shape- and Size-Adaptive Descriptors*. 2023 (cit. on p. 190).
- [272] Shigeki Yumoto et al. *Anomaly detection from images in pipes using GAN*. Dec. 2023 (cit. on p. 190).

- [273] Hirunima Jayasekara et al. *Detecting Anomalous Solder Joints in Multi-sliced PCB X-ray Images: A Deep Learning Based Approach*. May 2023 (cit. on p. 190).
- [274] Suliman M. Hafiz et al. *Civil Infrastructure Damage and Corrosion Detection: An Application of Machine Learning*. 2022 (cit. on p. 190).
- [275] Sebastian Larsen and Paul A. Hooper. *Deep semi-supervised learning of dynamics for anomaly detection in laser powder bed fusion*. Feb. 2022 (cit. on p. 190).
- [276] Filippo Galli et al. *Self-supervised pre-training of CNNs for flatness defect classification in the steelworks industry*. Mar. 2020 (cit. on p. 190).
- [277] Rahul Rai and Chandan K Sahu. “Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus”. In: *IEEE Access* 8 (2020), pp. 71050–71073 (cit. on p. 191).
- [278] Jared Willard et al. “Integrating physics-based modeling with machine learning: A survey”. In: *arXiv preprint arXiv:2003.04919* 1.1 (2020), pp. 1–34 (cit. on p. 191).
- [279] Sung Wook Kim et al. “Knowledge Integration into deep learning in dynamical systems: an overview and taxonomy”. In: *Journal of Mechanical Science and Technology* (2021), pp. 1–12 (cit. on p. 191).
- [280] Elias Jan Hagedorfer. “Knowledge Incorporation for Machine Learning in Condition Monitoring: A Survey”. In: *2021 International Symposium on Electrical, Electronics and Information Engineering*. 2021, pp. 230–240 (cit. on p. 191).
- [281] Jianjing Zhang and Robert X Gao. “Deep Learning-Driven Data Curation and Model Interpretation for Smart Manufacturing”. In: *Chinese Journal of Mechanical Engineering* 34.1 (2021), pp. 1–21 (cit. on p. 192).
- [282] Yanwen Xu et al. “Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges”. In: *Reliability Engineering & System Safety* (2022), p. 108900 (cit. on p. 192).
- [283] Adam Thelen et al. “A comprehensive review of digital twin—part 1: modeling and twinning enabling technologies”. In: *Structural and Multidisciplinary Optimization* 65.12 (2022), p. 354 (cit. on p. 192).
- [284] Adam Thelen et al. “A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives”. In: *Structural and multidisciplinary optimization* 66.1 (2023), p. 1 (cit. on p. 192).
- [285] Vipul Mann. *Domain-Informed Language Models for Process Systems Engineering*. English. 2024 (cit. on p. 193).
- [286] Sifan Wang. *Physics-Informed Machine Learning: Theory, Algorithms and Applications*. English. 2023 (cit. on pp. 193, 195).
- [287] Bahador Bahmani. *Geometry-Informed Data-Driven Mechanics*. English. 2024 (cit. on p. 193).



- [288] Shafi A. S. Romeo. *Nonlinear System Discovery and Machine Learning for Dynamical Systems*. English. 2023 (cit. on p. 193).
- [289] Collin Russell. *Applying Physics-Informed Neural Networks to the Solution of Parametric, Nonlinear Heat Conduction Problems*. English. 2023 (cit. on p. 193).
- [290] Phat K. Huynh. *Knowledge Integration in Domain-Informed Machine Learning and Multi-Scale Modeling of Nonlinear Dynamics in Complex Systems*. English. 2023 (cit. on p. 193).
- [291] Aref Ghaderi. *Physics-Informed Data-Driven Models for Inelastic, Aging, Failure Behavior of Crosslinked Polymers*. English. 2023 (cit. on p. 194).
- [292] Tasmiah Haque. *Physics Infused LSTM Network for Track Association Based on Marine Vessel Automatic Identification System Data*. English. 2023 (cit. on p. 194).
- [293] Mehrad Ansari. *Applications of Physics-Informed Machine Learning in Chemical Engineering*. English. 2023 (cit. on p. 194).
- [294] Thomas Stone. *Physics Informed Neural Networks for Ultrasonic Guided Wave Propagation in Solid Media*. English. 2023 (cit. on p. 194).
- [295] Walter Alvarado. *Understanding Chromatin Remodeling Through Physics-Based Machine Learning Approaches*. English. 2023 (cit. on p. 194).
- [296] Seyedeh P. Dajkhosh. *Structured Illumination Microscope Image Reconstruction Using Unrolled Physics-Informed Generative Adversarial Network (UPIGAN)*. English. 2023 (cit. on p. 194).
- [297] Peter Y. Lu. *Interpretable Physics-Informed Machine Learning Methods for Scientific Modeling and Data Analysis*. English. 2023 (cit. on p. 194).
- [298] Yang Yu. *Data-Physics Driven Reduced Order Homogenization*. English. 2023 (cit. on p. 195).
- [299] Zhuoran Qiao. *Physics-Informed Neural Approaches for Multiscale Molecular Modeling and Design*. English. 2023 (cit. on p. 195).
- [300] Konstantinos Prantikos. *Physics-Informed Neural Network Solution of Point Kinetics Equations For Pur-1 Digital Twin*. English. 2022 (cit. on p. 195).
- [301] Ali Kamali. *Physics-Informed Deep Learning for Quasi-Static Elasticity Imaging: Mechanical Characterization of Heterogeneous Domains in Two and Three Dimensions*. English. 2023 (cit. on p. 195).
- [302] Mohammad M. Rahman. *Physics-Aware Deep Learning for Radar-Based Cyber-Physical Human Systems*. English. 2023 (cit. on p. 195).
- [303] Swapnil S. Saha. *Physics-Aware Tiny Machine Learning*. English. 2023 (cit. on p. 195).
- [304] Maryam Aliakbari. *Physics Informed Neural Networks to Solve Forward and Inverse Fluid Flow and Heat Transfer Problems*. English. 2023 (cit. on p. 196).

- 
- [305] Matthew Seman. *Physics Informed Machine Learning Models for PDEs with Applications to Laser Bioeffects*. English. 2023 (cit. on p. 196).
- [306] Benjamin Moseley. *Physics-Informed Machine Learning : from Concepts to Real-World Applications*. English. 2022 (cit. on p. 196).
- [307] Gaétan Raynaud. *Study of Physics-Informed Neural Networks to Solve Fluid-Structure Problems for Turbine-Like Phenomena*. French. 2021 (cit. on p. 196).
- [308] Suraj Pawar. *Physics-Guided Machine Learning for Turbulence Closure and Reduced-Order Modeling*. English. 2022 (cit. on p. 196).
- [309] Shaan Desai. *Physics-Informed Neural Networks for Data-Efficient Learning*. English. 2021 (cit. on p. 196).
- [310] Kristina Monakhova. *Physics-Informed Machine Learning for Computational Imaging*. English. 2022 (cit. on p. 196).
- [311] Xiaolong He. *Thermodynamically Consistent Physics-Informed Data-Driven Computing and Reduced-Order Modeling of Nonlinear Materials*. English. 2022 (cit. on p. 197).
- [312] Nan Wang. *Modeling of Water Waves and Sediment Transport Using Physics-Based and Machine Learning Methods*. English. 2022 (cit. on p. 197).
- [313] Pu Ren. *Embedding Physics into Deep Learning for Modeling Spatiotemporal Systems*. English. 2022 (cit. on p. 197).
- [314] Zhao Zhang. *Freeway Traffic Flow Modeling and Forecasting Using Physics-Guided Machine Learning*. English. 2022 (cit. on p. 197).
- [315] Xianghao Kong. *Physics-Informed Machine Learning Models for Power Transmission Systems*. English. 2022 (cit. on p. 197).
- [316] Thelma A. Ihunde. *The Application of Physics Informed Neural Networks to Compositional Modeling*. English. 2022 (cit. on p. 197).
- [317] Berkcan Kapusuzoglu. *Physics-Informed Machine Learning for Uncertainty Quantification and Optimization*. English. 2022 (cit. on p. 198).
- [318] Abantika Ghosh. *Physics-Informed Machine Learning for Optical Metamaterials*. English. 2022 (cit. on p. 198).
- [319] Jr. Behl Mark V. *Development of Reduced Order Models Using Reservoir Simulation and Physics Informed Machine Learning Techniques*. English. 2020 (cit. on p. 198).
- [320] Sakib A. Zargar. *Pushing the Boundaries of Sparse Sensor Array Based Structural Health Monitoring Via Physics-Informed Deep Learning*. English. 2022 (cit. on p. 198).
- [321] Zhihao Chen et al. "Physics-informed generative neural network: an application to troposphere temperature prediction". In: *Environmental Research Letters* 16.6 (2021), p. 065003 (cit. on p. 198).
- [322] Olga Fuks. *Physics Informed Machine Learning and Uncertainty Propagation for Multiphase Transport in Porous Media*. English. 2020 (cit. on p. 198).

- 
- [323] Soroush Abbasi Koohpayegani. *Self-Supervised Learning by Compressing Representations for Lightweight Models*. 2022 (cit. on pp. 199, 203, 204).
- [324] Donghyun Kim. *Learning Generalizable and Transferable Representations across Domains and Modalities*. 2022 (cit. on pp. 199, 203, 204).
- [325] Abubakar Siddique. *Downstream Task Self-Supervised Learning for Object Recognition and Tracking*. 2023 (cit. on pp. 199, 202, 204).
- [326] Md M. Rahman. *Toward Developing Multi-modal Deep Simultaneous Learning: Theory and Applications*. 2024 (cit. on pp. 199, 200, 204).
- [327] Viviana F. Magalhães. *Self-Supervised Learning Techniques for Monitoring Industrial Spaces*. 2023 (cit. on pp. 199, 200, 204).
- [328] Honglie Chen. *Learning with Multimodal Self-Supervision*. 2021 (cit. on pp. 199, 202, 204).
- [329] Prashna K. Gyawali. *Learning with Limited Labeled Data in Biomedical Domain by Disentanglement and Semi-Supervised Learning*. 2021 (cit. on pp. 199, 203, 204).
- [330] Andrew O'Brien. *Dynamic Causality: Sparse Symbolic Regression as a Tool to Learn Causal Dynamic Structural Equations With Applications to Counterfactuals*. 2024 (cit. on pp. 199, 204).
- [331] Ashraful Islam. *Learning with Limited Labeled Data in Biomedical Domain by Disentanglement and Semi-Supervised Learning*. 2022 (cit. on pp. 199, 203, 204).
- [332] Abu Z. M. Faridee. *Building Robust Human Activity Recognition Models from Unlabeled Data*. 2022 (cit. on pp. 199, 203, 204).
- [333] Longlong Jing. *Learning Deep Visual Features from Limited Labeled Data*. 2021 (cit. on pp. 199, 203, 204).
- [334] Juntao Tan. *Explainable AI for Human and Science*. 2024 (cit. on p. 199).
- [335] Boyang Lyu. *Domain Generalization via Representation Learning*. 2024 (cit. on p. 200).
- [336] Dimitrios Koutrintzes. *Knowledge Transfer in Human-Artificial Intelligence Collaboration*. 2023 (cit. on p. 200).
- [337] Jinqi Cheng. *Learning Edge Detectors From Natural Images Without Human Annotations*. 2021 (cit. on p. 201).
- [338] Teodor Fredriksson. *Opportunities, Challenges and Solutions for Automatic Labeling of Data Using Machine Learning*. 2023 (cit. on p. 201).
- [339] Arjon Das. *ViewMix: Augmentation for Robust Representation in Self-Supervised Learning*. 2023 (cit. on p. 201).
- [340] Yayong Li. *Handling Sparse and Noisy Labels in Deep Graph Learning*. 2022 (cit. on p. 201).

- 
- [341] Oleh Rybkin. *Learning to Act From Diverse Data Sources via World Models*. 2023 (cit. on p. 201).
- [342] Deen D. Mohan. *Improving Representation Learning for Diverse Modalities and Distributions with Metric Learning*. 2023 (cit. on p. 201).
- [343] Ziniu Hu. *Make Knowledge Computable: Towards Differentiable Neural-Symbolic AI*. 2023 (cit. on p. 202).
- [344] Glenn Dawson. *A General Model for Noisy Labels in Machine Learning*. 2023 (cit. on p. 202).
- [345] Shichao Xu. *Learning from Limited and Imperfect Data in Cyber-Physical System*. 2023 (cit. on p. 202).
- [346] Cat P. Le. *Task Affinity and its Applications in Machine Learning*. 2023 (cit. on p. 202).
- [347] Rajshekhar Das. *Learning in the Wild with Limited Supervision*. 2023 (cit. on p. 202).
- [348] Tomas Jakab. *Self-Supervised Learning of Structural Representations of Visual Objects*. 2021 (cit. on p. 202).
- [349] Damian Mrowca. *Curiosity-Driven Learning for Physically Grounded Autonomous Agents*. 2021 (cit. on p. 202).
- [350] Mengye Ren. *Open-World Machine Learning with Limited Labeled Data*. 2022 (cit. on p. 203).
- [351] Vojtěch Šalanský. *Učení a Vnímání Robotu v Částečně Neznámém Prostoru*. 2022 (cit. on p. 203).
- [352] Hanbin Hu. *Machine Learning Techniques for Rare Failure Detection in Analog and Mixed-Signal Verification and Test*. 2021 (cit. on p. 203).
- [353] Fukai Li et al. “A new damage-mechanics-based model for rolling contact fatigue analysis of cylindrical roller bearing”. In: *Tribology International* 120 (2018), pp. 105–114 (cit. on p. 206).
- [354] Siddique Khan et al. “Vibration and Acoustic Method for Detection of Cracks in Bearings: A Critical Review”. In: *Advances in Engineering Design* (2021), pp. 221–229 (cit. on p. 206).
- [355] Sergey Porotsky and Zigmund Bluvband. “Remaining useful life estimation for systems with non-trendability behaviour”. In: *2012 IEEE Conference on Prognostics and Health Management*. IEEE, 2012, pp. 1–6 (cit. on p. 206).
- [356] Jing Liu and Yimin Shao. “A new dynamic model for vibration analysis of a ball bearing due to a localized surface defect considering edge topographies”. In: *Non-linear Dynamics* 79.2 (2015), pp. 1329–1351 (cit. on p. 207).

- [357] Nima Rezazadeh, Alessandro De Luca, and Donato Perfetto. “Unbalanced, cracked, and misaligned rotating machines: A comparison between classification procedures throughout the steady-state operation”. In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 44.10 (2022), pp. 1–14 (cit. on p. 212).
- [358] Sheng Fu et al. “Rolling bearing diagnosing method based on time domain analysis and adaptive fuzzy-means clustering”. In: *Shock and Vibration* 2016 (2016) (cit. on p. 214).
- [359] Sheng Guo et al. “A novel fault diagnosis method for rotating machinery based on a convolutional neural network”. In: *Sensors* 18.5 (2018), p. 1429 (cit. on pp. 219, 220).
- [360] Minghang Zhao et al. “Deep residual shrinkage networks for fault diagnosis”. In: *IEEE Transactions on Industrial Informatics* 16.7 (2019), pp. 4681–4690 (cit. on pp. 219–221).
- [361] Yan Xue, Dongyang Dou, and Jianguo Yang. “Multi-Fault Diagnosis of Rotating Machinery Based on Deep Convolution Neural Network and Support Vector Machine”. In: *Measurement* 156 (2020), p. 107571 (cit. on pp. 219, 221).
- [362] Yanhua Pang et al. “Spatio-temporal fusion neural network for multi-class fault diagnosis of wind turbines based on SCADA data”. In: *Renewable Energy* 161 (2020), pp. 510–524 (cit. on pp. 219, 221).
- [363] Yordanos Dametw Mamuya et al. “Application of Machine Learning for Fault Classification and Location in a Radial Distribution Grid”. In: *Applied Sciences* 10.14 (2020), p. 4965 (cit. on pp. 220, 223).
- [364] Abdelrahman Said et al. “Deep Learning-Based Fault Classification and Location for Underground Power Cable of Nuclear Facilities”. In: *IEEE Access* 10 (2022), pp. 70126–70142 (cit. on pp. 220, 222).
- [365] Soufiane Belagoune et al. “Deep Learning through LSTM Classification and Regression for Transmission Line Fault Detection, Diagnosis, and Location in Large-Scale Multi-Machine Power Systems”. In: *Measurement* 177 (2021), p. 109330 (cit. on pp. 220, 222).
- [366] DENG Weikun et al. “Physics-informed machine learning in prognostics and health management: State of the art and challenges”. In: *Applied Mathematical Modelling* 124 (2023), pp. 325–352 (cit. on p. 225).
- [367] Sara Kohtz et al. “Physics-informed machine learning model for battery state of health prognostics using partial charging segments”. In: *Mechanical Systems and Signal Processing* 172 (2022), p. 109002 (cit. on p. 227).
- [368] Benvolence Chinomona et al. “Long short-term memory approach to estimate battery remaining useful life using partial data”. In: *Ieee Access* 8 (2020), pp. 165419–165431 (cit. on p. 230).

- [369] Niloofar Kamyab, John W Weidner, and Ralph E White. “Mixed mode growth model for the solid electrolyte interface (SEI)”. In: *Journal of The Electrochemical Society* 166.2 (2019), A334 (cit. on p. 233).
- [370] Fan Xu et al. “Constructing a health indicator for roller bearings by using a stacked auto-encoder with an exponential function to eliminate concussion”. In: *Applied Soft Computing* 89 (2020), p. 106119 (cit. on p. 248).
- [371] Yanan Chen et al. “Degradation-trend-dependent remaining useful life prediction for bearing with BiLSTM and attention mechanism”. In: *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE. 2021, pp. 1177–1182 (cit. on p. 248).
- [372] Ido Ben-Shaul et al. “Reverse Engineering Self-Supervised Learning”. In: *arXiv preprint arXiv:2305.15614* (2023) (cit. on p. 251).
- [373] Luca Tonin, Felix Christian Bauer, and José del R Millán. “The role of the control framework for continuous teleoperation of a brain–machine interface-driven mobile robot”. In: *IEEE Transactions on Robotics* 36.1 (2019), pp. 78–91 (cit. on p. 256).
- [374] Hongyao Shen, Lingnan Pan, and Jun Qian. “Research on large-scale additive manufacturing based on multi-robot collaboration technology”. In: *Additive Manufacturing* 30 (2019), p. 100906 (cit. on p. 256).

# Appendix of Introduction

---

## A.1 “Ill-posed” problematic nature

We approach the modelling of PHM from the perspective of inverse modelling, as represented in Eq. (A.1). Here,  $\mathcal{A}_{inv} : \mathcal{X} \rightarrow \mathcal{Y}$  is an operator representing the inverse PHM model,  $p_{inv} \in X$  is the system state or parameter to be estimated (e.g., RUL, degradation states, faults), and  $\mathcal{G}_{inv} \in \mathcal{Y}$  is the observed data (e.g., sensor measurements, operational parameters), where  $\epsilon_{inv}(t)$  represents the compound noise and model uncertainties.

$$\mathcal{G}_{inv}(t) = h(\mathcal{A}_{inv}(p_{inv}, t)) + \epsilon_{inv}(t) \quad (\text{A.1})$$

In PHM, “ill-posed” occurs when Hadamard’s conditions for well-posedness are violated. These conditions require that a solution to a problem should exist, be unique, and depend continuously on the input data (stability):

- The **existence condition** is violated when the observed data falls outside the range of the PHM model. This violation often results from model limitations that fail to capture all potential system behaviours within “Sparse and noisy data.” “Scarce knowledge” exacerbates this issue by affecting accurately define  $\mathcal{A}_{inv}$ . Unknown failure mechanisms and complex system interactions mean that  $\mathcal{A}_{inv}$  is often an approximation, leading to model uncertainties. As a result, an exact solution  $p_{inv}$  that satisfies Eq. (A.1) and perfectly aligns with the observations  $\mathcal{G}_{inv}(t)$  may not exist.
- The **uniqueness condition** is compromised when multiple system states can produce identical observations. Uniquely determining the system state is difficult with insufficient information, resulting in an underdetermined system where the number of unknowns exceeds the number of equations. There are infinite sets of possible solutions that equally fit the available data. Furthermore, model simplifications and intrinsic limitations in sensing capabilities can result in non-injective mappings in  $\mathcal{A}_{inv}(p_{inv}, t)$ , as it introduces additional ambiguity in the mapping between system states and observations, potentially increasing the set of system states that could produce the same observed data.

- The **stability condition** is breached when  $\mathcal{A}_{inv}$  is highly sensitive to input perturbations. This manifests as substantial variations in the estimated system state due to measurement errors or noise. Data sparsity aggravates this issue by amplifying the influence of individual singular measurements. Additionally, the nonlinear nature of many PHM problems leads to situations where small changes in input can cause disproportionately large variations in output, further destabilizing the model. The model uncertainties represented by  $\Delta\mathcal{A}_{inv}$  add another layer of instability by propagating the small errors in the system.



APPENDIX B

# Appendix: Litterature review

---

## B.1 Supplementary notes on PIML and SSL in PHM

### B.1.1 Statistical analysis of PIML in PHM

The observations of Table. B.1 is summarized as follows:

- Table. B.1 details the training and testing metrics used in PHM tasks such as condition monitoring, fault diagnostics, and Remaining Useful Life (RUL) prediction for evaluating PIML models. Metrics like Mean Square Error (MSE), Mean Absolute Error (MAE), Precision, Recall, and F1-score are utilized, with MSE, MAE, and RMSE being the most common at 35.7%, 20.0%, and 10% respectively.
- Paper [69] applies binary cross-entropy, typically used in classifications, to predict degradation levels, transforming them into a classifiable format.
- The choice of embedded knowledge in PIML models often appears subjective and closely linked to the types of monitoring signals used, such as the correlation between strain signals and structural damage or between temperature and fatigue. This suggests a practical approach to embedding knowledge by leveraging available monitoring data.
- Table. B.1 also specifies the monitoring signals used in each task, predominantly one-dimensional time-series data like displacement, voltage, and temperature, with a minor focus on two-dimensional image signals.
- Predominantly, PIML applications target structural health monitoring using signals mainly derived from vibration and stress. The diversity of metrics and signals across PIML studies underscores the complex, multidimensional nature of PHM tasks, emphasizing the need for tailored approaches that align with specific application needs and system characteristics. This diversity serves as a guide for selecting suitable evaluation measures and sensor inputs in PHM applications.

B.1. Supplementary notes on PIML and SSL in PHM

**Table B.1:** Summary of the training metrics, testing metrics, and monitoring signals for PIML according to PHM tasks.

Ref.	Train metric	Test metric	Tasks	Signals
[186], [187], [96], [48]	MSE	MSE	CM	Displacement, voltage, vibration, currents, time measurements
[188], [100], [74], [86], [41], [80], [16]	MAE, Cross-entropy loss, Customized design loss	Precision, recall, F1-score, accuracy, macro F1, G-mean, RMSE	CM & FD	Vibration, acoustic, image, temperature, power, pressure, air flow
[43], [27], [38], [101], [189], [190], [66], [191], [94], [55], [192], [59], [90], [54], [98], [95], [34], [193], [60], [99], [79], [97]	MAE, MSE, Binary cross-entropy, Customized loss, Similarity distance, Kernel norm	Confusion matrix, recall, precision, f-measure, Pearson correlation coefficients test, Relative percentage error, Categorical cross-entropy, MAE, Customized metric	FD	Temperature, pressure, fuel coefficient, acoustic signal, strain, torque, acoustic emission, magnetic flux leakage image, far-field loads, stress ratio, corrosivity index, ultrasonic signal, guided wave signal, mode shapes signal, wave data, stress
[194], [20], [47], [195], [23], [88], [196], [71], [197], [31], [65], [52], [198], [91], [18]	Cross-entropy loss, MAE, RMSE	$\alpha_\lambda$ distribution accuracy, One $\sigma$ tolerance interval, RMSE, MAE, R2, Relative error rate	RP	Phase field images, vibration, voltage, current, temperature, capacities, stress or strain

Continued on next page

**Table B.1 – continued from previous page**

Ref.	Train metric	Test metric	Tasks	Signals
[45], [199], [91], [40], [200], [58], [201], [69], [63], [202], [56], [84], [53], [64], [83], [30], [203]	MAE, MSE, Binary cross-entropy, Customized loss, NMSE, RMSE, Negative log likelihood	MAE, RMSE, F1-score, MAPE, NMAE, Sensitivity analysis, absolute error variance, Discretization error	DP	Forces, vibrations, acoustic signal, cutting speed, temperature, stress or image, spindle motor current, far-field stress, viscosity, wind speed, rise time, displacement, stress, crack length, pressure

CM: Condition monitoring, FD: Fault diagnostic, RP: RUL prediction, DP: Degradation prediction.

### B.1.2 Statistical analysis of SSL in PHM

Planetary gearbox RUL prediction, Complex groove welding monitoring, Pothole detection power line insulator detection, Screw product surface defect detection, Ductile cast iron pipe defect classification, Aeroengine turbine blade defect detection, Corrosion detection in marine vessels, Stamping progressive die anomaly detection, Oil and gas pipeline defect detection, Civil infrastructure damage and corrosion detection, Laser powder bed fusion anomaly detection, Cyber-physical power systems fault diagnosis, Helical gearbox defect detection, Industrial blower ball bearing condition monitoring, Flatness defect classification in the steelworks industry)

**Table B.2: Summary of applications, models, and signal types for SSL in PHM.**

Application	Dataset/Case	Ref.	Models	Signals
Semiconductor manufacturing	WM-811K wafer map dataset	[204, 205, 206, 207, 208, 209]	CNN, GAN, PatchCore, Ensemble learning, VAE	2D

Continued on next page

Table B.2 – continued from previous page

Application	Dataset/Case	Ref.	Models	Signals
Industrial product surface defect detection	MVTec AD dataset, DAGM dataset, AITEX dataset, magnetic tile dataset	[210, 211, 212, 213, 214, 215, 216, 217, 218, 219]	GAN, Autoencoder, CNN, Attention mechanism, Transformer, Contrastive learning, Capsule network	2D
Rolling bearing fault diagnosis	CWRU bearing dataset, XJTU-SY bearing dataset, IMS bearing dataset, Paderborn dataset, self-made experiment rig dataset	[220, 221, 222, 223, 224, 225, 226, 227, 216, 228, 229, 230]	LSTM, CNN, Transformer, Autoencoder, Attention mechanism, GRU, Contrastive learning, Random forest, Wavelet transform	1D, 2D, 1D time series, 2D time-frequency
Steel surface defect detection	NEU-Seg dataset, Severstal Steel Defect Detection (SSDD) dataset	[231, 232, 233]	CNN, GAN, Attention mechanism	2D
Rotating machinery fault diagnosis	Motor bearing dataset, CWRU gearbox dataset	[234, 235, 236, 237, 238, 239, 240, 225]	CNN, LSTM, Autoencoder, Transfer learning, Domain adaptation, Contrastive learning, Transformer, GAN, LSTM Autoencoder, Stacked Denoising Autoencoder	1D, 1D time series

Continued on next page

Table B.2 – continued from previous page

Application	Dataset/Case	Ref.	Models	Signals
Wind turbine fault diagnosis and remaining useful life prediction	Wind turbine gearbox vibration data, SCADA and vibration data from wind turbines	[241, 242, 243, 244, 245, 246]	LSTM, CNN, Transformer, Autoencoder, Attention mechanism, Meta learning, Wiener process, Semi-supervised learning algorithms, Variational Autoencoder, Kernel Density Estimation	1D, Multivariate time series, Time series
Lithium battery defect detection	-	[247, 248]	GAN, CNN, Markov chain, Neural network	2D, 3D
Centrifugal pump fault diagnosis	Centrifugal pump vibration data	[249]	CNN, Contrastive learning	1D, 2D kurtogram images
Air conditioning system fault detection	-	[250]	Machine learning	1D
Nuclear power plant fault detection	Simulated time-series data from nuclear power plants	[251, 252]	CNN, Diffusion model	1D, 2D imaged time-series
Electrical equipment fault diagnosis	Simulated data of induction motor faults	[253, 242], 254,	CNN, LSTM, Contrastive learning, Machine learning algorithms	1D, 2D, Time series
Chemical process fault detection and monitoring	Tennessee Eastman Process, Chemical process datasets	[255]	Kernel LSTM, Reduced Partial Squares, PLS, VAE, Kernel Least	1D, Multivariate time series

Continued on next page

B.1. Supplementary notes on PIML and SSL in PHM

**Table B.2 – continued from previous page**

<b>Application</b>	<b>Dataset/Case</b>	<b>Ref.</b>	<b>Models</b>	<b>Signals</b>
Aircraft turbofan engine remaining useful life prediction	C-MAPSS datasets	[243, 256]	LSTM, Autoencoder, Transformer, domain adaptation	Multivariate time series
Planetary gearbox remaining useful life prediction	Planetary gearbox degradation data	[244]	LSTM, Fractional Generalized Pareto Motion	Time series
Cyber-physical power systems fault diagnosis	IEEE 118-bus system simulated data	[257]	GAN	Multivariate time series
Transformer bushings anomaly detection	Current magnitude and phase angle data from transformer bushings	[258]	LSTM coder	Autoencoder Multivariate time series
Monitoring and abnormal detection across different manufacturing and industrial transmission settings	no specific mentioned	[259, 260, 261, 238, 263]	Sparse autoencoder, CNN, Attention and transformer, Markov transition field, LSTM, Autoencoder	1D time, multivariate time series

Continued on next page

Table B.2 – continued from previous page

Application	Dataset/Case	Ref.	Models	Signals
Including but not limited to defect detection, quality inspection, and anomaly monitoring across various industries like aerospace, civil infrastructure, electronics, and automotive	no specific mentioned	[264, 265, 266, 267, 215, 240, 268, 166, 269, 270, 225, 271, 272, 273, 253, 227, 274, 275, 276]	CNN, Segmentation model, Location-aware CNN, Neighborhood coordinate descriptor, Denoising autoencoder, Contrastive learning, Gaussian density estimation, Siamese network, Entropy pruning, Image segmentation, GAN	2D

In Table. B.2, the signal used in SSL for PHM span 1D, 2D, and multivariate time series data. Compared to PIML, 2D image-based data is prevalent in SSL applications. Time series data, both univariate and multivariate, is majorly encountered in applications like rolling bearing fault diagnosis, wind turbine fault diagnosis, and aircraft turbofan engine RUL prediction. In addition, the models commonly used in SSL include convolution neural networks (CNN), long short-term memory (LSTM), auto-encoders, transformers, and generative adversarial networks (GAN).

## B.2 Review articles on PIML

The review articles in Table. B.3 section has provided valuable insights into PIML, including the establishment of a PIML taxonomy, the identification of key research challenges, and the recognition of PIML as a promising approach to address issues such as physical consistency, data scarcity, and model interpretability. However, these existing reviews primarily focus on the methodological aspects of embedding physics within machine learning architectures and do not specifically address the unique challenges and applications within the field of PHM.



**Table B.3:** Existing review papers that mention PIML in PHM

Authors	Topics	Challenges	Taxonomy
Rai, Rahul, and Chandan K. Sahu[277]	Cyber-physical system's dynamic behavior modeling	<ul style="list-style-type: none"> <li>- Discretization approximation of the continuous system behavior in a chaotic environment</li> <li>- Scenario-oriented PIML hybrid framework</li> <li>- Efficient extraction of causal and model parameter relationships in big data</li> </ul>	<ul style="list-style-type: none"> <li>- Physics-based data pre-processing</li> <li>- Physics-guided ML algorithm structure design</li> <li>- Physics-based ML regularization item</li> </ul>
Willard, J., Jia, X., Xu, S., Steinbach, M.[278]	Engineering and environmental systems modeling, model solving methods	<ul style="list-style-type: none"> <li>- Embedding incomplete physics knowledge</li> <li>- Keeping physical consistency in data mining</li> <li>- Sparse data and uncertainty quantitative identification</li> </ul>	<ul style="list-style-type: none"> <li>- Physics-based regularization item in ML algorithm</li> <li>- Physics-guided ML initialization</li> <li>- Physics-informed ML algorithm architecture design</li> </ul>
Kim, S. W., Kim, I., Lee, J., Lee, S.[279]	Physics-informed deep learning in dynamical systems behavior modeling	<ul style="list-style-type: none"> <li>- Designing prior informed deep learning framework</li> <li>- ML training data scarcity</li> <li>- Keeping physical consistency</li> </ul>	<ul style="list-style-type: none"> <li>- Physics-informed Feature engineering</li> <li>- Physics-informed NN structure</li> <li>- Physics-informed loss function</li> </ul>
Jan Haggendorfer, Elias[280]	Condition monitoring	<ul style="list-style-type: none"> <li>- ML black-box nature explanation</li> <li>- Training data scarcity</li> <li>- Keeping physical consistency</li> </ul>	<ul style="list-style-type: none"> <li>- Parallel/Series physics-ML combination structure</li> <li>- Physics-based regularization item in ML objective function</li> </ul>

Continued on next page

Table B.3 – Continued from previous page

Authors	Topics	Challenges	Taxonomy
Finegan, D. P., Zhu, J., Feng, et.al.[19]	Battery cell state prediction	Keeping physical consistency	<ul style="list-style-type: none"> <li>- Physics-based data pre-processing</li> <li>- Physics-guided ML algorithm architecture design</li> <li>- Physics-based regularization item in ML algorithm</li> </ul>
Jianjing Zhang, Robert X. Gao[281]	Data curation and model interpretation for smart manufacturing	<ul style="list-style-type: none"> <li>- Non-interpretable prediction logic in deep learning</li> <li>- Error or imbalance training data</li> <li>- Data and data labels scarcity</li> </ul>	<ul style="list-style-type: none"> <li>- Physical model bias compensation and unknown parameters estimation via deep learning</li> <li>- Involving Physics-constraints into deep learning training</li> </ul>
Xu, Yanwen and Kohtz, et.al.[282]	Reliability analysis and risk assessment, Uncertainty quantification	<ul style="list-style-type: none"> <li>- Scenario-oriented PIML hybrid framework and its computational efficiency</li> <li>- Incompleteness of physics knowledge and limited representatives of the training dataset</li> <li>- The need for accurate and reliable data to create an accurate digital twin model</li> </ul>	<ul style="list-style-type: none"> <li>- Physics-informed architecture</li> <li>- Physics-informed loss function</li> </ul>
Thelen Adam, Zhang Xiaoge, and Fink Olga et al.[283, 284]	Physical system modeling	<ul style="list-style-type: none"> <li>- Integrating data from different sources and formats</li> <li>- Scaling up the digital twin model to larger and more complex systems</li> <li>- Validating the digital twin model against the physical system it represents</li> </ul>	<ul style="list-style-type: none"> <li>- Modifying the loss function</li> <li>- Generating synthetic data</li> <li>- Pre-training on physics-based data</li> <li>- Correcting models with unmodeled physics</li> <li>- Learning to predict inputs</li> </ul>

## B.3 Focus of existing doctoral theses

Phd thesis on PIML.

**Table B.4:** Summary of recent PhD theses on physics-informed machine learning (PIML).

Ref.	Year	Focus	Contribution
Mann, Vipul [285]	2024	Integrating domain knowledge into language models for process systems engineering	Developed approaches to incorporate domain expertise into language models for improved accuracy, interpretability, and generalization
Wang, Sifan [286]	2023	Theoretical foundations, algorithms, and applications of PIML	Investigated challenges and limitations of PINNs and DeepONet; proposed loss re-weighting algorithms and architectures for improved performance
Bahmani, Bahador [287]	2024	Geometry-informed data-driven mechanics	Introduced manifold embedding data-driven paradigm and geometric autoencoders for learning noise-free embeddings
Romeo, Shafi A.S. [288]	2023	Nonlinear system discovery and machine learning for dynamical systems	Developed methods for extracting physics from data, including sparse identification of nonlinear dynamical systems (SINDy)
Russell, Collin [289]	2023	PINN for solving parametric, nonlinear heat conduction problems	Demonstrated PINN's ability to solve heat conduction problems with parameterized, temperature-dependent material properties without training data
Huynh, Phat K. [290]	2023	Knowledge integration in domain-informed machine learning and multi-scale modeling	Developed hybrid frameworks combining physics-based models with machine learning for various damage scenarios

Continued on next page

B.3. Focus of existing doctoral theses

Table B.4 – continued from previous page

Ref.	Year	Focus	Contribution
Ghaderi, Aref [291]	2023	Physics-informed data-driven models for inelastic, aging, failure behavior of crosslinked polymers	Introduced hybrid models for predicting durability and properties of elastomers under environmental damage
Haque, Tas-miah [292]	2023	Physics-infused LSTM for marine vessel track association based on AIS data	Demonstrated improved performance of physics-infused LSTM over physics-based models for vessel tracking with time gaps and overlaps
Ansari, Mehrad [293]	2023	Applications of PIML in chemical engineering	Combined physics-based modeling with machine learning for computational fluid dynamics, epidemiological modeling, and peptide design
Stone, Thomas [294]	2023	PINNs for ultrasonic guided wave propagation in solid media	Explored PINN as an alternative to FEM for efficiently simulating ultrasonic guided wave inspections
Alvarado, Walter [295]	2023	Understanding chromatin remodeling through physics-based machine learning	Employed meso-scale modeling, manifold learning, and convolutional autoencoders to study chromatin folding and tetranucleosome motifs
Dajkhosh, Seyedeh P. [296]	2023	Structured illumination microscope image reconstruction using unrolled PIGAN	Developed physics-informed GAN with attention-based super-resolution for improved 3D-SIM image reconstruction
Lu, Peter Y. [297]	2023	Interpretable PIML methods for scientific modeling and data analysis	Designed physics-informed architectures and representation learning methods for system identification, spatiotemporal analysis, and conservation law discovery

Continued on next page

### B.3. Focus of existing doctoral theses

Table B.4 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
Yu, Yang [298]	2023	Data-physics driven reduced order homogenization	Developed hybrid data-physics driven approach for reduced-order multiscale modeling of complex material systems
Qiao, Zhuoran [299]	2023	Physics-informed neural approaches for multiscale molecular modeling and design	Introduced orbital-based geometric deep learning and stochastic process models for predicting quantum chemical properties and protein-ligand complexes
Prantikos, Konstantinos [300]	2022	PINN solution of point kinetics equations for nuclear reactor monitoring	Demonstrated feasibility of training a PINN to solve point kinetics equations for a startup transient of the PUR-1 reactor
Kamali, Ali [301]	2023	Physics-informed deep learning for quasi-static elasticity imaging	Introduced El-UNet architecture and self-adaptive spatial loss weighting for improved accuracy in solving inverse elasticity problems
Wang, Rui [286]	2023	Physics-guided deep learning for dynamics forecasting	Developed approaches to incorporate physical constraints, leverage multi-fidelity data, and embed symmetries for improved generalization
Rahman, Mohammad M. [302]	2023	Physics-aware deep learning for radar-based cyber-physical human systems	Proposed physics-aware GANs, domain adaptation, and cross-modal fusion for data-efficient learning from multi-frequency radar
Saha, Swapnil S. [303]	2023	Physics-aware tiny machine learning	Introduced TinyNS framework for automated neurosymbolic architecture search within hardware constraints

Continued on next page

B.3. Focus of existing doctoral theses

Table B.4 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
Aliakbari, Maryam [304]	2023	PINNs for solving forward and inverse fluid flow and heat transfer problems	Developed multi-fidelity modeling combining low-fidelity CFD with PINN and ensemble PINN for improved accuracy and uniqueness
Seman, Matthew [305]	2023	PIML models for PDEs with applications to laser bioeffects	Explored PINN, DeepONet, and FNO for solving heat diffusion PDE in multi-layer skin and ocular tissue models
Moseley, Benjamin [306]	2022	PIML: from concepts to real-world applications	Assessed scalability of PIML techniques to complex, real-world problems in lunar science and geophysics
Raynaud, Gaé-tan [307]	2021	PINNs for solving fluid-structure problems in turbine-like phenomena	Developed modal approach and investigated robustness of PINNs to incomplete, sparse, or noisy measurements
Pawar, Suraj [308]	2022	Physics-guided machine learning for turbulence closure and reduced-order modeling	Proposed frame invariant neural network and concatenated neural network for data-efficient, generalizable turbulence modeling
Desai, Shaan [309]	2021	PINNs for data-efficient learning	Investigated physics-informed architectures for learning biases, integrators, Hamiltonians to model complex systems from sparse data
Monakhova, Kristina [310]	2022	PIML for computational imaging	Incorporated imaging physics into neural networks via algorithm unrolling, differentiable models, unsupervised learning, and GANs

Continued on next page

B.3. Focus of existing doctoral theses

Table B.4 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
He, Xiao-long [311]	2022	Thermodynamically consistent physics-informed data-driven computing and reduced-order modeling	Developed data-driven solvers, autoencoders, and non-intrusive ROM for nonlinear materials modeling
Wang, Nan [312]	2022	Modeling water waves and sediment transport using physics-based and machine learning methods	Developed coupled flow-wave models, soft computing models, hybrid approaches, and PINNs for nearshore processes
Ren, Pu [313]	2022	Embedding physics into deep learning for modeling spatiotemporal systems	Proposed convolutional-recurrent PINN, super-resolution, and sparse regression for forward and inverse modeling of PDEs
Zhang, Zhao [314]	2022	Freeway traffic flow modeling and forecasting using physics-guided machine learning	Developed hybrid PIML, PGML, and PG-LSTM for improved accuracy and generalization in traffic state estimation and prediction
Kong, Xiang-hao [315]	2022	PIML models for power transmission systems	Developed Proximal Bilateral Random Projection for event detection and physics-based neural ODEs for generator parameter estimation from PMU data
Ihunde, Thelma A. [316]	2022	Application of PINNs to compositional modeling	Demonstrated PINNs' ability to honor governing physics in compositional fluid flow modeling compared to standard neural networks

Continued on next page

### B.3. Focus of existing doctoral theses

Table B.4 – continued from previous page

Ref.	Year	Focus	Contribution
Cofré Martel, Sergio Manuel Ignacio	2022	PINN framework for prognostics and health management using big machinery data	Developed preprocessing methodology and interpretable PINN-RUL framework for remaining useful life estimation in complex engineering systems
Kapusuzoglu, Berkcan [317]	2022	PIML for uncertainty quantification and optimization	Investigated combining physics-based models with machine learning for design optimization under uncertainty in additive manufacturing
Ghosh, Aban-tika [318]	2022	PIML for optical metamaterials	Developed physics-informed learning to incorporate symmetries and solve eigenvalue problems in electromagnetic metamaterial design and characterization
Behl, Mark V. [319]	2020	Reduced-order modeling using reservoir simulation and PIML	Demonstrated hybrid physics-ML reduced-order models for efficient and accurate reservoir production forecasting
Zargar, Sakib A. [320]	2022	Sparse-sensor structural health monitoring via physics-informed deep learning	Developed PINN for guided wave reconstruction and deep learning models for impact/damage diagnostics from reconstructed wavefields
Chen, Zhao [321]	2021	Physics-informed learning of complex systems with sparse data	Introduced Bayesian learning, sparsity-promoted PIDL, symbolic neural networks, and Bayesian physics-encoded forecasting for sparse data
Fuks, Olga [322]	2020	PIML and uncertainty propagation for multiphase transport in porous media	Extended FROST method for uncertainty quantification and investigated PIML for handling shocks in two-phase transport PDEs.

**Phd thesis on SSL.** The thesis related to SSL highlights several major contributions to the methodologies development of SSL. Improved architectures and



loss functions have been proposed, including novel neural network designs (e.g., Siamese networks, autoencoders) and objective functions (e.g., contrastive loss, reconstruction loss), which enhance self-supervised representations [323, 324, 325]. Multi-modal and cross-modal methods have been developed to leverage complementary information from multiple modalities (e.g., audio-visual data), enabling applications in diverse domains [326, 327, 328]. The combination of self-supervision with semi-supervised learning has emerged as a promising paradigm, where self-supervised pre-training followed by fine-tuning on a small labeled dataset improves performance in low-data regimes [160, 329]. Additionally, theoretical analyses have provided insights into the principles of SSL, such as connections to mutual information estimation and the role of data augmentations [168, 330].

In applications, SSL has been effectively applied across various domains to enhance the generalization ability of diagnostic models and improve performance in scenarios with limited labeled data. In biomedical data analysis, techniques have been used to analyze electrocardiogram signals and x-ray images, resulting in improved diagnostic models [329, 331]. In human activity recognition, SSL has enabled the development of robust models using unlabeled sensor data, with applications in healthcare, entertainment, and fitness [332, 167]. For industrial monitoring, these techniques have been utilized for tasks such as defect detection in wafer maps and learning representations for machine vision [327, 324]. In computer vision, self-supervised methods have been widely applied to problems like object recognition, tracking, and representation learning, supporting downstream tasks such as few-shot learning and domain adaptation [333, 325, 232]. Additionally, in natural language processing, SSL has facilitated the development of efficient representations for text data with limited annotations, benefiting applications like sentiment analysis, and machine translation [166].

Ref.	Year	Focus	Contribution
[334]	2024	Explainable AI	Developed methods for making AI systems more explainable to humans and applicable to scientific problems
[330]	2024	Dynamic causality	Proposed using sparse symbolic regression to learn causal dynamic equations from data and applied it to counterfactual reasoning

Continued on next page

B.3. Focus of existing doctoral theses

Table B.5 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
[335]	2024	Domain generalization	Developed representation learning techniques to improve model performance on new domains
[160]	2023	Semi-supervised learning	Combined semi-supervised learning with self-supervision to handle both closed-set and open-set scenarios
[326]	2024	Multi-modal learning	Advanced theoretical understanding and practical applications of multi-modal deep simultaneous learning
[327]	2023	Industrial monitoring	Demonstrated the effectiveness of self-supervised learning techniques for monitoring industrial environments
[161]	2023	Few-shot and semi-supervised learning	Proposed efficient methods for few-shot and semi-supervised learning in computer vision with limited labeled data
[163]	2023	Nuclear nonproliferation	Augmented limited gamma spectroscopy data and applied self-supervised learning for nuclear nonproliferation monitoring
[336]	2023	Human-AI collaboration	Investigated techniques for effective knowledge transfer between humans and AI systems during collaboration
[166]	2023	NLP with limited data	Developed NLP methods that can work well in low-resource settings with limited data and compute
[165]	2023	Heterogeneous data	Enabled knowledge discovery and natural language querying of heterogeneous data sources

Continued on next page

### B.3. Focus of existing doctoral theses

Table B.5 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
[337]	2021	Edge detection	Showed how to learn edge detectors from unlabeled natural images without human annotation
[164]	2023	Efficient representation learning	Proposed representation learning methods that are more data and label efficient
[338]	2023	Automatic data labeling	Provided an overview of techniques, challenges and solutions for automatically labeling data with machine learning
[339]	2023	Robust representation	Introduced ViewMix augmentation strategy to learn more robust representations with self-supervision
[169]	2023	Structured representation learning	Incorporated structural information into the representation learning process
[162]	2023	Explainable models	Used self-supervised pre-training to improve the explainability of learned models
[340]	2022	Graph learning with noisy labels	Developed techniques to learn from graphs with sparse and noisy label information
[341]	2023	World model learning	Enabled agents to learn action policies from diverse data via learned world models
[342]	2023	Multi-modal representation learning	Used metric learning to align representations across modalities and learn from limited data
[167]	2022	Human activity recognition	Proposed sequential self-supervised learning to share representations across sensor modalities for HAR

Continued on next page

### B.3. Focus of existing doctoral theses

Table B.5 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
[343]	2023	Neural-symbolic AI	Integrated neural and symbolic approaches for more interpretable and generalizable AI systems
[344]	2023	Noisy label learning	Proposed a unified model to handle various types of label noise in machine learning
[345]	2023	Cyber-physical systems	Enabled learning in cyber-physical systems with limited and imperfect sensor data
[216]	2023	Causal representation learning	Analyzed deep visual representations using causal frameworks and unlabeled data
[346]	2023	Task affinity	Leveraged task relationships to improve knowledge transfer and multi-task learning
[347]	2023	Learning with limited supervision	Developed methods to learn in realistic limited-supervision settings
[325]	2023	Object recognition and tracking	Utilized self-supervision on downstream tasks to improve object recognition and tracking
[328]	2021	Multimodal learning	Advanced self-supervised representation learning from multiple modalities
[348]	2021	Visual object representations	Learned structural visual representations without human labels via self-supervision
[232]	2022	Few-shot learning	Showed the importance of representation learning for few-shot model building
[349]	2021	Autonomous agents	Used curiosity as a self-supervised objective to improve learning in situated autonomous agents

Continued on next page

### B.3. Focus of existing doctoral theses

Table B.5 – continued from previous page

<b>Ref.</b>	<b>Year</b>	<b>Focus</b>	<b>Contribution</b>
[324]	2022	Domain and modality transfer	Advanced representation learning techniques for improved cross-domain and cross-modal transfer
[323]	2022	Lightweight models	Compressed self-supervised representations for more efficient lightweight models
[168]	2022	Theory	Advanced theoretical understanding of self-supervised representation learning principles
[332]	2022	Human activity recognition	Demonstrated how to build robust HAR models using unlabeled sensor data
[331]	2022	Biomedical applications	Utilized disentanglement and semi-supervised learning for biomedical problems with limited labels
[350]	2022	Open-world learning	Enabled machine learning systems to cope with limited labels in open-world settings
[351]	2022	Robot learning	Studied how robots can learn perception and control in partially unknown environments
[324]	2022	Semi-supervised defect recognition	Showed benefits of self-supervision for semi-supervised wafer defect classification with few labels
[333]	2021	Visual feature learning	Learned effective visual features from limited labeled images
[352]	2021	Rare failure detection	Applied machine learning for detecting rare failures in analog and mixed-signal hardware
[329]	2021	Biomedical applications	Used disentangled representations and semi-supervised methods for biomedical tasks with limited labels

The thesis related to SSL highlights several major contributions to the methodologies development of SSL. Improved architectures and loss functions have been proposed, including novel neural network designs (e.g., Siamese networks, autoencoders) and objective functions (e.g., contrastive loss, reconstruction loss), which enhance self-supervised representations [323, 324, 325]. Multi-modal and cross-modal methods have been developed to leverage complementary information from multiple modalities (e.g., audio-visual data), enabling applications in diverse domains [326, 327, 328]. The combination of self-supervision with semi-supervised learning has emerged as a promising paradigm, where self-supervised pre-training followed by fine-tuning on a small labeled dataset improves performance in low-data regimes [160, 329]. Additionally, theoretical analyses have provided insights into the principles of SSL, such as connections to mutual information estimation and the role of data augmentations [168, 330].

In applications, SSL has been effectively applied across various domains to enhance the generalization ability of diagnostic models and improve performance in scenarios with limited labeled data. In biomedical data analysis, techniques have been used to analyze electrocardiogram signals and x-ray images, resulting in improved diagnostic models [329, 331]. In human activity recognition, SSL has enabled the development of robust models using unlabeled sensor data, with applications in healthcare, entertainment, and fitness [332, 167]. For industrial monitoring, these techniques have been utilized for tasks such as defect detection in wafer maps and learning representations for machine vision [327, 324]. In computer vision, self-supervised methods have been widely applied to problems like object recognition, tracking, and representation learning, supporting downstream tasks such as few-shot learning and domain adaptation [333, 325, 232]. Additionally, in natural language processing, SSL has facilitated the development of efficient representations for text data with limited annotations, benefiting applications like sentiment analysis, and machine translation [166].

Although these applications do not specifically focus on PHM, the demonstrated potential of SSL in addressing the challenges of limited labeled data and advancing the state-of-the-art inspires further exploration and application within the PHM domain. However, several challenges need to be addressed for board usage:

- Lack of supervisory signals: SSL relies on automatically generated pseudo-labels or proxy tasks, which may not always align well with the downstream task of interest. This can limit the quality of the learned representations [160, 161, 162].
- Sensitivity to data augmentations: The performance of self-supervised methods often heavily depends on the choice of data augmentations used to

generate different “views” of the input data. Designing effective augmentations requires domain knowledge and can be challenging for some modalities [163, 164].

- Scalability to diverse domains: Many self-supervised techniques are developed and evaluated primarily on image datasets. Extending these methods to other domains like text, audio, or sensor data may require non-trivial adaptations [165, 166, 167].
- Limited theoretical understanding: While empirical results have shown the effectiveness of SSL, the theoretical underpinnings of why and when these methods work well are still not fully understood [168, 169].

# Appendix: Physics-informed machine learning in PHM

---

## C.1 Detailed description of the data generation process for the validation case in Section 3.2.1

Critical components like bearings are susceptible to fatigue damage, often initiated internally. Detection of degradation signs is challenging, especially during the crack expansion phase when they are not visible [353]. However, once visible cracks emerge, they tend to propagate rapidly, hastening the path to failure [354]. Thus, challenges in using vibration signal RUL prediction include:

- Vibration signals exhibit a subtle “light trend”, complicating early degradation trend capture [355].
- Capturing the underlying non-linear degradation dynamics without explicit knowledge of the bearing’s condition is challenging.

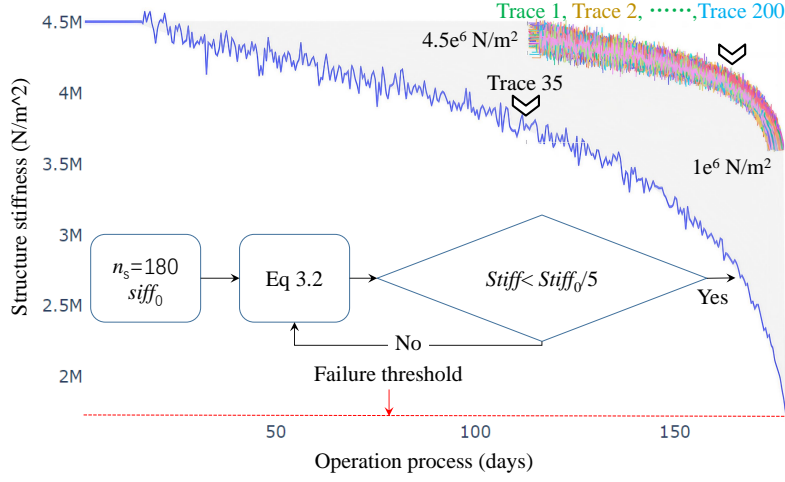
Bearing degradation, modeled by stiffness deterioration, significantly affects vibration response amplitude (Eq. C.1 [171]). Notably,  $Vib_p$  represents peak vibration signal value, while  $stiff$  denotes equivalent contact stiffness.  $\varepsilon$  reflects system load imbalance due to extrinsic excitation, with  $m$  as equivalent system mass and  $\Omega$  as rotation speed. Typically,  $\varepsilon$  and  $m$  remain unknown in real conditions, with only  $\Omega$  and  $Vib_p$  being measurable.

$$Vib_p = \frac{\varepsilon\Omega^2}{stiff - m\Omega^2} \quad (C.1)$$

Figure C.1 depicts 200 simulated bearing stiffness degradation trajectories, averaging a failure time of  $8.04 \times 10^5$  cycles with a standard deviation of  $1.47 \times 10^4$  cycles. Among these, 50 trajectories are reserved for testing, while the remaining 150 serve for training and validation. Each trajectory encompasses distinct health states, non-linear degradation periods, and uncertainties from operational conditions.



C.1. Detailed description of the data generation process for the validation case in Section 3.2.1



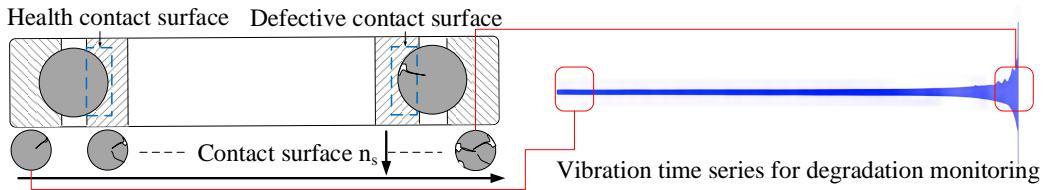
**Figure C.1:** Simulation process of contact stiffness degradation.

The stiffness for different damage states is calculated by Equation (C.2) [356]:

$$stiff = \frac{1}{2 \left( \frac{(\cos \gamma)^{5/2}}{n_s k_p} \right)^{2/3}} + U_{stiff} \quad (C.2)$$

Here,  $\gamma = 20^\circ$  is the contact angle,  $k_p$  is the Hertzian elastic contact stiffness, and  $U_{stiff}$  represents model uncertainty, assumed to follow a skewed distribution with parameters centered at 10% mean and 5% variance of  $stiff$ .  $n_s = 180$  is the initial number of contact surfaces, with  $R = 0.003$  m as the roller radius,  $\nu = 0.3$  for Poisson's ratio, and  $E = 2.1 \times 10^{11}$  Pa as Young's modulus.

$$k_p = \frac{4R^{1/2}}{6 \left( \frac{1-\nu^2}{E} \right)} \quad (C.3)$$



**Figure C.2:** Defective contact and roller failure schematic.

The simulated decrease in  $n_s$  is modeled by:

$$n_{si} = n_{s0} \times steps \times i \times (180 - steps \times i) + U_{deg}, \quad i \in \mathbf{N} \quad (C.4)$$

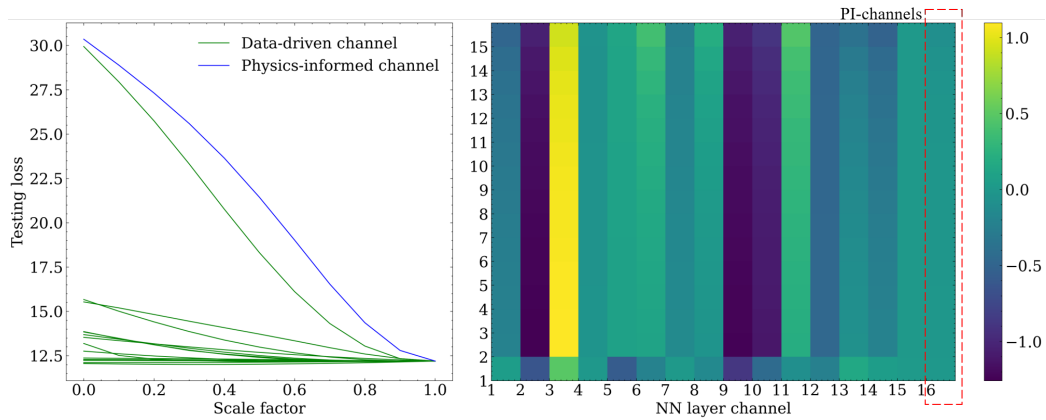
## C.2. Deep discussion on the performance of the PIML models proposed in Section 3.2.1

$U_{deg}$  also follows a skewed distribution, with mean and variance set to 10% and 5% of  $n_{si}$ , respectively. During degradation, vibration amplitudes are calculated every six hours, and the resultant simulated signal is modelled by Equation (C.5):

$$x(t) = vib_p \times \sin\left(\frac{2\pi\Omega}{60}t\right) + vib_p \times \sin\left(\frac{2\pi\Omega}{60}t(1 + n_{d1})\right) + n_{d2} \quad (C.5)$$

## C.2 Deep discussion on the performance of the PIML models proposed in Section 3.2.1

**Investigating the impact of embedded physics-knowledge on the final results.** This section investigates the impact of embedded knowledge in PIML through two perspectives. Firstly, it conducts a channel-by-channel test on the layer embedded with physical knowledge, assessing the channel’s influence on the final test loss across different weight compression ratios (0-1). Secondly, it calculates the output of the hidden layer corresponding to each channel and averages the values across all samples to generate a heatmap of information output. The final results are illustrated in Fig. C.3, C.4, and C.5.



**Figure C.3:** Heat map of the PI-input space model.

In the channel information heat map of the PIFM, see Fig. C.3, a predominant focus on channels containing the  $\Omega^2/Vib_p$  is observed. This is because changes in the weighting of the information for this channel will have the greatest impact on changes in the loss. However, the first Conv1D layer’s output of this feature does not occupy the most prominent position on the feature heat map.

This phenomenon is still present in the PI-layer model, but since the location of the physical knowledge is limited to the hidden layer at this point, the effect

C.2. Deep discussion on the performance of the PIML models proposed in Section 3.2.1

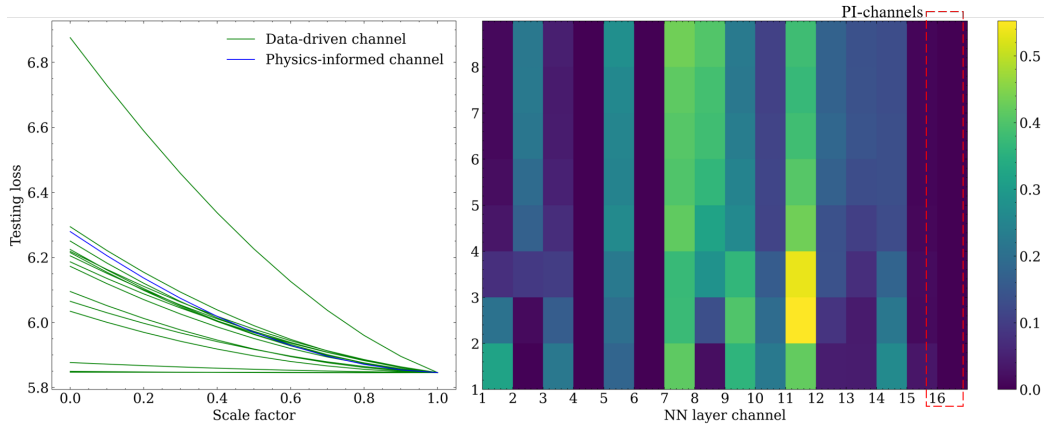


Figure C.4: Heat-map of PI-layer model.

of the change in the weight of its corresponding channel is significant but not the dominant factor, and at this point the effect of the corresponding output of the first “Conv1d” layer in “Resblock” after the “Add” layer in the corresponding output heat map of the section is not significant.

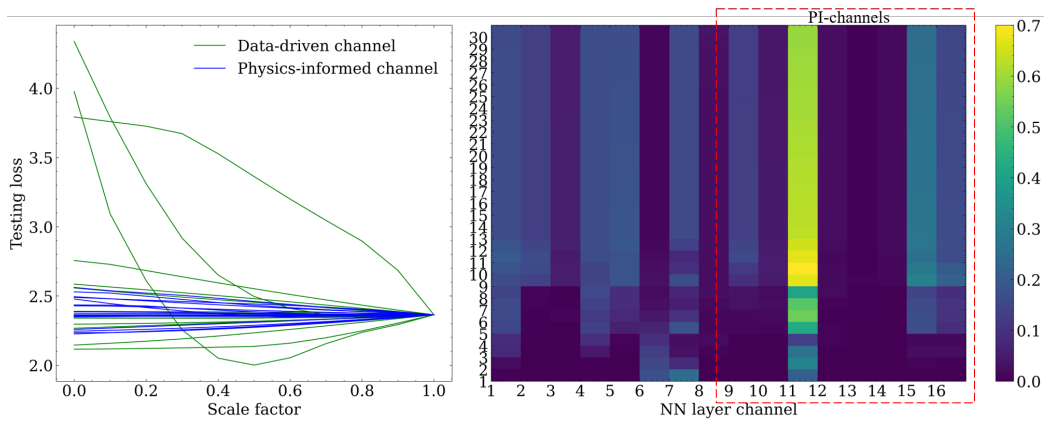


Figure C.5: Heat map of the PI-loss model “Add” Layer.

On the basis of the PI-layer model, the branches embedded with physical knowledge have separate outputs, and the final result is obtained by combining the results of the branch NN and Main NN, so that it can be seen that features belonging to the PI-branches are highlighted, but correspondingly, the fluctuation of their output loss under the variation of the weights of these channels is overall smaller.

Since then, the following insights about embedding physical knowledge in different network structures is obtained: the location and manner in which phys-

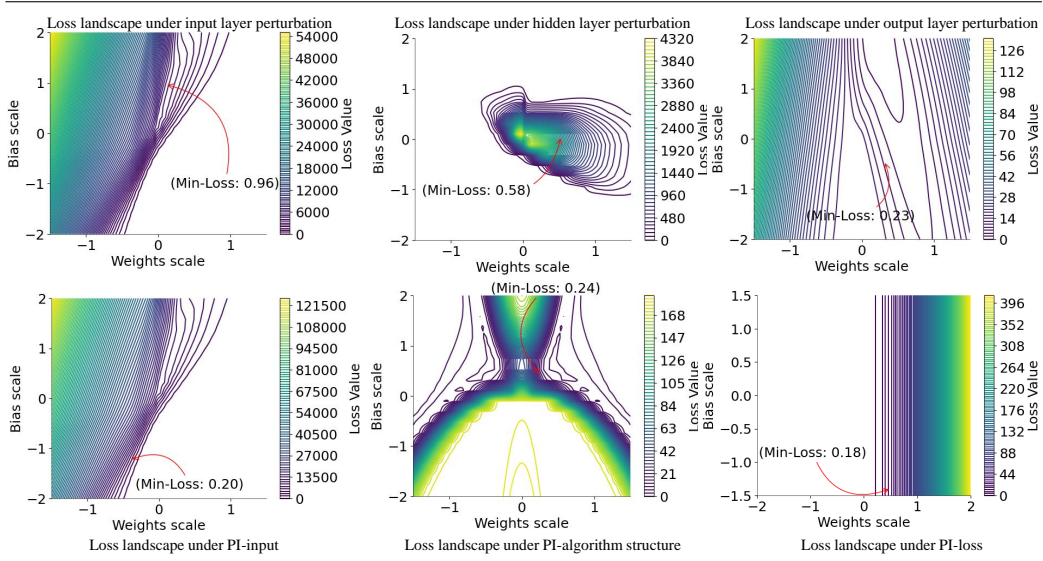
ical knowledge is embedded can greatly affect its effect. In general, the closer to the input layer, the more effective the embedding is, but it may also be diluted by the subsequent network layers. In architecture design, there is a trade-off between highlighting physical features and obtaining robust integrated representations, and the information under the physical knowledge informed by cross-layer connections should be involved in the final decision-making as much as possible to reduce the dilution.

***Impact of embedded physical knowledge on model optimization.*** The preceding theoretical analysis demonstrated that the informed mathematical formulation knowledge significantly influences the model’s gradient-seeking optimization, particularly impacting the loss landscapes. This study, therefore, examines the loss contour maps of both the benchmark model and the physics-informed positions in PI-TCN to elucidate the effect of embedded knowledge on model performance.

The values of weights and biases extracted from the trained model are utilized to construct the coordinates for the contour maps. An array of equally spaced grid points is generated to express, based on the ranges of the combinations of weights and biases, raw data at different scaling scales to serve as coordinates for the contour plots. The corresponding loss function values are calculated for each combination of weights and biases. The findings are depicted in Fig. C.6.

Analysis of the minimum loss path reveals a distinct, narrow “canyon” in both models, indicating high sensitivity to slight perturbations in the input layer’s weights and biases. Deviations from this path lead to a rapid increase in loss, highlighting a significant performance decline. The loss landscape is predominantly smooth, with few local minima. The vicinity of the minimum loss point features a “flat” area, forming a “basin” that provides fault tolerance near the optimum. With the integration of physical features to construct the physics-input space model, the fundamental contour of the optimal path remains consistent; however, the gradients flanking the minimum loss path become gentler, and the “basin’s” extent broadens, suggesting that physical constraints offer a larger parameter adjustment space near the optimal solution, thus reducing the likelihood of overfitting. The loss range for the physics-input space model extends from 13,500 to 121,500, which is 2.25 times that of the benchmark model. This expansion in loss value space is attributed to the enriched information from the physical features, likely facilitating the exploration of a wider range of loss values. Furthermore, the minimum loss value decreases from 0.96 in the benchmark model to 0.20 in the physics-informed model, indicating that the incorporation of physical features reduces the optimization challenge and enables the achievement of lower loss values.

## C.2. Deep discussion on the performance of the PIML models proposed in Section 3.2.1



**Figure C.6:** Comparison of loss landscapes under different conditions.

The horizontal axis represents the scaling factor for the original weights of the layer, while the vertical axis represents the scaling factor for the original biases. The red arrow curve indicates the path of minimum loss.

Comparative analysis of the layer parameter perturbation test at the middle hidden layer shows that the benchmark model’s minimum loss path or “canyon” is more complex and curved, unlike the straight path observed in the PI-layer model. The “canyon” of the PI-layer model is narrower and more pointed, running vertically through the middle, with steep slopes on either side forming a “V” shape, altering the entire loss landscape. This narrow path suggests that the PI-layer model is “locked” into a very specific and narrow range of weights for minimum loss, making it less flexible and robust than the benchmark model. In contrast, the benchmark model exhibits a much wider range of losses and is less sensitive to weight perturbations.

When perturbations occur in the output layer, the minimum loss paths of the two models differ significantly. The benchmark model’s path is straight and narrow, resembling a “V” shape, whereas the PI-loss model’s path is wider and closer to a parabola, indicating greater robustness under weight perturbations. The minimum loss value of the PI-loss model is approximately 0.18, lower than the benchmark model’s 0.23, suggesting that the PI-loss model guides convergence to an optimal point with superior generalization performance, achievable over a broader range of loss search. The PI loss model’s optimal region is gentler, and the overall landscape is smoother, facilitating stable convergence through the gradient descent algorithm.

## C.3 Rotor unbalance and shaft crack experiments

Rotating equipment, critical to energy production and transmission, operates under harsh conditions, making it susceptible to rotor unbalances and other component defects due to manufacturing errors and extreme operating environments [357]. These defects can escalate into catastrophic failures, such as shaft transverse cracks, leading to severe economic losses and safety risks [cost]. A new PIML approach is proposed to address these challenges with two main objectives:

1. Effectively utilizing rotor responses to detect and localize combined failures of rotor unbalance and shaft cracks.
2. Developing a versatile model applicable across different rotor configurations to accommodate the diversity of rotating system structures.

Fig. C.7 depicts the PT 500 diagnostics platform [platform], equipped with a three-phase AC motor operating at 3000 *rpm* and featuring multiple components such as two long shafts, one short shaft, three supports, two discs, and a flange coupling. This setup allows for investigating diverse rotor structures by repositioning supports and discs. Vibration data is captured by two accelerometers on different supports, sampling at 4096 Hz via NI-DAQ 9174.

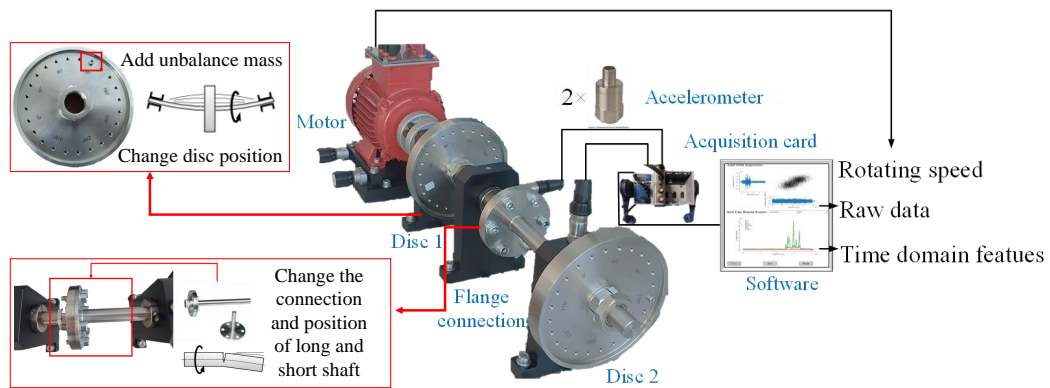


Figure C.7: Schematic diagram of the experimental platform.

**Multi-faults experiments.** In this study, the unbalance fault was created artificially by adding a  $2g$  screw to a counterweight hole on the disc (Fig. C.7). The position of the screw on the disc varied to investigate different unbalance positions and the number of screws used to simulate the degree of unbalance.

In the shaft-cracking experiments, different crack positions are simulated by connecting the long or short shaft to the rotor shaft (as in Fig. C.7).

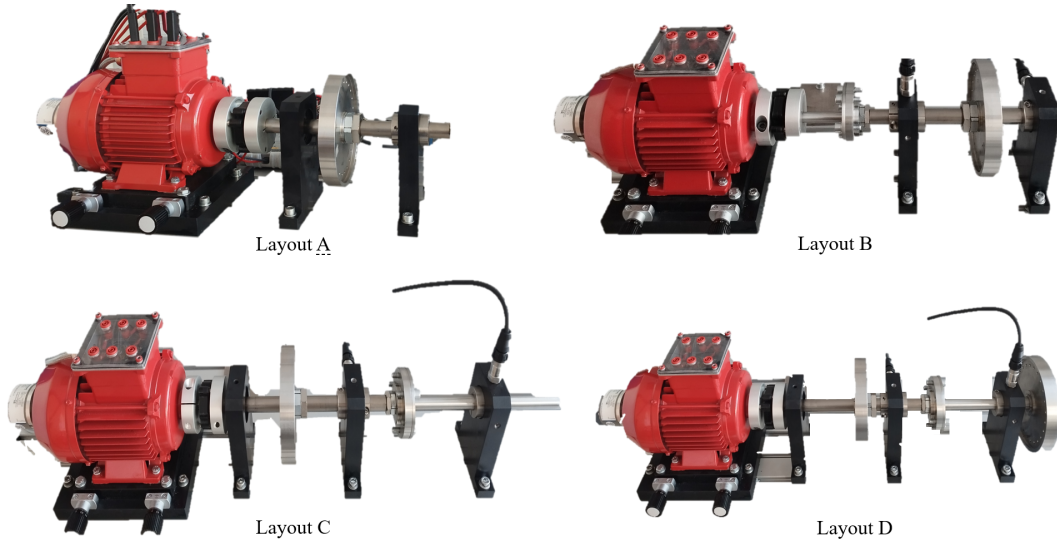
### C.3. Rotor unbalance and shaft crack experiments

Table C.1 summarizes different experiments investigated in this work. Four health states of the rotor are considered: healthy states (H), unbalance fault (U), crack fault (C), and combined fault (U&C). The positions of unbalanced defects and shaft cracks in different experiments are varied along the shaft length. Also, to highlight the robustness of the proposed model, different rotor structures (with 3 shaft lengths and 4 layouts, see Fig. C.8) are tested by varying the rotating speed.

**Table C.1: Overview of the experimental setup.**

Healthy states (H), unbalance fault (U), crack fault (C), and a combined fault (U&C).

Status	Shaft length (m)	Structure	Rotating speed (rpm)	Fault position (m)	Samples
H_1	0.355	Layout A	1200, 1500, 1800, 2100, 2400, 2700, 3000	-	300
U_1	0.355	Layout A	1200, 1500, 1800, 2100, 2400, 2700, 3000	0.175	300
H_2	0.409	Layout B	1500, 1800	-	40
C_1	0.409	Layout B	1500, 1800	0.120	40
U_2	0.409	Layout B	1500, 1800, 2400	0.290	60
H_3	0.605	Layout C	1200, 1500, 1800	-	60
C_2	0.605	Layout D	1500, 1800, 2400	0.355	60
U_3	0.605	Layout D	1500, 1800, 2400	0.207	60
U_4	0.605	Layout D	1500, 1800, 2400	0.110	60
U_5	0.605	Layout D	1500, 1800, 2400	0.155	60
U&C_1	0.605	Layout C	1200, 1500, 1800, 2100, 2400	C:0.155, U:0.586	60
U&C_2	0.605	Layout C	1200, 1500, 1800, 2100, 2400	C:0.355, U:0.175	60



**Figure C.8:** Different structure layouts of the test bench.

**Labeling data.** The manually labeled fault type, location, and rotor behavior features are defined for the multi-task learning (MTL) process:

Firstly, the classification labels are encoded using a binary one-hot method to denote the healthy state as (1, 0, 0), the unbalanced defect as (0, 1, 0), the shaft crack as (0, 0, 1), and the combined defects (resulting from both rotors unbalance and shaft cracks) as (0, 1, 1). In the RFEMNN, for multi-label classification results, sigmoid functions are used to produce a probability score for each possible label. By establishing a threshold, the resulting multi-label classification is then transformed from probability scores to binary values of either 0 or 1.

Secondly, the position label is provided by the axial distance of the defect from one end of the rotor. Since experimental tests involve several rotor structure layouts with different shaft lengths, it is necessary to normalize the defect distance by the corresponding whole shaft length  $L_{axis}$  separately.

Finally, the rotor vibration behavior label is the temporal statistical features, such as margin factor, impulse factor, peak factor, wave factor, kurtosis, skewness, RMSE, variance, and mean. They are used as labels to constrain the neural network's behavior by guiding RFEMNN to recover these characteristics of the raw data. These features are explained in detail in reference [358]. The predicted value corresponding to this label in NN is obtained through FEM inference using NN structural information rather than direct signal processing of the original data.

All the monitoring vibration signals are sliced into 39342 samples with a length of 256 points. Note that the labels of fault types, location, and temporal vibration features are assigned for each sample, as illustrated in Tab. C.2.



**Table C.2:** Illustration of the data labeling.

Healthy (H), unbalance (U), crack (C), and a combined fault (U&C).

Status	Fault type	Fault location(%)	Time features
H_1	(1, 0, 0)	(0, 0)	Margin factor
U_1	(0, 1, 0)	(0.175, 0)/0.355×100	Impulse factor
H_2	(1, 0, 0)	(0, 0)	Peak factor
C_1	(0, 0, 1)	(0, 0.120)/0.409×100	Wave factor
U_2	(0, 1, 0)	(0.290, 0)/0.409×100	Kurtosis
H_3	(1, 0, 0)	(0, 0)	Skewness
C_2	(0, 0, 1)	(0, 0.355)/0.605×100	Rmse
U_3	(0, 1, 0)	(0.207, 0)/0.605×100	Variance
U_4	(0, 1, 0)	(0.110, 0)/0.605×100	Mean
U_5	(0, 1, 0)	(0.155, 0)/0.605×100	
U&C_1	(0, 1, 1)	(0.586, 0.155)/0.605×100	
U&C_2	(0, 1, 1)	(0.175, 0.55)/0.605×100	

## C.4 RFEMNN structure details and the related physics introduction

Table. C.3 summarizes the physical parameters involved in the RFEMNN model and presents their determination method.

**Table C.3:** Physics parameters in RFEMNN.

Symbol	Parameter meaning	Determination method
$\rho$	Structure density	Design parameter
$R_d$	Shaft section outer diameter	Field measurement or Design parameter
$r_d$	Shaft section inner diameter	Field measurement or Design parameter
$th_i$	Thickness of the disc	Field measurement or Design parameter
$s_{ji}$	Support stiffness	Estimation or Design parameter
$d_{ji}$	Support damping	Estimation or Design parameter

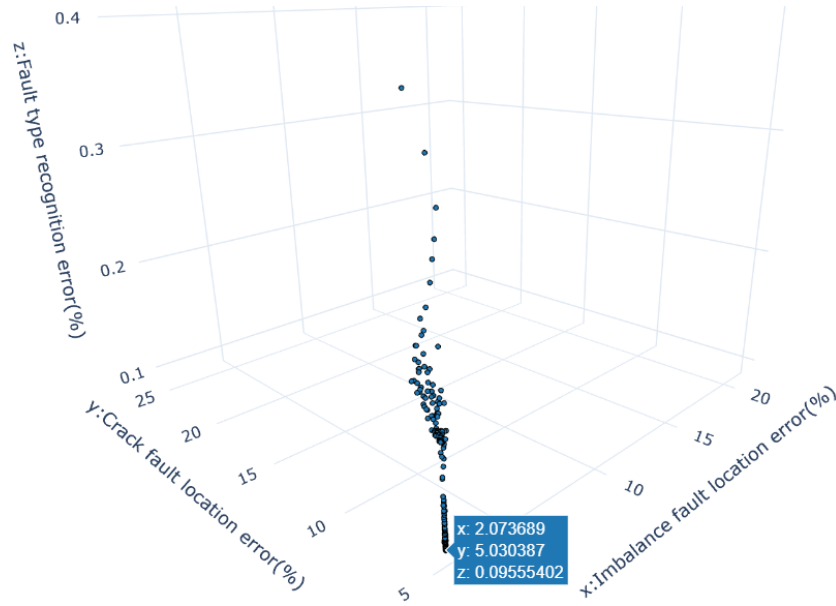
The support stiffness and damping come from the coupling of the bearing and the support structure. In practice, it is difficult to obtain their exact values but can be adjusted during the training by the “variation matrices” presented in

Fig. 3.12.

## C.5 Multi-task supervised training

The proposed model utilizes a multi-task learning framework optimized through a composite loss function as defined in Eq. (C.6). The components of the loss function include  $L_t$ ,  $L_m$ , and  $L_s$ , which correspond to fault type recognition, unbalance fault location error, and shaft crack fault location error, respectively. Additionally,  $L_v$  quantifies the accuracy of reconstructing temporal vibration features. The respective loss functions are binary cross entropy for  $L_t$  and mean square error for  $L_m$ ,  $L_s$ , and  $L_v$ . These components are weighted by  $\alpha_t$ ,  $\alpha_m$ ,  $\alpha_s$ , and  $\alpha_v$  respectively, as shown in the following equation:

$$L_o = \alpha_t L_t + \alpha_m L_m + \alpha_s L_s + \alpha_v L_v \quad (\text{C.6})$$



**Figure C.9:** Optimized models selected based on fault identification, unbalance localization, and crack localization.

The complexity of multi-task supervised training stems from the need to simultaneously optimize faults recognition, fault localization, and temporal vibration feature analysis. Addressing this requires careful consideration of the following factors:

1. Varying output magnitudes: To mitigate discrepancies in output magnitudes, the loss weights ( $\alpha_t$ ,  $\alpha_m$ , and  $\alpha_s$ ) for fault type and position are set to 100, while  $\alpha_v$  for vibration features is adjusted to 1.
2. Conflicting optimization goals: While the overall diagnostic performance hinges on the accuracy of fault localization and identification ( $L_m$ ,  $L_s$ , and  $L_t$ ), optimizing the global loss function ( $L_o$ ) does not necessarily ensure simultaneous optimization of individual loss components.

To tackle these challenges, the RFEMNN model uses  $L_o$  to steer the training, saving the state every three epochs to a model pool. Training halts if  $L_v$  fails to improve after 50 epochs. The best model is then chosen from the pool based on its overall task performance, as shown in Fig. C.9.

## C.6 RFEMNN's evaluation metrics

The classification metrics, i.e., accuracy ( $Ac$ ), false alarm rate ( $Fa$ ), and missing rate ( $Mis$ ), are calculated by Eq. (C.7). In this equation, the values  $a_{ij}$  and  $r_i$  are given by Table C.4, where the term  $a_{ij}$  represents the number of samples belonging to the state  $i$ , but the proposed model indicates state  $j$ .

**Table C.4:** Confusion matrix for multiple fault diagnostics results.

		<b>diagnostics results</b>			
		<b>Healthy</b>	<b>Unbalance</b>	<b>Crack</b>	<b>Un &amp; Cra</b>
<b>True state</b>	<b>Healthy</b>	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$
	<b>Unbalance</b>	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$
	<b>Crack</b>	$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$
	<b>Un&amp;Cra</b>	$a_{41}$	$a_{42}$	$a_{43}$	$a_{44}$
	<b>Sum:</b>	$r_1 = a_{11} +$ $a_{21} + a_{31} +$ $a_{41}$	$r_2 = a_{12} +$ $a_{22} + a_{32} +$ $a_{42}$	$r_3 = a_{13} +$ $a_{23} + a_{33} +$ $a_{43}$	$r_4 = a_{14} +$ $a_{24} + a_{34} +$ $a_{44}$

$$\begin{cases} Fa = \frac{a_{12}+a_{13}+a_{14}}{a_{11}+a_{12}+a_{13}+a_{14}} \\ Ac = \frac{a_{11}+a_{22}+a_{33}+a_{44}}{\sum_{i=1}^4 \sum_{j=1}^4 a_{ij}} \\ Mis = \frac{a_{21}+a_{31}+a_{41}}{\sum_{i=1}^4 r_i} \end{cases} \quad (C.7)$$

To assess defect localization performance, the mean absolute error is used to evaluate location errors of unbalanced defects ( $L_U$ ) and shaft cracks ( $L_C$ ). The average localization accuracy of all defects is denoted by  $\bar{p}$ . Metric  $T$  is defined in Eq. (C.8), which combines fault localization and type identification accuracy to evaluate overall model

## C.7. Supplementary comparison of RFEMNN and other benchmark models

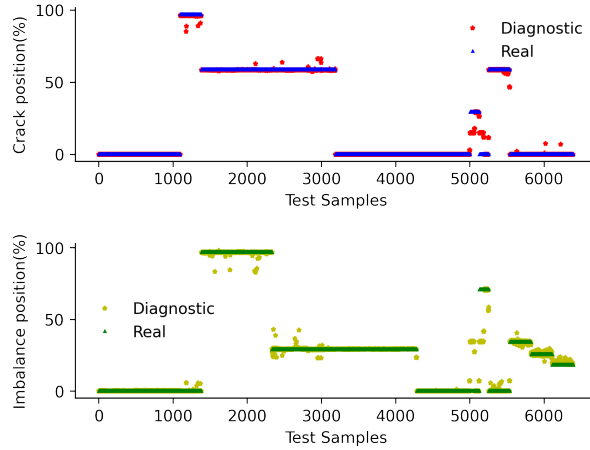
performance.  $L_{U_{predict}}$  and  $L_{C_{predict}}$  represent the mean error of diagnostics fault location, measured as a percentage of shaft length, while  $L_{U_{real}}$  and  $L_{C_{real}}$  indicate actual fault locations measured with the same method. The fault locations along the axial direction are expressed as percentages for four variables: unbalanced fault diagnostics location  $U_{predict}$ , unbalanced fault real location  $U_{real}$ , shaft crack fault diagnostics location  $C_{predict}$ , and shaft crack fault real location  $C_{real}$ .

$$\begin{cases} T = \frac{a_{11}+a_{22}+a_{33}+a_{44}}{\sum_{i=1}^4 r_i} \times \bar{p} \\ L_U = \frac{U_{predict} - U_{real}}{U_{real}} \\ L_C = \frac{C_{predict} - C_{real}}{C_{real}} \end{cases} \quad (C.8)$$

## C.7 Supplementary comparison of RFEMNN and other benchmark models

### C.7.1 Recognition results of the proposed RFEMNN and other SOTA models

Using the proposed RFEMNN framework, the confusion matrices of fault classification results are presented in Table. C.5. Besides, the result of the fault localization is shown in Fig. C.10.



**Figure C.10:** Average results of fault localization with 10 fold-cross validation on diagnostics fault location.

The fault localization results of the RFEMNN in different rotor structures are presented in Fig. C.10, where the horizontal axis represents the number of test samples. Each

### C.7. Supplementary comparison of RFEMNN and other benchmark models

test sample was randomly selected from different experimental data of faults. The vertical axis represents the relative position of the fault along the entire axis length.

**Table C.5: Confusion matrices of RFEMNN faults recognition.**

		<b>Diagnostics results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	1648	167	0	0
	<b>unbalance</b>	100	2262	0	0
	<b>Crack</b>	0	0	411	0
	<b>Im&amp;Cra</b>	0	0	0	1805
	<b>Total:</b>	1748	2429	411	1805

From Table.C.5 and Fig. C.10, one can see that:

1. Considering different rotor structures with multiple rotor-speed-varying processes, the diagnostic results provided by RFEMNN are close to the ground truth on both aspects: fault identification and error location. Particularly, unbalanced faults, shaft cracks, and combined defects are completely distinguished by the proposed RFEMNN (see Table.C.5).
2. All combined defects are detected and identified correctly.
3. There are still some cases of false and missing alarms. In detail, 167 healthy samples are incorrectly misrepresented as faulty ones while 100 unbalanced faulty samples are misidentified as healthy (Table.C.5). However, the false (*Fa*) and missing (*Mis*) rates are small enough. They are 9.2% and 1.56%, respectively.
4. For localization of unbalanced defects and shaft cracks, the results provided by RFEMNN are close to the true ones (see Fig. C.10). Only a small number of samples have significant prediction errors.
5. As can be seen from Fig. C.10, the results of the shaft crack localization are generally better than the ones of unbalanced defects. This can be explained by the fact that the shaft cracks have greater effects on vibration signals than the unbalanced faults, because any rotating system has its initial unbalance.

In this study, four state-of-the-art models used for diagnosing unbalanced and shaft crack defects are reconstructed for comparison with the proposed RFEMNN. These models include: 1) Continuous wavelet transform scalogram assisted CNN (CWTS) [359], 2) Deep Residual Shrinkage Networks (DRSNs) [360], 3) Deep CNN with a support vector machine classifier (semi-DCNN) [361], and 4) Spatio-temporal fusion neural network (STFNN) [362].

However, the above models only consider fault identification without defect localization. To the best of our knowledge, no existing PHM framework in the literature has

### C.7. Supplementary comparison of RFEMNN and other benchmark models

been developed for fault identification and defect localization on rotating shafts. Traditionally, these tasks are handled by distinct ML algorithms, as evidenced in the literature [363, 364]. To facilitate a valid comparison with the proposed model, it is essential to adapt existing models that incorporate defect localization. Therefore, three established benchmark models have been selected for reconstruction and comparison: 1) ANN [364], 2) LSTM [365], and 3) Extreme learning machine (ELM) [363].

In addition to the above benchmark models, a CNN-LSTM model with the same architecture as our proposed RFEMNN but without the customized physical layers, called NO\_RFEM\_NN, is investigated. It is used to prove the importance of embedding physics knowledge into ML.

Eqs. C.7 and C.8 are used to evaluate the multi-fault diagnostics metrics of the proposed RFEMNN and other benchmark models. The results are shown in Table. 3.1.

**Table C.6: Confusion matrices of CNN (raw data as input) [359].**

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	1783	11	0	21
	<b>unbalance</b>	1199	70	0	1094
	<b>Crack</b>	4	12	0	1789
	<b>Im&amp;Cra</b>	0	54	0	1751
	<b>Total:</b>	765	2665	241	2723

**Table C.7: Confusion matrices of CWSCNN [359].**

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	718	1097	0	0
	<b>unbalance</b>	38	1452	0	873
	<b>Crack</b>	9	62	241	99
	<b>Im&amp;Cra</b>	0	54	0	1751
	<b>Total:</b>	765	2665	241	2723

**Table C.8: Confusion matrices of DRSNs (wavelet spectrum as input) [360].**

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	1777	38	0	0
	<b>unbalance</b>	71	2035	52	205
	<b>Crack</b>	0	0	400	11
	<b>Im&amp;Cra</b>	0	80	1	1724
	<b>Total:</b>	1848	2153	453	1940

**Table C.9:** Confusion matrices of DRSNs (raw data as input) [360].

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	1156	650	4	5
	<b>unbalance</b>	170	1917	37	239
	<b>Crack</b>	13	5	347	46
	<b>Im&amp;Cra</b>	0	253	12	1540
	<b>Total:</b>	1339	2825	400	1830

**Table C.10:** faults recognition confusion matrices of CNN-SVM[361].

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	893	572	9	341
	<b>unbalance</b>	418	1079	38	828
	<b>Crack</b>	83	148	48	132
	<b>Im&amp;Cra</b>	90	205	1	1509
	<b>Total:</b>	1482	2004	96	2810

**Table C.11:** faults recognition confusion matrices of STFNN[362].

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	751	346	0	718
	<b>unbalance</b>	184	913	0	1266
	<b>Crack</b>	0	0	0	411
	<b>Im&amp;Cra</b>	0	0	0	1805
	<b>Total:</b>	935	1259	0	4546

**Table C.12:** faults recognition confusion matrices of STFNN.

		<b>Diagnostic results</b>			
		<b>Healthy</b>	<b>unbalance</b>	<b>Crack</b>	<b>Im&amp;Cra</b>
<b>True state</b>	<b>Healthy</b>	1747	64	1	3
	<b>unbalance</b>	1414	307	6	636
	<b>Crack</b>	50	106	74	181
	<b>Im&amp;Cra</b>	2	57	0	1746

## C.7. Supplementary comparison of RFEMNN and other benchmark models

	<b>Total:</b>	3213	534	81	2566
--	---------------	------	-----	----	------

### C.7.2 Fault location results

#### Artificial Neural Networks

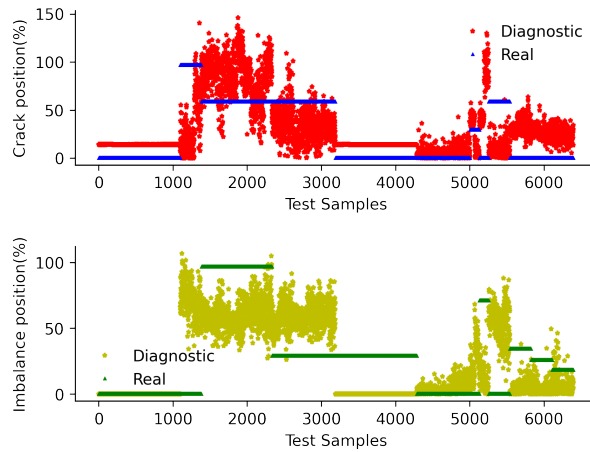


Figure C.11: Average results of ANN-based fault localization with 10 fold-cross validation[364].

#### Long and short-term memory neural networks.

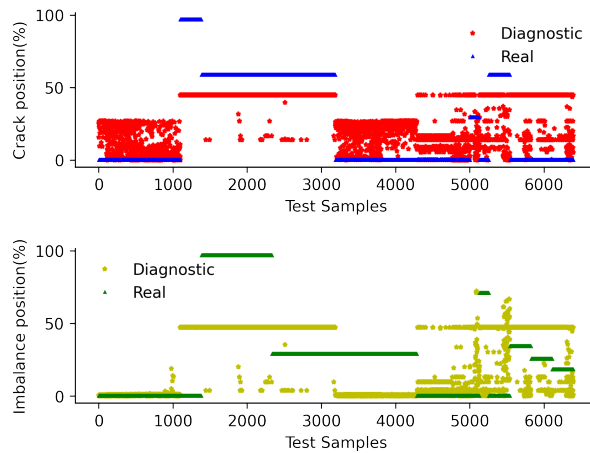
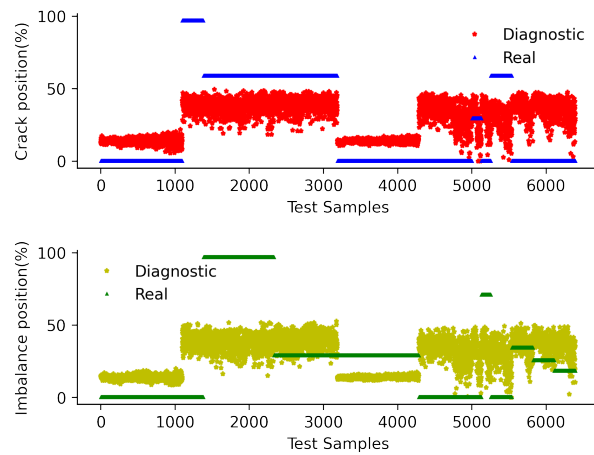


Figure C.12: Average results of LSTM-based fault localization with 10 fold-cross validation[365].

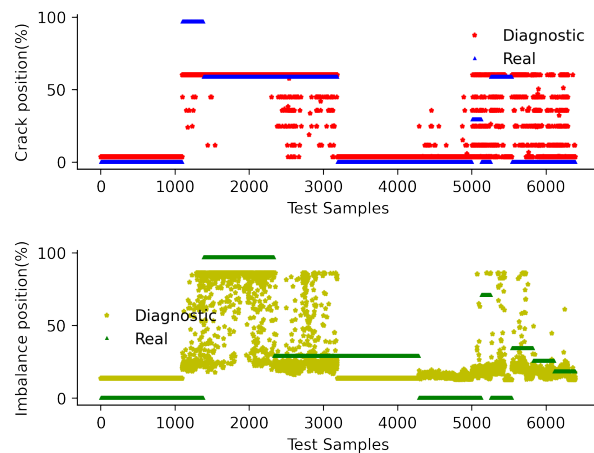
#### Extreme Learning Machine





**Figure C.13:** Average results of ELM-based fault localization with 10 fold-cross validation[363].

#### NO\_FEM\_NN

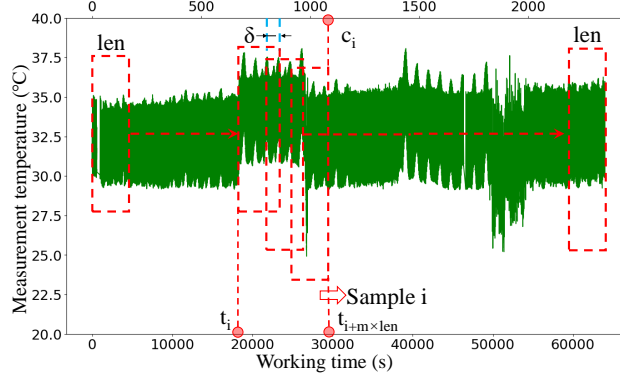


**Figure C.14:** Average results of NO\_FEM\_NN based fault localization with 10 fold-cross validation.

## C.8 SEI-DCN model structure

**End to end input.** Raw sensor data are initially truncated to construct the inputs  $x_{in}$ . The sliding window technique splits continuous time series data into overlapping training samples from the start to the end of the monitoring sequence, as shown in

Fig. C.15. For instance, when truncating temperature measurements, the sliding win-



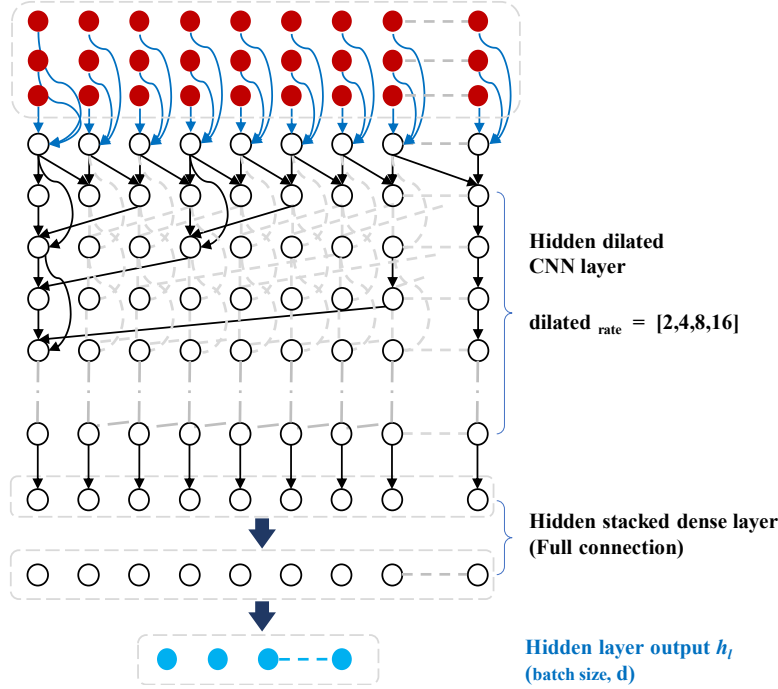
**Figure C.15:** Truncate the original sequence in a fixed-length time window to construct an end-to-end input.

Window length parameter  $m$  determines the number of samples in each window, with each sample having a fixed dimension of  $[len, 1]$ . Another parameter,  $\delta$ , consecutive window, ensure the continuity of the truncated samples. Each training sample gathers data from the period starting at  $t_i + m \times len$ , retaining its 2D-dimensional structure for subsequent channel merging and splitting processes. As a result, each training sample  $x_i$  (where  $i = 1, 2, \dots, n$ ) has a dimension of  $[m, len, 1]$ . The number of discharge cycles remaining  $c_i$  at the time point  $t_i + m \times len$  is recorded as the sample's RUL label. The parameters  $len$  and  $m$  determine the extent of cyclic data covered in a training sample. Smaller values of  $len$  and  $m$  can enhance the proposed model's practical value by increasing the data's granularity and thereby improving the model's prediction accuracy.

It is worth noting that within each truncated data segment, we have used a fixed length of time rather than a fixed number of cycles, as pinpointing the exact timing of the 3-4 charge/discharge cycles is difficult and results in a variable length of the input sequence segments. The window length  $len$  roughly contains the information of 3-4 complete charge/discharge cycles. Since this is an end-to-end model for online prediction in real-world applications, "sample  $i$ " in Fig. C.15 may start or end at any point within a discharge or charge cycle. Although the specific start and end points within a cycle can vary, the fixed time length ensures that the data spans approximately 3-4 cycles. This model's predictive ability does not rely on complete cycle granularity, making it more practical for real-world applications.

**End to end data-driven branch.** The data-driven branch employs an end-to-end (E2E) deep learning (DL) paradigm to acquire degradation-related features for the battery RUL prediction from raw data directly. This approach eliminates the need for intricate signal processing or expert domain intervention.

The input comprises three parallel channels: temperature, voltage, and current. The samples within each channel are partitioned into multiple batches. In each learning



**Figure C.16:** Schematic diagram of D-CNN with a data flow.

batch, these independent physics measurements are combined through “Concatenate” layers to construct the inputs  $X_{in}$  with the dimension  $[m, len, 3]$ .

A Dilated convolution Neural Network (D-CNN) is shown in Fig. C.16 to demonstrate the application of the E2E paradigm above. D-CNN has a series of dilated convolution layers with increasing dilation factors ( $dilated_{rate}$ ) to encapsulate the interdependences between temperature, voltage, and current concerning the battery’s discharge cycles. By leveraging dilated convolutions, DCN can encompass larger receptive fields of each layer by incorporating gaps between weight connections to build positional jump connections.

Subsequently, the final feature representation is obtained through a dense layer with  $d_1$  neurons, yielding the data-driven feature  $h_l$  (dimension:  $[batch\ size, d_1]$ ).

**Physics informed branch.** The physics-informed branch utilizes the “physics-embedded algorithm structure” paradigm, as detailed in [366], to obtain the physics-consistency features and insights relevant to degradation.

The PI branch utilizes the SEI growth formula proposed by Attia et al. [175] as shown in Eq.(C.9) and (C.10). In Eq. (C.9),  $D$  is the SEI degradation parameter.  $D_0$  represents the initial or baseline SEI degradation parameter.  $E_a$  is the effective activation energy for SEI growth, determined by the Arrhenius relation.  $k_B$  denotes Boltzmann’s constant, a fundamental physical constant.  $T$  is the monitoring temperature of the battery. This expression captures the exponential dependence of SEI degradation, aligning with the Arrhenius equation’s insights into temperature-dependent chemical processes.

$$D = D_0 \exp\left(-\frac{E_a}{k_B T}\right) \quad (\text{C.9})$$

The  $p_{out}$  of is estimated using the equation:

$$p_{out} = \frac{C}{D} + b \quad (\text{C.10})$$

In this context,  $p_{out}$  refers to an approximate lifespan indicator, as it is not the final RUL output of the proposed PIML model. This distinction arises due to the simplified nature of the models, often incorporating empirical parameters and assumptions that do not fully capture the complexities of actual battery operation and degradation phenomena. This formulation emulates the Paris laws in material fatigue and posits that a battery's cycle life is inversely proportional to the rate of SEI degradation, adjusted by the empirical factor  $C$ .

Considering the input-output relationships and the logical relationship between Eq.(C.9) and (C.10), only the temperature data are taken as the input of the PI branch. and the cross-layer connections are used in Fig.C.20. Eq.(C.9) and (C.10) are the activation functions of these custom design layers.

**Merge the two branches' information.** In the output part, the final ‘‘Full connectivity neural network (FCNN)’’ considers the output  $p_{out}$  of the PI branch in the decision-making process and the data insights from feature  $p$ . FCNN is tailored for the many-to-one process to generate the output of the remaining discharging cycles.

The data flow processing of the entire PIML framework can be represented by the following equation. In the data-driven branch, the input tensors  $x_1, \dots, x_n$  undergo processing through  $L$  hidden layers, where each layer progressively extracts higher-level features. This process can be modeled in Eq. (C.11):

$$\mathbf{h}^{[l]} = g^{[l]} \left( \mathbf{W}_h^{[l]} g^{[l-1]} \left( \mathbf{W}_h^{[l-1]} \dots g^{[2]} \left( \mathbf{W}_h^{[2]} g^{[1]} \left( \mathbf{W}_h^{[1]} \mathbf{x} + \mathbf{b}_h^{[1]} \right) + \mathbf{b}_h^{[2]} \right) \dots + \mathbf{b}_h^{[l-1]} \right) + \mathbf{b}_h^{[l]} \right) \quad (\text{C.11})$$

Each layer applies a linear transformation (via the weights  $W^{[l]}$  and biases  $b^{[l]}$ ) followed by a non-linear activation function  $g^{[l]}$ . The final network output is a continuous transformation product from the inputs with dimensions  $[m, len, n]$  to the last hidden layer output features  $h_l$ .

Simultaneously, in  $K$  layers PI-branch, the entire process can be represented as a data flow in the NN through Eq. (C.12):

$$\mathbf{p} = g^{[K]} \left( \mathbf{W}_p^{[K]} F^{[K-1]} \left( \mathbf{W}_p^{[K-1]} \dots g^{[2]} \left( \mathbf{W}_p^{[2]} f^{[1]} \left( \mathbf{W}_p^{[1]} \mathbf{x} + \mathbf{b}_p^{[1]} \right) + \mathbf{b}_p^{[2]} \right) \dots + \mathbf{b}_p^{[K-1]} \right) + \mathbf{b}_p^{[K]} \right) \quad (\text{C.12})$$

Comparing Eq. (C.12) with Eq. (C.11), it's worth noting that the custom architecture in the PIML branch replaces the conventional intermediate hidden layer to represent the physics input-output relationship.

## C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

In summary, the complete proposed PIML model can be represented by Eq. (C.13).

$$y = h_f \left( \overbrace{w_1 s^{[r]} \left( \mathbf{W}_d^{[r]} s^{[r-1]} \left( \mathbf{W}_d^{[r-1]} \dots s^{[2]} \left( \mathbf{W}_d^{[2]} s^{[1]} \left( \underbrace{\mathbf{W}_f^{[p]} p + \mathbf{W}_f^{[h_l]} h_l^{[l]} + \mathbf{b}_f^{[1]}}_{\text{Feature fusion}} \right) + \mathbf{b}_d^{[2]} \right) \dots + \mathbf{b}_d^{[r-2]} \right) \right) + w_2 p_{\text{out}} + \mathbf{b}_d^{[r]} \right)}_{\text{Decision fusion}} \right) \quad (\text{C.13})$$

In this context,  $p$  is defined and elucidated in Eq.(C.12), while  $h_l$  is formulated and expounded upon in Eq.(C.11). The weights  $w_1$  and  $w_2$  pertain to the final output layer of the proposed PIML model, while  $\mathbf{b}^{[r]}$  represents the bias associated with this final output layer.

**The configuration of the PIML model.** Table. C.13 presents the benchmark model’s data processing and optimized learning hyperparameters for reference.

Combining the optimized DCNN architecture as the data-driven branch in the proposed PIML framework, the hyperparameters of the “PI branch” and the “decision network” in Fig. C.20 are detailed in Table. C.14. Additionally, the coefficients utilized in the physics model equations, Eq. C.9 and Eq. C.10, are also provided in Table. C.14 for reference.

## C.9 Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

**Controlled variable comparison test.** A standard DCNN model developed by Hong et al. [176] serves as a benchmark model for investigating the superior performance of the PIML paradigm. The DCNN model has been validated on the MIT-Stanford dataset [176, 177]. In addition to directly referencing the existing models in the literature that has been validated on the MIT-Stanford dataset, we further performed structural parameter optimisation on the DCNN in developed by Hong et al. [176] to construct a DCNN that outperforms the original one. This new DCNN is also used as the model in the data-driven branch of the SEI-DCN proposed in this paper. The key distinction between DCNN in SEI-DCN and benchmark model lies in the presence of a PI branch and the incorporation of “Physics-informed alignment” and “Joint training” throughout the model. Consequently, we set the control variables as detailed in Table. C.15.

In evaluating the performance of ML models, two key metrics are commonly employed: the number of monitoring data cycles required for prediction and the calculation accuracy of the predicted RUL, which are indicated by the predicted mean absolute error (MAE) of the remaining discharging cycles. The number of data cycles required is a critical measure [367] for predicting remaining discharging cycles, making timely decision-making, and achieving efficient resource allocation. In the time span of the sample used

C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

**Table C.13: Data processing and hyperparameters for benchmark model.**

Parameters	Physics meaning / Description	Setting
$len$	The length of time represented by the sliding window	2500(s), approximately 3 4 cycle length.
$\delta$	The overlap area of two windows	2250(s)
<b>Hyperparameters</b>		
Input channels	Number of types of input monitoring quantities	3
Batch size	The number of processed samples in one forward and backward pass	256
Input shape	Input data shape	[Batch size, 1, 2500]
Concatenate <sub>in</sub>	Inputs' concatenation layer output shape	[Batch size, 2500, 3]
Conv1D-Bat <sub>n</sub>	The number of 1D convolution layer with batch normalization after inputs concatenation	3
Kernel size	The number of 1D convolution layer with batch normalization after inputs concatenation	3
Conv1D-Bat <sub>d</sub>	1D convolution layer output shape	[Batch size, 2500, 64]
Dilated <sub>rate</sub>	Jump interval of dilated convolution	[1, 2, 4, 8, 16]
Kernel size	The 1D convolution layer kernel size	3
$K_{num}$	The number of kernels used in 1D convolution layer	64
Padding mode	The specific way in NN in which extra values (usually zeros) are added to the edges of input data	Casual
Flatten	Flattened output shape	(None, 160000)
Stack-Dense 1	3 stacked dense layer as the decision NN in the data-driven branch	$K_{num}$ in each dense layer:[128,64,1]
Activation 1	Activation function for all dense and convolution layers	Relu
Optimizer	Type of optimizer used	Adam(learning rate=0.001)
Validation ratio	How much data is randomly divided in the training set as the validation set	0.2
Loss function	Loss function used for model training	Mean squared error

C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

**Table C.14: Hyperparameters in physics informed branch.**

Parameters	Physics meaning / Description	Setting
Units <sub>1</sub>	Number of neurons corresponding to SEI growth's Arrhenius equation terms	64
Ea	Activation energy parameter in the Arrhenius equation, indicative of SEI growth rate	0.122 eV
kB	Boltzmann constant, relates temperature to energy	$1.380649 \times 10^{-23}$ J/K
D0	Pre-exponential factor in the Arrhenius equation, a rate constant	$1.0 \text{ s}^{-1}$
Extractor	Frozen data-driven branch serves as the feature extractor for the feature fusion and decision fusion.	Inputs: $3 \times$ [Batch size, 1, 2500]
		Output 1, data-driven branch feature: [Batch size, 64]
		Output 2, data-driven branch RUL: [Batch size, 1]
Fusion <sub>d</sub>	Number of features from different branches in concatenate layer	Concatenate layer dimension: [Batch size, 128]
		Feature from data-driven branch: [Batch size, channel 1-64]
		Feature from PI branch: [Batch size, channel 64-128]
Dense-PI	Dense layer preceding the physics-informed (PI) layer	Number of neurons: 64
PI-output	Output layer of the PI branch, predicting the physical informed metrics	Output dimension: [Batch size, 1]
Stack-Dense 1	Three successive dense layers in the decision-making neural network branch	Neurons in each layer: [128,64,1]
Activation 2	Activation function for all dense layers in the decision network	Relu

C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

**Table C.15:** Schematic representation of the differences in SEI-DCN compared to the original DCNN model.

Control variables	Benchmark DCNN	SEI-DCN
Learning strategy	One-time supervised training	The proposed 3 steps training strategy
Network architecture	Multi-Channel Input - Deep Null Convolution - Stacked Fully Connected Layers Output	Adding a parallel PI branch to the DCNN
Parameters scale	20,559,681	20,749,976

to pre-RUL, the less lifecycle involved, the higher the model capacity to make accurate predictions with limited data. Concurrently, the mean absolute error (MAE) quantifies the agreement between the model’s predictions and the actual cycle life, indicating its reliability.

The encoder structure we selected has already been validated on the benchmark dataset as a published paper in *Applied energy* [176], maintaining the same structure, hyperparameters, and sample truncation lengths for consistent comparison. The key difference in our approach compared to the state-of-the-art DCNN is the addition of a PI branch and the application of our proposed multi-step training method. This controlled variable experiment aims to highlight the specific advantages of our method.

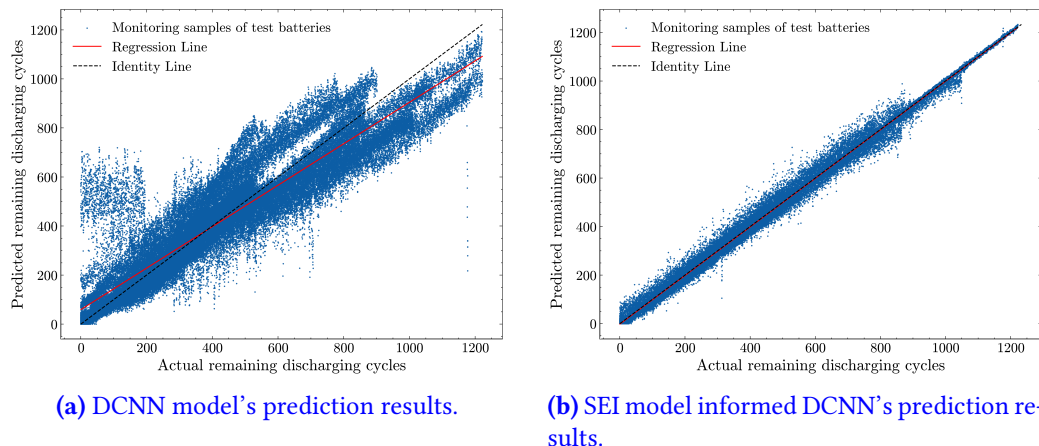
Moreover, our framework is flexible and can accommodate other data-driven models beyond DCNN. Notable alternatives that effectively utilize partial cycle information include the bilateral-branched visual transformer with dilated self-attention or RNN-LSTM. These options are also worth exploring for their potential benefits, as their ability in these “small-early cycles” scenarios is already detailed in references [fei2023deep], and [368].

**Comparison results with solely data-driven branch predictions.** The quantitative comparison is made between the DCNN model alone and the SEI-DCN model, whose data-driven branch is identical to the DCNN model in terms of structure, parameters, inputs, and outputs. The qualitative prediction results are detailed in Fig. C.17, with a detailed quantitative comparison of predictions for each test cell presented in Fig. C.18. Fig. C.17’s “Identity line” represents perfect predictions, where predicted values match true values. The “Regression line” indicates the general trend of predictions. Ideally, the “Regression line” should align with the “Identity line.” The blue scatter points show the actual predictions. Results in Fig. C.17 reveal that both models effectively capture the linear decay trend of remaining discharging cycles. However, our SEI-DCN model outperforms traditional DCNNs by providing more accurate predictions, aligning better with trends, reducing prediction deviation, and with fewer outliers. This improvement



## C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

underscores the value of incorporating physics and tailored learning strategies, providing the enhanced system security prognostic service. To further compare the two mod-

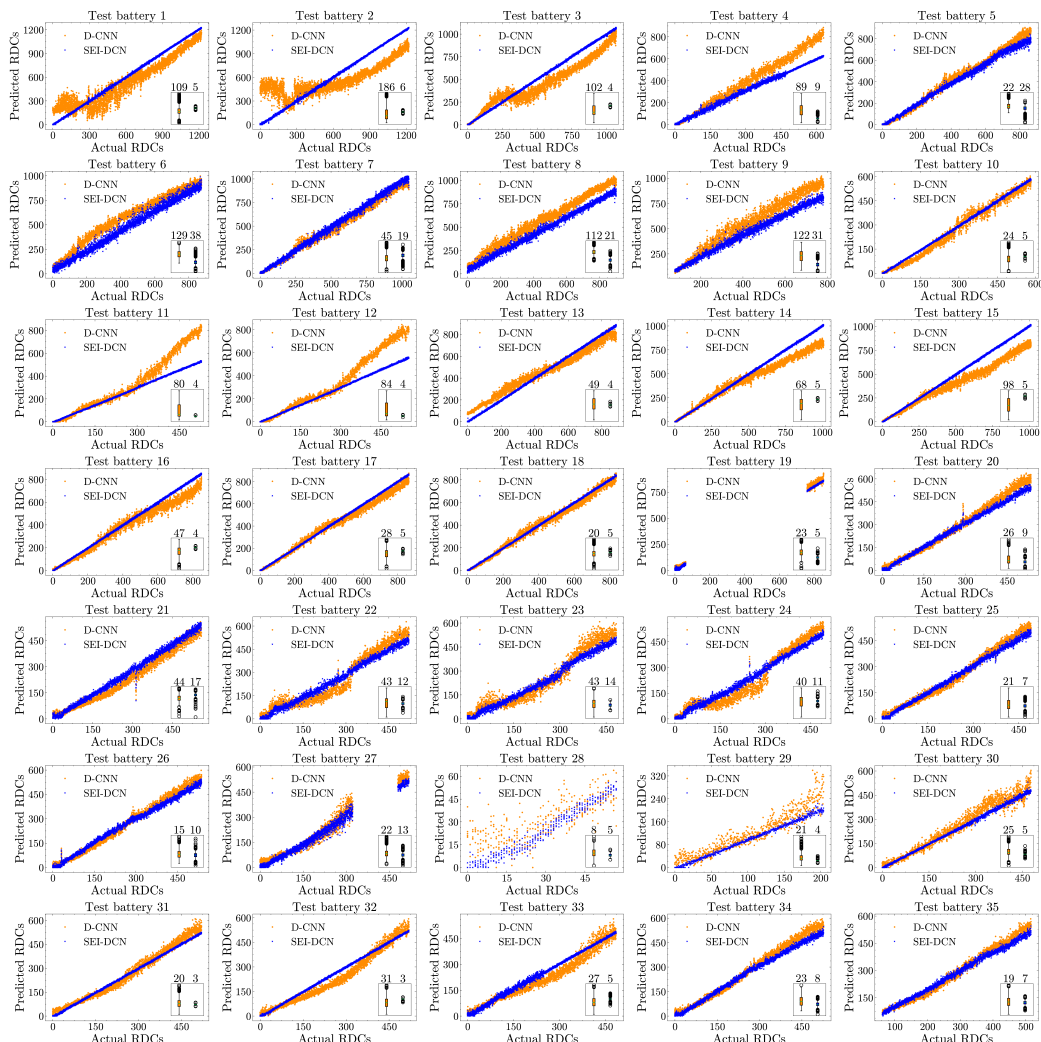


**Figure C.17:** Demonstration of predicted trajectory results for remaining discharging cycles of test battery packs.

els, we utilized box plots to visualize the predictions for each group. Fig. C.18 presents the results of this comparison. Both methods accurately predicted the RUL decreasing trend for batteries with varying lifespans (45 to 1200 remaining charge cycles) despite the measurements missing gaps (batteries 19, 27). However, early in the battery life, the DCNN's predictions are significantly higher than actual values, in contrast to the more accurate SEI-DCN (our approach) predictions for batteries 4, 11, 12, and 29. This accuracy makes SEI-DCN more suitable for early-life task planning in fast-charging batteries. Towards the end of the battery life, with RUL nearing zero, DCNN's predictions remained overly optimistic compared to actual values and SEI-DCN predictions for batteries 1, 2, 23, and 28. This suggests that SEI-DCN provides more reliable information for safely operating fast-charging batteries in their late degradation stages. Moreover, a comparison of overall prediction errors revealed that SEI-DCN had a narrower error distribution with fewer outliers, indicating its greater robustness and reliability over the purely data-driven DCNN.

In test cell 5, SEI-DCN slightly underperformed compared to DCNN, with a predicted MAE of 28 versus 22, even though the upper and lower quartiles of SEI-DCN's box plots are closer. This demonstrates that SEI-DCN has more predicted outliers smaller than the true value for cell 5, whereas the predicted outliers for the DCNN are all larger than the true value. Combined with the overall prediction results, we argue that the predictions of the SEI-DCN are biased towards conservatism by the constraints of knowledge of physics. Moreover, in cells 6 to 9 and 17, SEI-DCN has worse performance with an MAE higher than 15 compared to its performance in other cells. This suggests the potential for further model improvements and adjustments to enhance its generalization across different cells by finding more representative physics and applying long-term learning.

## C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models



**Figure C.18:** Box plots the prediction errors of remaining discharging cycles (RDCs) for DCNN and SEI-DCN (our approach).

The x-axis of each subplot represents the true RUL, while the y-axis represents the predicted RUL. The embedded boxplot of each subplot has boxes on the left representing the prediction error of DCNN as a benchmark model and boxes on the right representing the prediction error of SEI-DCN as a PIML model. The number above the boxplot is the predicted MAE.

## C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

**Decision mechanistic exploration in learning process.** In a hybrid model that combines data-driven and physics-informed branches, efficiently exploiting useful features from both branches is essential. We focus on the proposed PIML model’s treatment of fused features  $[h_l, p]$  within the concatenate layer with dimensions [Batch size, 128]. This layer effectively merges data-driven features spanning channels 1 to 64 with PI features spanning channels 64 to 128. We plotted a 3D schematic of the channel weights to illustrate the integration process, as shown in Fig. C.19.

We transformed it from a cartesian coordinate system to a cylindrical coordinate system. which unveils the relative significance assigned to the features from each branch. This visualization is a valuable tool for understanding the rationale behind the prioritization of certain features and the consequential impact of their weights on prediction outcomes. The weights distribution map generated after the two-step training process reveals a dense matrix of weights across numerous units in the dense layer. This indicates a comprehensive consideration of features across various channels during the model’s decision-making process.

When the model is trained in three steps, Fig. C.19(b) shows a sparser distribution of weights with more pronounced extremes. This suggests that after completing the three-step joint training process, the model learns to assign more pronounced weights, to specific hidden layer features. This is a desirable result in many machine learning tasks, as it is often associated with a more refined and potentially more generalized model. At this point, the weights for PI branch have decreased in magnitude. We believe that the knowledge from both branches has been unified and integrated into a more expressive data-driven branch. Indeed the PI branch following the second step can be seen as a regularization, helping the data-driven branch in the final step to find a way out of a local optimum and converge closer to the global optimum.

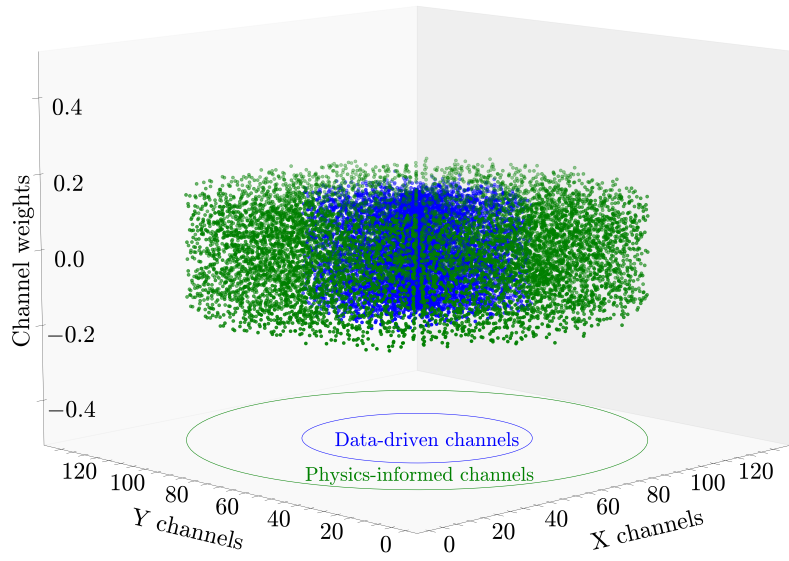
**Evaluating the flexibility of the PI branch for knowledge switching.** To evaluate the flexibility of PI branching in knowledge switching, this study changes the knowledge model embedded in the PI branch while keeping the data-driven branch intact, using a new empirical SEI model [369] to constrain the output relationship of the PI branch. The new empirical formula elucidates the relationship between the thickness  $L(t)$  of the SEI layer and time  $t$ . It provides the following Eq. (C.14):

$$L(t) = \sqrt{\frac{2D_s C_s^0 \epsilon t}{c_P}} \quad (\text{C.14})$$

The entire SEI-informed DCNN model is shown in Fig. C.20.

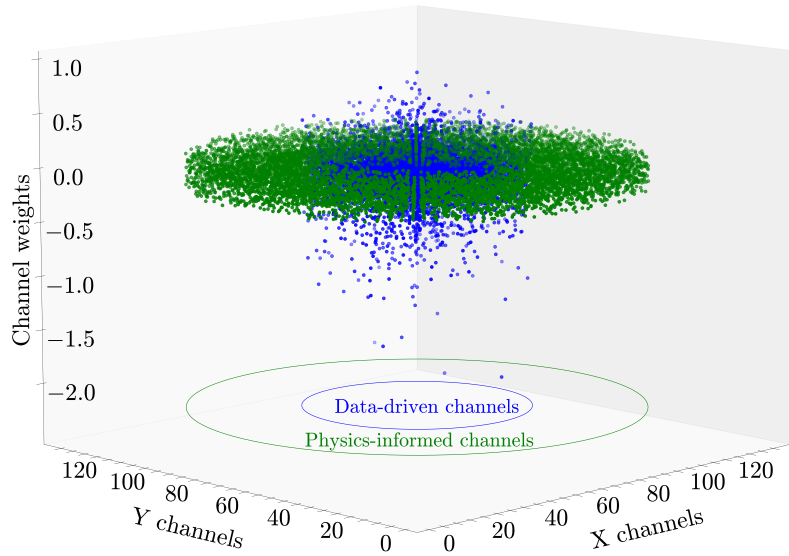
The model in Fig. C.20 is denoted as “PIML model 1”, the model in Fig. C.20 is denoted as “PIML model 2” Their comparison results are shown in Table. C.16.

Table. C.16 summarizes the performance metrics for two PIML models, each designed with distinct SEI growth knowledge. Within our proposed parallel framework, it becomes evident that models presented in Eq.(C.9) and Eq.(C.10) outperform the model described in Eq. (C.14) after two steps. This suggests that PIML model 1 and its incorpo-



(a) Channel weights distribution.

The channel weights of the “Physical alignment” pre-trained data-driven model in processing the integrated features  $[h_l, p]$ .

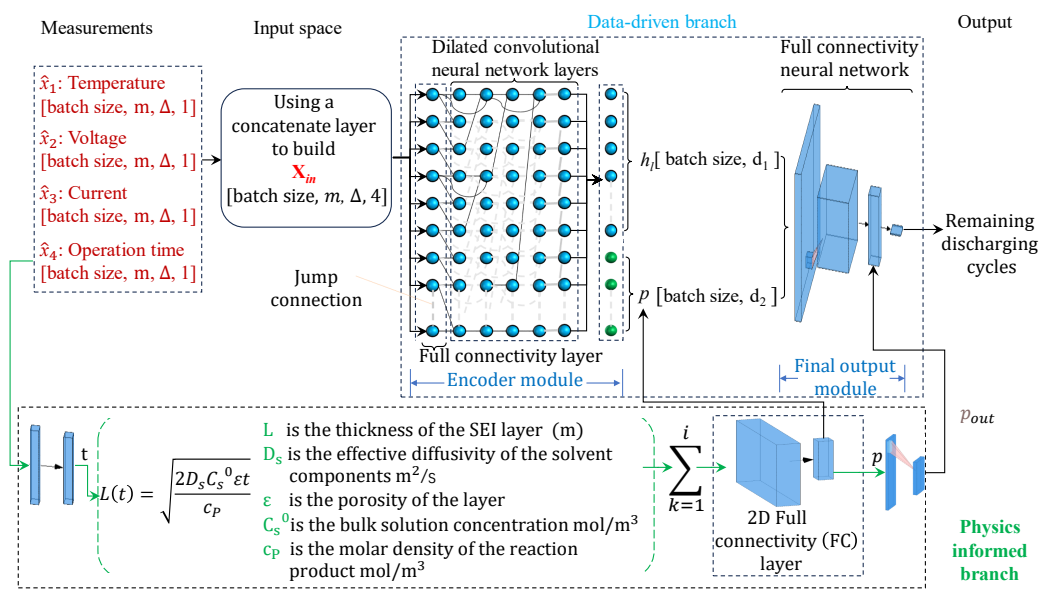


(b) Channel weights distribution in the “Joint training” pre-trained PIML model for processing the integrated features  $[h_l, p]$ .

**Figure C.19:** Schematic depiction of the modulation in channel weights after integrating features across various branches during the multi-step learning phases.

The model above has been trained in the second step, and the model below has been trained in the full three steps.

## C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models



**Figure C.20:** SEI layer thickness growth model informed DCNN functional module integration diagram.

ration of physics exhibit superiority within the same data-driven branch and informed framework.

**Table C.16:** Performance metrics of different PIML models.

Model	Mean absolute percentage error (%)		Cycle Error (MAE)	
	After “Physics-informed alignment”	After “Joint training”	After “Physics-informed alignment”	After “Joint training”
PIML 1	10.9	8.4	15	11
PIML 2	12.7	8.3	17	9

Another conclusion that can be drawn is that the three-step training process outperforms the two-step training process for both of the proposed PIML models. Specifically, the MAPE of “PIML model 1” improves from 10.9% to 8.4%, and the cycle error is reduced from 15 to 11. Similarly, the MAPE of PIML Model 2 improved from 12.7% to 8.3%, and the cycle error was reduced from 17 to 9. Even though the overall MAPE declined by only a limited number of percentage points, the actual forecast cyclic error was reduced by a very large amount. These improvements suggest that the additional training step effectively enhances the model’s accuracy and ensures the model’s flexibility in embedding different physics knowledge. This optimization enables improved generalization and more accurate predictions. This highlights that even if the initial knowledge is sub-

### C.9. Supplementary controlled variable comparison test results of the SEI-DCN and other SOTA models

---

optimal in that it is not directly generating certain RUL-related metrics, but rather an empirical understanding of changes in system behaviour (e.g., SEI thickness growth), it can be refined through this complementary training to achieve promised performance.

# Appendix: Improved Self Supervised Learning strategy for Prognostics and Health Management

## D.1 Novel contrastive metrics for SSL

**Table D.1:** Common Contrast Loss Functions.

Loss Function	Formula	Description
<b>Cosine Similarity Loss</b>	$-\cos(f(x_{\text{pos}}), f(x_{\text{neg}}))$	Measures angular difference between positive and negative features. Ignores magnitude.
<b>NT-Xent Loss</b>	$-\log \left( \frac{\exp\left(\frac{\text{sim}(f(x_{\text{pos}}), f(x_{\text{anchor}}))}{\tau}\right)}{\sum \exp\left(\frac{\text{sim}(f(x_{\text{neg}}), f(x_{\text{anchor}}))}{\tau}\right)} \right)$	Distinguishes positive and negative pairs using a similarity function and temperature $\tau$ .
<b>Triplet Loss</b>	$\max(0, d(f(x_{\text{anchor}}), f(x_{\text{pos}})) - d(f(x_{\text{anchor}}), f(x_{\text{neg}})) + \text{margin})$	Enforces a margin between positive and negative pairs based on their distance $d$ .
<b>CPCL</b>	$-\log \left( \frac{\exp(f(x_{t+k})^T W_k f(c_t))}{\sum \exp(f(x_{\text{neg}})^T W_k f(c_t))} \right)$	Captures temporal dependencies in sequential data with context vector $f(c_t)$ and prediction weight $W_k$ .

## D.2 Prognostia bearing dataset and the data processing method for the proposed SSL approach validation

### D.2.1 PRONOSTIA dataset introduction

This section entails the validation of the model’s performance on the “PRONOSTIA Bearing Dataset” which originates from run-to-failure experiments conducted on a research platform as detailed in [178]. Acceleration sensors are placed to capture both horizontal and vertical vibration signals, with a sampling frequency of 25.6 kHz. Data is recorded at 10-second intervals, each recording lasting for 0.1 seconds. The dataset encompasses three distinct operating conditions and comprises six complete degenerate trajectories for training purposes, along with eleven incomplete trajectories designated for testing. Specifically, the testing trajectories are truncated in proximity to the point of impending failure, facilitating the prediction of the RUL at the truncation point. The training dataset comprises data from six bearings undergoing degradation monitoring. Additionally, there are ten bearings in the test set.

After data preprocessing, the number of unlabeled samples obtained for pre-training is 60000. The number of labeled samples obtained for testing is 30277. The number of samples with labeled data for downstream training is 15857

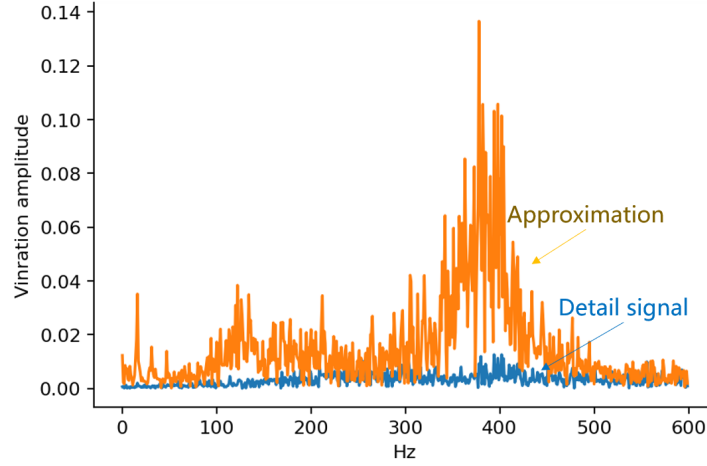
### D.2.2 Data preprocessing setting

This research applies the “db” wavelet transformation to the original data. The obtained approximate and detail coefficient truncated spectrum is shown in Fig. D.1. The truncation length is half of the whole spectrum, and it is used as the input for the CNN with a computation size of (2, 1280).

## D.3 CNN-LSTM benchmark model configuration

In this study, we designed a vanilla model with a focus on both data processing and learning hyperparameters to ensure efficient and accurate predictions as shown in Table. D.2. The data processing pipeline involves truncating the data into windows of 6 samples ( $\Delta t$ ) with an overlap of 5 samples ( $\delta$ ) and a sliding step of 1 ( $s$ ), yielding a total of 2560 data points per window. The data is sampled at a frequency of 25.6 kHz ( $F_s$ ), with a 10-second interval between each sample. For spectral analysis, 64 non-zero frequency components ( $m$ ) were selected to calculate the energy in the frequency domain. In terms of model architecture, we implemented residual shrinking blocks with 128 filters and used convolution kernels of sizes 15, 7, and 5, combined with a stride of 1. The





**Figure D.1:** The result of the discrete wavelet transformation.

model incorporates LSTM layers with 128, 64, and 32 units, followed by dense layers with 32 and 16 units, and applies a dropout rate of 0.2 to prevent overfitting. The model is trained using the Adam optimizer with an initial learning rate of 0.001, a decay rate of 0.96, and decay steps of 10,000. The loss function used is MAE. Furthermore, resampling of both training and testing data was performed to reduce sequence information variance, ensuring robustness in the model’s performance.

## D.4 Qualitative prediction of the degradation trajectory

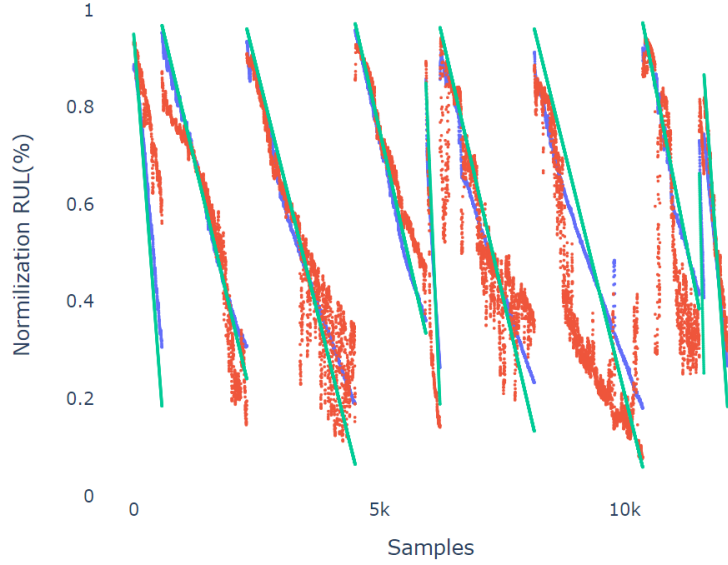
Fig. D.2 presents the predicted RUL values of the SSL and benchmark models. The horizontal axis shows the number of segments after the sliding window, which varies based on each bearing degradation trajectory length. The vertical axis represents the normalized RUL in percentage. The results indicate that SSL performs better and is closer to the actual RUL values. In contrast, the benchmark model has a high rate of prediction errors in RUL.

## D.5 Deep enhancement of the contrastive SSL

The proposed SSL approach, depicted in Fig. D.3, involves pre-training a Siamese feature extractor  $f$ , using a pretext task and a novel contrastive loss function. This function leverages sequence order information to create contrastive sample pairs from times

**Table D.2:** Data processing and learning hyperparameters for benchmark model.

Parameters	Physics meaning / Description	Value
$\Delta t$	The number of samples in one truncated window	6
$s$	The slid steps in data truncation	1
$\delta$	The overlap area of two windows	5
Num	The number of data points in one truncated window	2560 points
Interval	Sampling interval between two samples	10s
$F_s$	Sampling frequency	25.6 KHz
$m$	Number of non-zero frequency components of the frequency domain used to calculate energy, selected from scratch in the spectrum	64
$gap_t$	The number of samples separated between the short sequences $x_{t_1}$ and $x_{t_2}$	6
<b>Hyperparameters</b>		
Filters	Number of filters in residual shrinking blocks	128
Kernel size	Kernel sizes used in residual shrinking blocks	15, 7, 5
Stride	Stride used in residual shrinking blocks	1
LSTM units	Units in LSTM layers	128, 64, 32
Dense layer units	Units in Dense layers	32, 16
Dropout rate	Dropout rate in the prediction layer	0.2
Initial learning rate	Initial learning rate for the optimizer	0.001
Resampling times for 6 training trajectories	Construction of SSL samples to reduce sequence information abundance variance	[3,5,2,4,3,1]
Resampling times for 10 testing trajectories	Construction of SSL samples to reduce sequence information abundance variance	[3,1,1,1,5,1,1,1,10,3]
Learning rate decay rate	Decay rate for learning rate	0.96
Learning rate decay steps	Decay steps for learning rate	10000
Optimizer	Type of optimizer used	Adam
Loss function	Loss function used for model training	Mean Absolute Error



**Figure D.2:** RUL predictions on 10 different degradation trajectories of bearings.

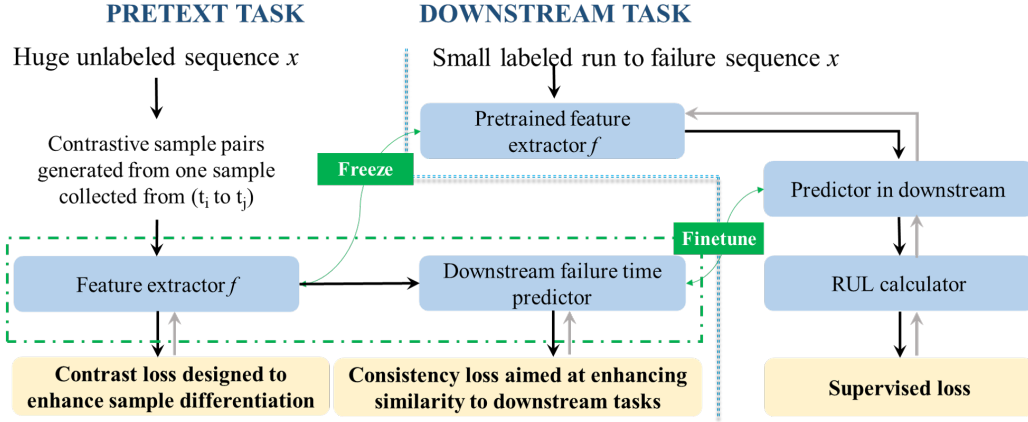
Green, red, and blue lines represent the real RUL, prediction values of the benchmark model, and the ones of the SSL model, respectively.

$t_i$  to  $t_j$ . A custom loss function ensures consistency with the downstream RUL prediction task by enforcing equality of failure times. After pre-training,  $f$ 's parameters are frozen for downstream tasks, preserving the feature representation. A predictor module is fine-tuned with a consistency loss function to align features with task-specific needs. The predictor's output informs an RUL calculator, evaluated by a supervised loss, with an iterative refinement loop for continuous improvement.

**Pretraining phase improvement** Fig. D.4 illustrates the pretraining workflow of the proposed SSL paradigm. Initially, raw data undergoes preprocessing to extract feature sequences and timestamps. This preprocessed data is then utilized to generate contrastive input pairs through a sequential reverse operation. These pairs are fed into a shared feature extractor  $f(\cdot)$  during the pretext task to learn discriminative features (Section D.5), guided by a pretraining contrastive loss  $L_1$ . Concurrently, a failure time predictor models the failure times as part of the pretext task (Section D.5), with performance assessed by a downstream consistency loss  $L_2$ .

**Contrastive input pairs.** A neural network-based operation is employed for temporal axis reversal to build contrastive input pairs (positive and negative sequences) to enhance the comprehension of temporal patterns within input data. This reversal serves the generation of sequential-order contrastive sample pairs.

The construction of traditional contrastive pairs often relies on extensive data aug-



**Figure D.3:** Innovation points of the proposed new SSL learning paradigm.

The blue dotted line represents the dividing line between upstream and downstream tasks.

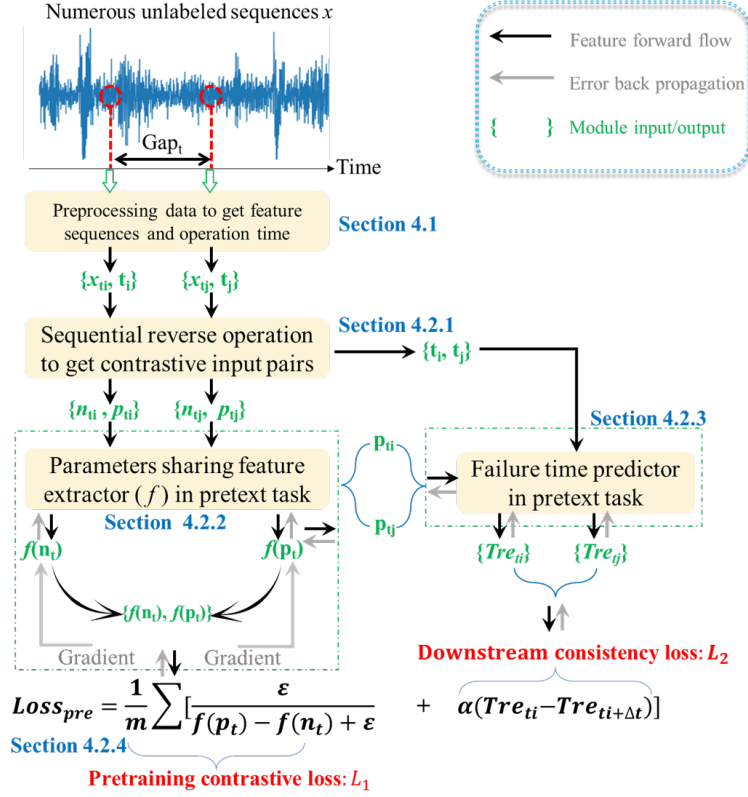
mentation or the generation of new data samples, which can be computationally costly and risk overfitting. In contrast, as shown in Fig. D.5, temporal axis reversal is a lightweight technique applied directly to the input tensor, efficiently utilizing the existing dataset without the need for additional sample generation.

**Feature extractor.** The feature extractor  $f$  processes the frequency feature sequence as input, extracting hidden patterns  $f(\cdot)$ , which are subsequently used as input for the failure time predictor.

As shown in Fig. D.6, this paper proposes to build a Siamese structure for contrastive SSL by repeatedly employing the feature extractor  $f$  to process the frequency features from different time points. For example, sample  $x_{t_i}$  and  $x_{t_j}$  represent sequences truncated by two sliding windows with a time interval  $gap_t$ . After data pre-processing and applying the “Time dimension reverse” method illustrated in Figure D.5, we obtain temporal frequency features input pairs  $p_{t_i}, n_{t_i}$  and  $p_{t_j}, n_{t_j}$ , corresponding to  $x_{t_i}$  and  $x_{t_j}$ . In the pretext task, the inputs of  $f$  are the contrastive input pairs  $p_{t_i}$  and  $n_{t_i}$  while its outputs are the hidden patterns  $f(p_{t_i})$  and  $f(n_{t_i})$ . For samples from  $t_j$ , the operation is the same. **A specific implementation is to directly reuse the output features  $f(p_{t_j})$  and  $f(n_{t_j})$  without re-learning conditions.** Note that the behavior of  $f$  is constrained by the learning objective that will be defined later.

**Failure time predictor.** Failure time predictor takes the hidden pattern  $f(\cdot)$  and operation time  $t$  as inputs, producing the failure time  $Tre$  used as input for the RUL calculator.

As shown in Fig. D.7, the Failure time predictor is also utilized again in the pretext task. It generates pseudo-failure thresholds  $Tre_{t_i}, Tre_{t_j}$  based on the input pairs  $f(p_{t_i}),$



**Figure D.4:** Novel SSL workflow for implementation of the proposed solution in pretext task's framework.

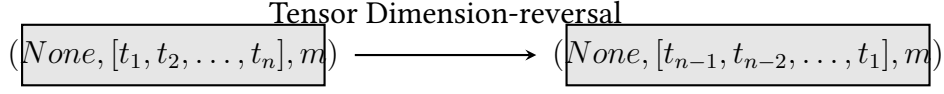
$t_i$  and  $f(p_{t_j})$ ,  $t_j$ . These thresholds, denoted as  $Tre_{t_i}$ ,  $Tre_{t_j}$ , are optimized concerning the consistency loss that will be outlined in next paragraph.

**Consistency loss layer.** A custom-designed pre-trained loss layer is employed to constrain and guide the learning of the feature extractor and failure time predictor in pretraining. This custom loss function, shown in Eq. (D.1), acts as an activation function. The custom-designed loss layer's inputs are  $p_{t_1}$ ,  $n_{t_1}$ ,  $p_{t_i+gap_t}$ ,  $n_{t_i+gap_t}$ ,  $Tre_{t_i}$ , and  $Tre_{t_i+gap_t}$  while its output is  $\mathcal{L}_p$ .

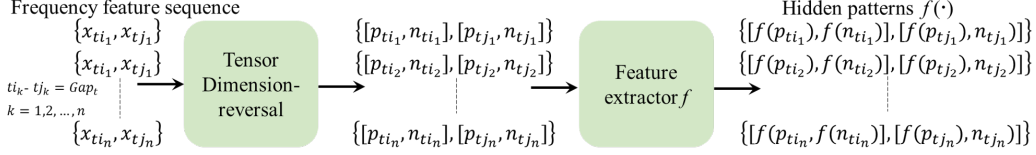
$$L_p = \frac{1}{n} \left\| \underbrace{\sum_{k=1}^n \left( \frac{1 + \epsilon}{(f(p_{t_k}) - f(n_{t_k}) + \epsilon)} + \frac{1 + \epsilon}{(f(p_{t_k+gap_t}) - f(n_{t_k+gap_t}) + \epsilon)} \right)}_{L_1} + \underbrace{\alpha \sum_{i=1}^m (Tre_{t_k} - Tre_{t_k+gap_t})}_{L_2} \right\|_1 \quad (D.1)$$

The overall pre-training loss  $\mathcal{L}_p$  is the sum of  $L_1$  and  $L_2$ . A small positive constant  $\epsilon$  prevents division by zero.  $\alpha$  is the loss weight which is set to 0.5 in this study.

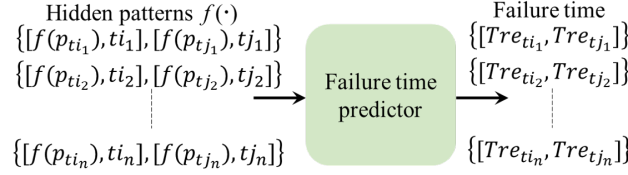
## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses



**Figure D.5:** Inversion of the input temporal feature dimension to generate the contrastive tensor sample.



**Figure D.6:** Basic structure of feature extractor.



**Figure D.7:** Basic structure of failure time predictor.

**Finetuning phase improvement.** When applying the pre-trained model downstream, the architecture of the feature extractor  $f$  maintains consistency of structural parameters in pre-training and downstream tasks. In the downstream task, the pre-trained weights of  $f$  are frozen. The input of the feature extractor is only the positive sequence order features  $p_t$ .

In the downstream task, the pre-trained weights of the failure time predictor are loaded and fine-tuned using labeled data. The output of this fine-tuned model corresponds to the prediction of failure time,  $Tre_t$ , which serves as the input to the RUL calculator layer.

## D.6 Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

### D.6.1 Qualitative results

As the RUL approaches its end, prediction deviations increase, which is critical in practical applications. To understand the reasons, we reviewed the experimental conditions of the aging test. The following points are proposed:

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

To the test data, the degradation experiment was stopped when the vibration amplitude exceeded 20g, with the bearings allowed to degrade naturally. By the termination point, some bearings were significantly deteriorated while others were only slightly affected. For example, in bearings 1, 5, 9, and 10, the overall vibration amplitude mostly remained at 5-10g until the end of the experiment, indicating that the true point of complete failure was not reached in all cases.

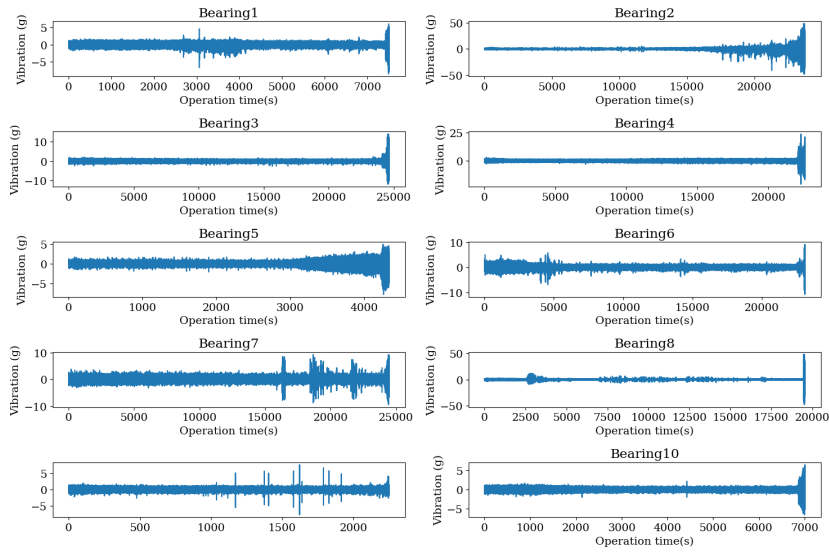


Figure D.8: Horizontal direction vibration.

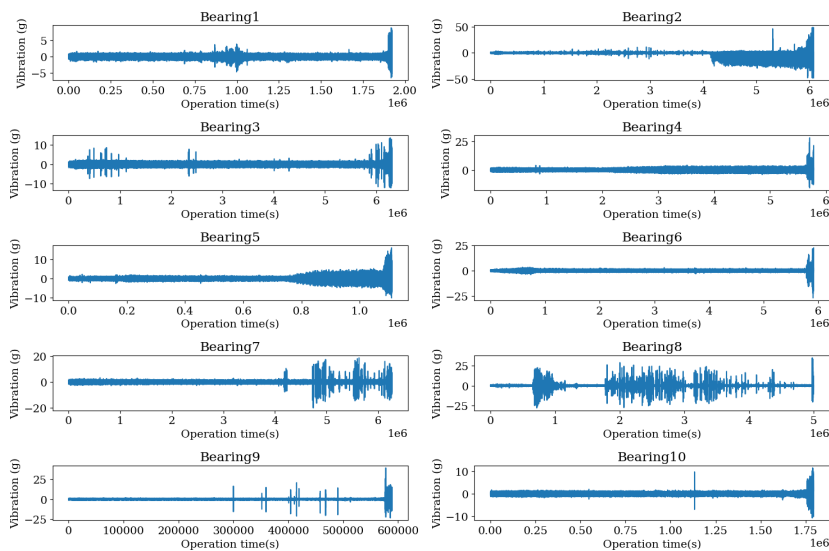


Figure D.9: Vertical direction vibration.

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

---

In these instances, the overload condition of 20g was observed only in limited monitoring directions (e.g., horizontal and vertical vibrations, see Fig. D.8 and D.9), suggesting that the actual point of failure was delayed. This delay results in greater prediction deviations near the end of RUL for these bearings. We believe this is the reason for the observed discrepancies in those specific cases. Ultimately, while there are specific instances of increased deviation, our method’s overall performance remains robust and effective across various scenarios, as reflected in the comprehensive results.

### D.6.2 Uncertainty in predictions

In the traditional inference phase, the dropout layer is usually off (i.e., no dropout operation is performed), using all neurons for prediction. To quantify the epistemic uncertainty in our model’s predictive ability we applied the Monte Carlo Dropout method, keeping the dropout layer active during inference and performing forward propagation  $T = 20$  times. Each forward pass provides a different prediction, and we estimate uncertainty by calculating the variance among these predictions.

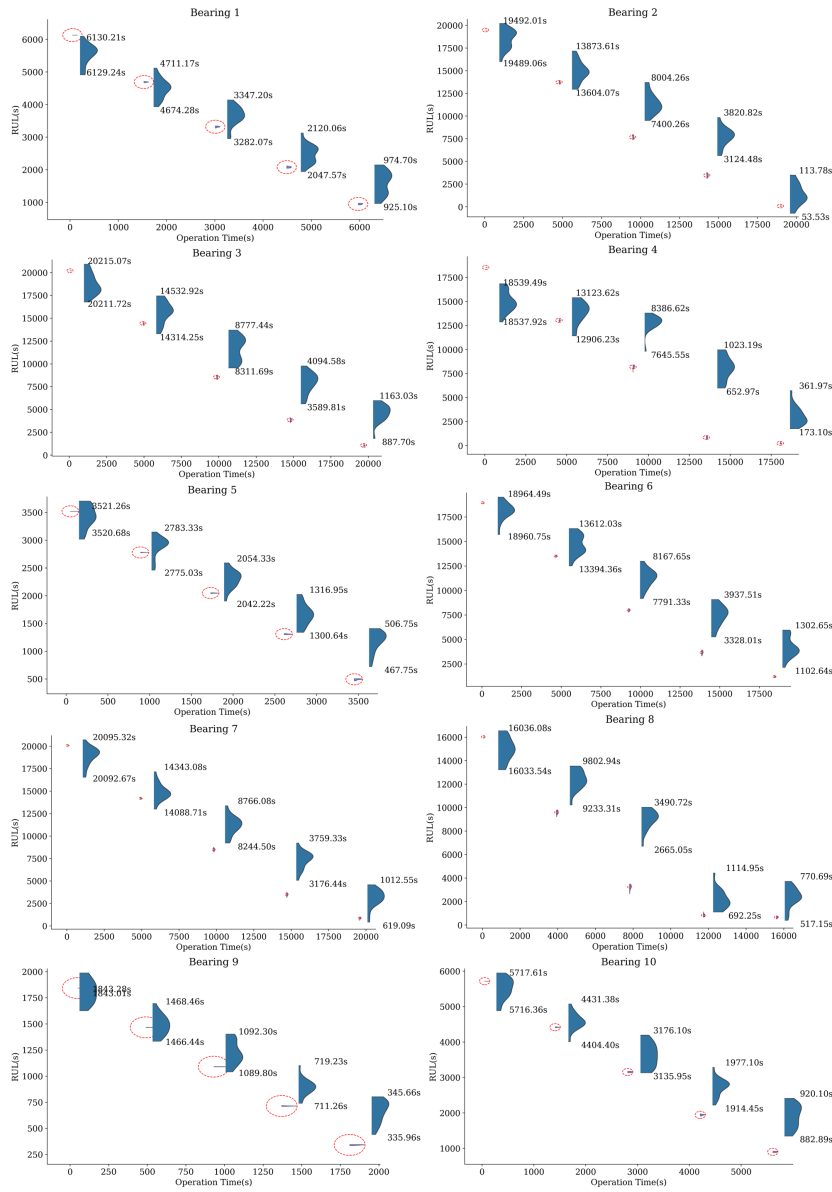
In Fig. D.10, by maintaining an active dropout layer with a dropout rate of 0.3 (30% of neurons will be randomly discarded and only 70% will be activated and used) during inference and conducting 50 forward passes, the model generates a prediction distribution, enabling uncertainty estimation. Across all bearings, predictions typically show a downward trend with increasing operation time. Uncertainty, depicted by the variance and width of the blue shaded area, is generally higher in the middle stages of operation, reflecting greater epistemic uncertainty, and decreases as the bearings approach the end of their useful life. This pattern indicates the model’s higher confidence in predictions at later stages. The varying uncertainty levels across different bearings highlight the critical role of uncertainty estimation in predictive maintenance, providing a measure of confidence in the predictions and aiding in more accurate and reliable decision-making. In addition, bearings 1, 5, 9, and 10 show a smaller distribution of prediction uncertainty, and Fig. D.8 and D.9 show that the overall signal fluctuations of these bearings are smoother and less affected by noise and transient changes, thus reflecting that the signal quality and robustness to transient changes are important for the stable distribution of the model’s prediction results. Notably, the overall standard deviation is reported as 82.63 seconds, offering a deeper insight into the reliability of the predictions.

### D.6.3 Computation cost comparison

In addition, to show the computation cost of the different methods, to calculate the Floating Point Operations (FLOPs) for the given neural network, we did the estimation according to the structure details shown in the SOTA. For example, to the “SSL+GRU” model (SSL SOTA) in [116], we start with the convolutional layers. For the Conv1d1 layer with an input shape of (62, 560), kernel size of 4, and stride of 4, the FLOPs are computed as:  $\text{FLOPs} = 15,640 \times 4 \times 1 \times 2 = 125,120$ . For Conv1d2 with an input shape of (15, 640), kernel size of 2, and stride of 2, the FLOPs are:  $\text{FLOPs} = 7,820 \times 2 \times 1 \times 2 = 31,280$ .



## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses



**Figure D.10:** Prediction uncertainty result of the proposed method in 50 times Monte Carlo tests.

The X-axis represents run time, and the Y-axis represents RUL. The prediction trajectory is divided into six equal segments, and the distribution of predictive uncertainty is shown at five points along this trajectory, from the start of health to the end of the trial. The true uncertainty scale is indicated within the red dashed line. Since the uncertainty values are very small compared to the overall RUL scale, they are magnified in the zoomed-in plots to the right of each red dashed box. These plots show the average RUL prediction, with the blue shaded area representing the 95% confidence intervals (CIs).

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

For Conv1d3 with an input shape of (7, 820), kernel size of 1, and stride of 2, the FLOPs are:  $\text{FLOPs} = 3,910 \times 1 \times 1 \times 2 = 7,820$ . Moving to the GRU layer with a hidden size of 50, the FLOPs are calculated as:  $\text{FLOPs} = 6 \times 50 \times (50 + 50) = 30,000$ . For the fully connected layers, this SSL SOTA model’s “FC1” with input size 50 and output size 256 has:  $\text{FLOPs} = 2 \times 50 \times 256 = 25,600$ , “FC2” with input size 256 and output size 512 has:  $\text{FLOPs} = 2 \times 256 \times 512 = 262,144$ , and “FC3” with input size 512 and output size 2560 has:  $\text{FLOPs} = 2 \times 512 \times 2560 = 2,621,440$ . Summing these values, the total FLOPs for the network is approximate: 3,103,404. Also, it uses the whole model for both training and fine-tuning, so its overall flops are twice this value. The results are shown in detail in Table. D.3.

**Table D.3: Comparison of the computation costs of different methods.**

Methods	Flops
<b>SAE+GRU [370]</b>	2,184,420
<b>SSL+GRU(SOTA) [116]</b>	3,103,404, increased flops due to SSL:3,103,404
<b>CNN-LSTM(Benchmark model)</b>	4,279,840
BiLstm+Attention [371]	No information
<b>CNN-LSTM with contrastive SSL (No Downstream Info)</b>	4,861,442
<b>Proposed model</b>	6,142,040, increased flops due to SSL:16,627,296

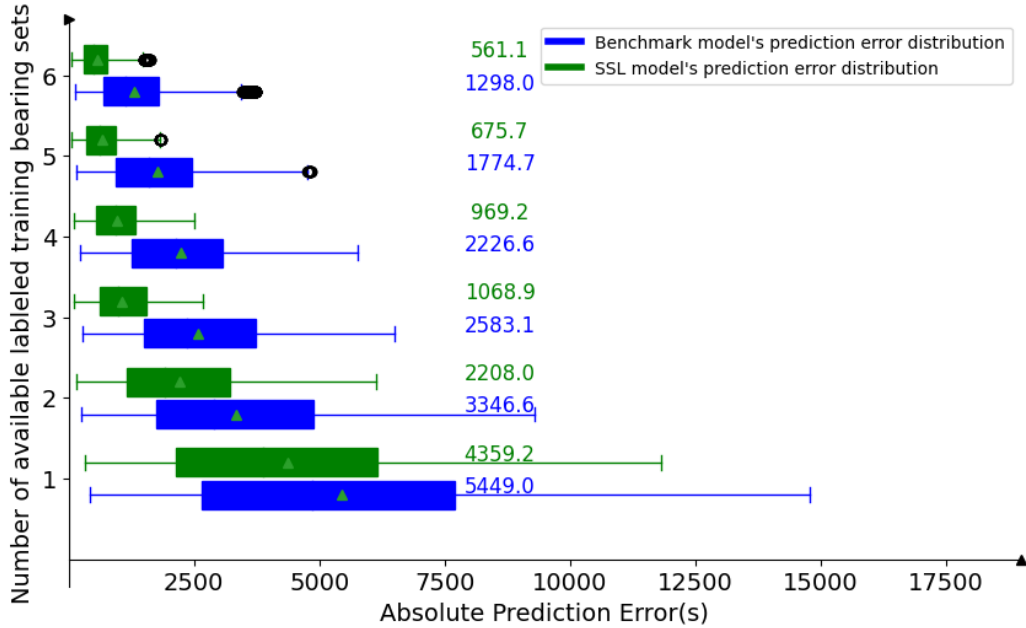
The analysis of Table. D.3 and the performance metrics of various models reveals that while the incorporation of Self-SSL significantly increases the computational cost, as seen with the proposed model’s FLOPs rising to 16,627,296, this added computational expense is justified by the substantial improvements in performance. The proposed SSL model consistently outperforms other state-of-the-art methods in terms of Mean Absolute Error (MAE) and prediction consistency across all test bearings. For instance, it achieves a markedly lower MAE of 0.0716 for Bearing 1 compared to the benchmark model’s 0.1941. This performance enhancement underscores that the additional computation required for SSL is a necessary trade-off, as it enables effective learning from unlabeled data, leading to more accurate and reliable RUL predictions, which is critical for predictive maintenance applications.

### D.6.4 Impact of labeled data availability on the proposed SSL model

This subsection examines the proposed SSL model’s performance with limited labeled data. Fig. D.11 compares the prediction error distributions of the benchmark and SSL

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

models across various sizes of labeled training datasets. The horizontal axis shows absolute prediction error (0 to 18000 seconds), and the vertical axis represents the number of labeled datasets (1 to 6). Box plots display prediction error distributions, indicating the interquartile range (IQR) of errors, with triangles for medians, whiskers for non-outlier ranges, and individual points for outliers.



**Figure D.11:** Box plots of absolute prediction errors under the impact of different amounts of labeled data.

As shown in Fig. D.11, the SSL model consistently outperforms the benchmark model under varying data availability. The SSL model maintains stable error distribution even when labeled datasets are reduced from six to three, unlike the benchmark model, which shows a sharp increase in prediction error. With six labeled datasets, the SSL model achieves a median prediction error of 561.1 seconds, compared to 12980 seconds for the benchmark model. The SSL model's error distribution is more compact, indicating lower median errors and a tighter error range. However, both models' performance declines significantly with fewer than three labeled datasets, highlighting the need for a minimum dataset size for reliable predictions and suggesting further research into enhancing SSL robustness against data scarcity.

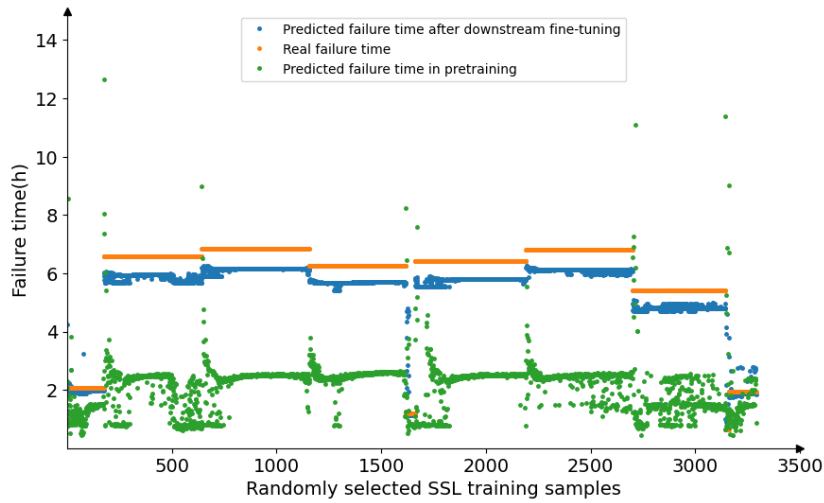
### D.6.5 Impact analysis of model architecture

This subsection examines the model's structural and architectural functionality. It focuses on the learning process and changes in the expressive capacity of its internal modules, thereby supporting the logic behind the proposed pretext design.

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

### Pre-training and downstream task relationships

Eq. (4.4) presents a refined approach for predicting RUL, introducing a pre-training rule that enhances feature representation for subsequent RUL estimation tasks. During pre-training, the model learns to associate temporally distinct samples from the same equipment trajectory with consistent RUL outputs. This step is not about precise value prediction but about setting the foundational understanding of RUL within the model. Consequently, the downstream task focuses on fine-tuning the model’s outputs to align with the precise RUL values, using high-quality labels to achieve accurate predictions.



**Figure D.12:** Comparison of prediction results of failure times under different paths.

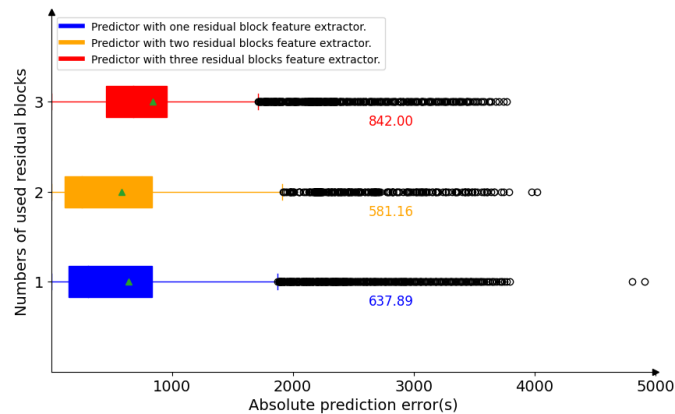
To validate our learning objective, we analyzed the predictive threshold from the “Custom Design Normalized RUL Calculator.” The test sample outcomes are displayed in Figure D.12. Reviewing these results alongside Figure 4.18, which presents the SSL model’s results, we find that most predicted thresholds are below the end-of-life times observed in experiments, except bearings 1, 5, and 10, due to their smaller sample sizes. Notably, there are cases where the predicted RUL is greater than expected based on the experimental end-of-life times. These instances lead us to consider two significant points of discussion:

- The model’s final RUL output is calculated by applying sigmoid normalization. This method introduces a learning bias by converting large RUL values into a bounded range from 0 to 1, thereby introducing an additional non-linearity not present in the original data.
- The RUL labels used for training are normalized and truncated at a predetermined experimental end time rather than reflecting the true time of failure. Consequently, these labels may not represent the actual RUL precisely, leading to a potential bias in the model’s output.

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

Furthermore, the phenomenon depicted in Fig. D.12 illustrates the relationship between pre-training and downstream learning. Pre-training enables the model to learn degradation relationships, establishing their relationship with failure thresholds, indicated by the green dots. Additionally, the threshold calculations in the model pre-trained SSL match the trend of actual results. This shows the benefit of including structures for early predictions in SSL pre-training. In conclusion, *fine-tuning improves the behavior of the predictor and aligns the prediction thresholds more closely. Minimizing the discrepancy between the predicted and actual thresholds, or identifying different alignment criteria during the unlabeled pre-training phase, can boost the effectiveness of the SSL model.*

**Exploring the versatility of model structures.** Some studies observed enhanced performance in each layer of the feature extractor during SSL training [372]. Interestingly, these studies revealed that using the final layer’s output is sub-optimal for the original downstream task. To explore how the distinct network components may excel in different downstream tasks, this section explores the different combinations of the network components. As shown in Fig. 4.15, three residual blocks are stacked together, forming the feature extractor. This study evaluates the performance differences in SSL models when the feature extractor uses only the initial, first, second, or all residual blocks for freezing during downstream fine-tuning.

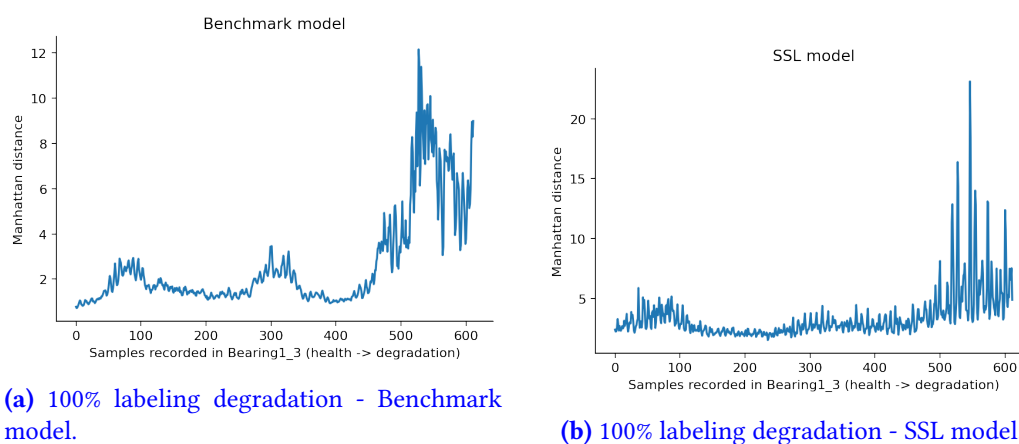


**Figure D.13:** Prediction error when residual blocks of different layers are connected to the predictor.

The results in Fig. D.13 show that the model performs best when only two residual blocks are used in combination with the predictor, with an overall prediction error of 581.16 (s), and with a lower bound on the mean of the error, and a more left-skewed distribution of the overall boxplot.

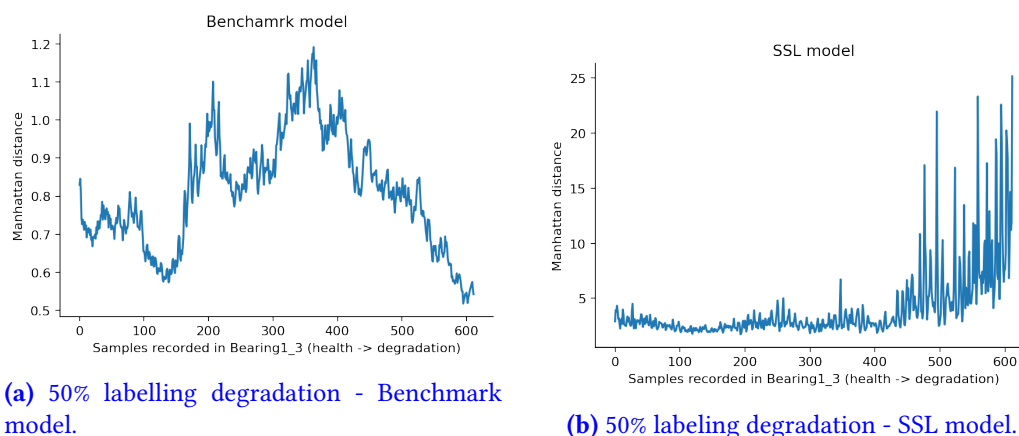
**Decoding the effectiveness of SSL hidden layers without using labels.** In this subsection, we compare the encoding feature results of the benchmark model and

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses



**Figure D.14:** Recognition ability of different models for sequential data under 100% labeling degradation process.

the SSL model at different label percentages. To do this, we train the SSL and benchmark models with varying percentages of labeled data and then truncate the CNN and part of the stacks LSTM to obtain the sequential encoding features. The layer model takes the input of the CNN-LSTM and projects its output into the input of the LSTM part. We can determine the differences in the results by comparing the Manhattan distance of the encoding features when the layer model processes the contrastive pairs. The following results demonstrate these differences. Notably, the SSL model's section is pre-trained, and some layer parameters are frozen. Meanwhile, the benchmark model completely adjusts each layer parameter for different data sets via error back-propagation.



**Figure D.15:** Recognition ability of different models for sequential data under 50% labelling degradation process.

Results in Fig. D.14 show Manhattan distances between sequential degradation seg-

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

---

ments on the vertical axis and the number of trajectory samples on the horizontal axis. Larger distances indicate greater dissimilarity between positive and negative sequences, capturing significant differences. With 100% labelled data, both models exhibit similar sequence recognition performance, revealing that small feature encoding differences appear early but become more pronounced later in the degradation process. This may partially account for their similar effectiveness in detecting degradation trends when sufficient labeled data is available.

However, when the training labeled dataset is reduced to only 50% of its original size, the Benchmark model shows a notable decline in performance, attributed to insufficient training data. This is apparent in the results displayed in Figure D.15, where the model struggles to discern sequential information, diminishing the distinction between early and late degradation stages. This performance drop suggests some degree of overfitting, likely because the halved dataset primarily includes data not close to the failure point, limiting the model’s ability to learn diverse degradation patterns.

### D.6.6 Generalization test of the proposed SSL strategy

Motivated by the need to address limitations in current research and expand the applicability of SSL methods in industrial settings, we propose to leverage a recent comprehensive milling process dataset [184] to conduct additional experiments. While previous studies have primarily focused on validating SSL methods using public datasets, there is a notable gap in research examining their effectiveness in noisy industrial environments with sparse labelling. This dataset, which provides multivariate time series data from milling processes across different machine tools and varying tool wear conditions, collected in the platform shown in [184], offers a unique opportunity to test the robustness and generalization capabilities of SSL methods in a more realistic and challenging context.

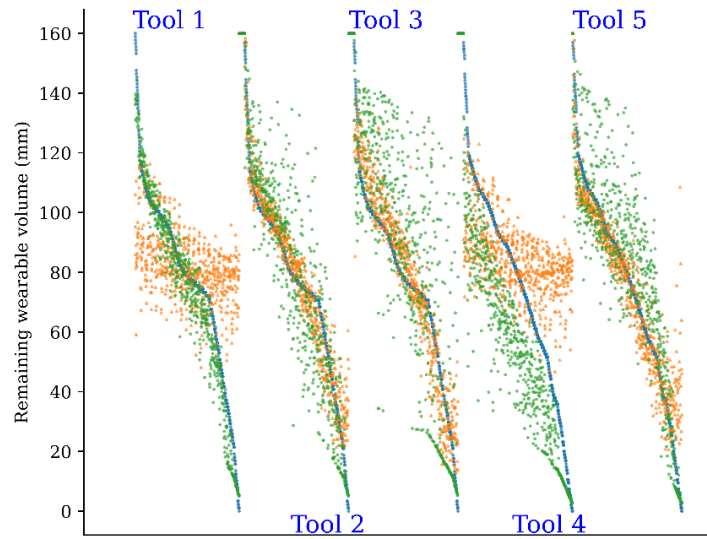
The dataset’s diverse signals, including process forces, drive forces, torques, and position data from different milling cutters and 5-axis milling centres, allow us to simulate the complexity and variability found in actual manufacturing environments. By utilizing this data, we aim to investigate how effectively our SSL method can model sparse labels under noise interference, a common challenge in industrial applications. Furthermore, to investigate the generalizability of our algorithm and the need for more extensive experimental verification, we will expand our testing to include multiple datasets and explore the performance of Contrastive Self-Supervised Learning applied to various models.

We still use the same structure as the validated SSL model on the bearing dataset as the baseline structure, the only changes are due to the different tool wear and bearing aging experiments:

- Tool wear has a very clear failure condition, so the failure threshold leading to the failure time no longer has to be estimated by the neural network. Here we use a wear amount of 160 mm as the end-of-life condition, i.e. the failure threshold used in the model.

## D.6. Supplementary results on the in-depth validation of the proposed enhanced SSL and various evaluation analyses

- The inputs to the bearing experiments are pre-processed frequency domain features, whereas for tool wear, the inputs are raw industrial monitoring data with noise in the shape of (4096, 4), and the four physical quantities monitored are Tool deflection position, Cutting force, Cutting torque, Motor output. The sampling frequency is 500Hz.
- In the bearing experiment, we predict RUL, and the predicted target in the tool wear measurement is the tool volume wear degree.



**Figure D.16:** Comparison of tool volume wear degree prediction results.

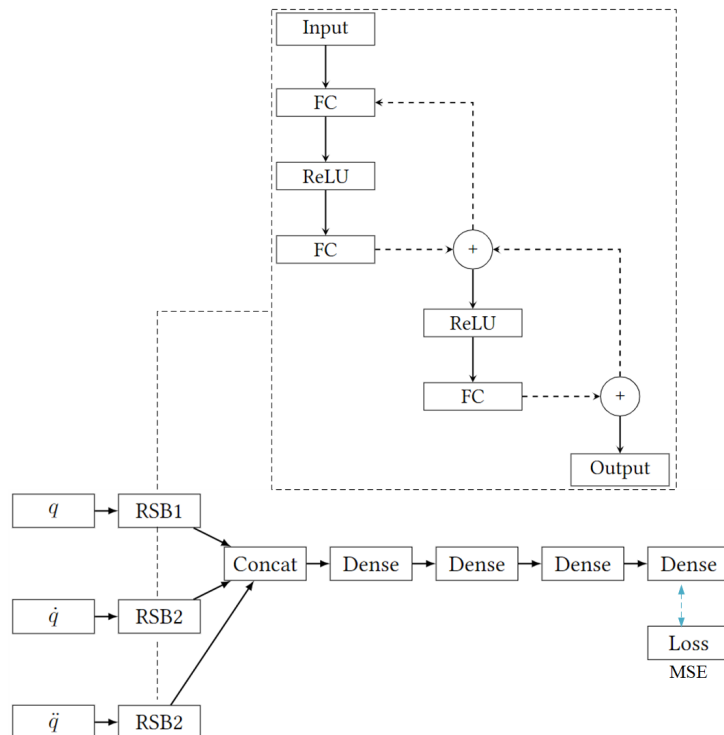
blue dots-ground truth, green “\*” symbols-proposed SSL model, yellow triangles-benchmark without SSL strategy.

As shown in Fig. D.16, the comparison of the tool’s tool volume wear degree prediction results highlights distinct differences between the proposed SSL model and the benchmark model without the SSL strategy. The blue dots represent the actual wear tracks, the green asterisks indicate the predictions of the proposed SSL model, and the yellow triangles denote the benchmark model’s predictions. Despite the overall MAE of 21.265 for the SSL model compared to 33.382 for the benchmark, the SSL model demonstrates superior performance across the tools, notably in cases such as Tool 2 and Tool 4. This suggests that the SSL model may be more robust in handling specific conditions or noise in industrial settings, providing a closer match to the actual wear tracks in these instances. This robustness underscores the potential of SSL methods in improving predictive accuracy and generalizability in noisy industrial environments with sparse labeling, highlighting areas for further optimization and broader experimental validation.



# Appendix: Breaking the old PIML and SSL paradigms to build a common PHM model across scenarios

## E.1 Detailed description of benchmarking models based on deep residual contraction networks



**Figure E.1:** Framework of the deep residual shrinkage network (DRSN).

## E.2. Proposed metric for evaluating inversion results of robotic arm dynamics

The proposed DRSN module has three inputs, all of which yield a (batch size, 1) input for the subsequent layers. It consists of 6 residual blocks with a fully connected layer. The output of the block is summed with a shortcut connection by adding a residual connection with the ReLU activation function. This model uses the mean square error of torque prediction as the loss function and is trained using the Adam optimizer.

## E.2 Proposed metric for evaluating inversion results of robotic arm dynamics

To evaluate the proposed method's performance in adapting the robot manipulator trajectory, we use mean square error (MSE), Fréchet distance [373], and Polygon Area difference [374]. Traditional metrics like MSE may not fully capture performance nuances. The Area Difference metric offers a holistic view by measuring cumulative deviation over the motion range, crucial for smooth operation. The Fréchet Distance provides a comprehensive measure of accuracy in following the desired trajectory, considering the sequence and timing of the arm's motion. These metrics are ideal for robotics, where precise movement and adherence to the intended path are critical.

**Polygon Area Difference** Polygon area is calculated by using Shoelace's formula Eq.(E.1):

$$\text{Area} = 0.5 \cdot \left| \sum_{i=0}^{n-2} (q_i \cdot y_{i+1} - q_{i+1} \cdot y_i) + (q_{n-1} \cdot y_0 - q_0 \cdot y_{n-1}) \right| \quad (\text{E.1})$$

where  $(q_i, y_i)$ ,  $i = 0, 1, \dots, n - 1$ , characterize the polygon vertices. Next, the absolute difference between the areas of the predicted and observed trajectories is shown in Eq.(E.2).

$$D = |\text{Area}_{\text{predicted}} - \text{Area}_{\text{actual}}| \quad (\text{E.2})$$

A smaller value of  $D$  indicates a better match between the predicted and the actual trajectories.

**Fréchet Distance** This metric measures the similarity between two curves, taking into account the location and arrangement of points. It is particularly useful for comparing robot manipulators' trajectories as it takes into account their spatial and temporal aspects. Given two curves  $P$  and  $Q$  represented by sequences of points, the Discrete Fréchet Distance (DFD) can be calculated by dynamic programming. The equation for the DFD is recursively defined as follows:

$$\text{DFD}(P, Q) = c(|P|, |Q|) \quad (\text{E.3})$$

### E.3. Supplementary validation results and discussion of the proposed E2NN performance

---

Where  $c(i, j)$  is a function defined as:

$$c(i, j) = \begin{cases} d(P_i, Q_j) & \text{if } i = 1 \text{ and } j = 1 \\ \max(d(P_i, Q_j), \min(c(i-1, j), c(i-1, j-1), c(i, j-1))) & \text{if } i > 1 \text{ and } j > 1 \\ \infty & \text{otherwise} \end{cases} \quad (\text{E.4})$$

Where  $d(P_i, Q_j)$  is the Euclidean distance between points  $P_i$  and  $Q_j$ , and  $|P|$  and  $|Q|$  are the lengths of the trajectories  $P$  and  $Q$  respectively. A smaller value means a better fit.

**Time cost** Apart from precision, computational efficiency is also a critical factor that determines the suitability of a dynamics algorithm for real-time applications. Therefore, the evaluation takes into account the expected computation time for each point, and the final assessment will include the average prediction time for a single point.

## E.3 Supplementary validation results and discussion of the proposed E2NN performance

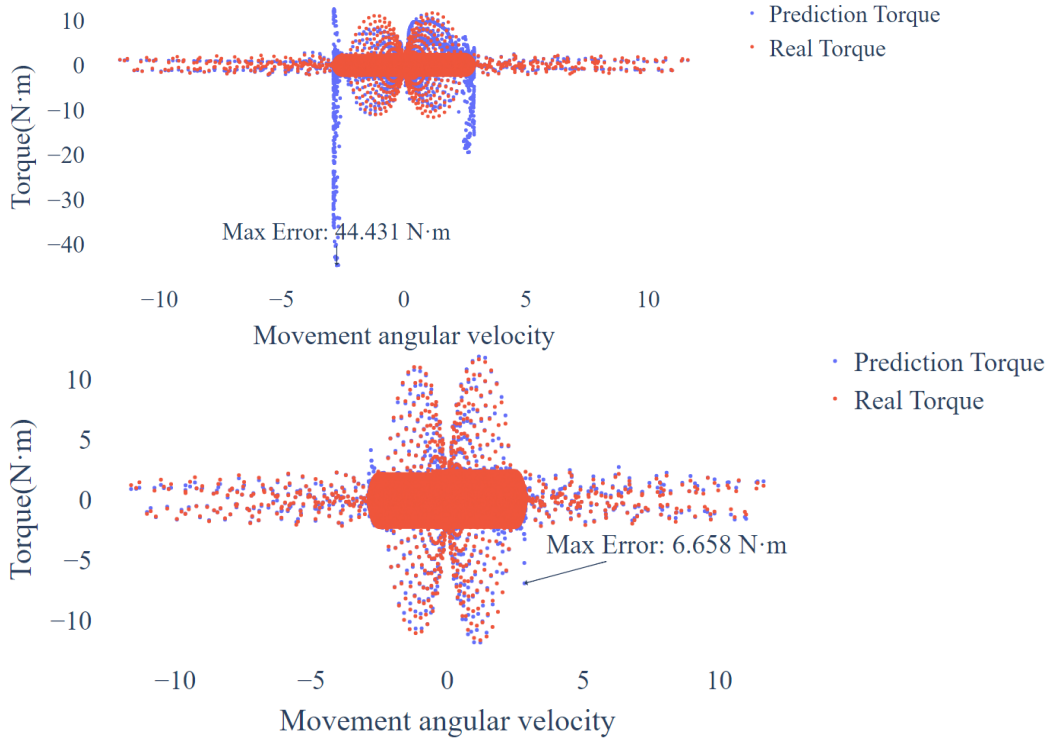
**Performance evaluation through simulated data.** The model is initially configured for 5000 training iterations. However, implementing an early stop mechanism enabled efficient convergence at 822 iteration epochs, with each step taking between 5-8 ms. This is achieved by training the model in TensorFlow, incorporating early stopping triggered after 60 iteration epochs without improvement, and using a checkpoint system to save the model parameters at the point of minimum training loss. This revised approach ensures a balance between sufficient training and computational efficiency. At training convergence, the torque prediction MAE of the Benchmark DRSN model is 0.10415 and the MAE of E2NN is 0.10716. Using the metrics proposed in Section. C.7, the test results on the 53976 movement recording points are shown in Table. E.1 and Fig. E.3. Moreover, the E2NN has a smaller model size and faster response speed than the benchmark DRSN, which makes it a more efficient and practical solution for robot trajectory fitting tasks. The results also show that while the benchmark model and the proposed method have similar error metrics values, the benchmark model's performance is achieved through over-fitting to local points, resulting in a group of large and discrete prediction points around the angular velocity scope in (-5,5). The errors' maximum value is equal to 44.431 N·m, which reflects ML's violation of physical facts. In contrast, the proposed E2NN architecture exhibited an overall trend that is closer to the ground truth.

The trajectory fitting results revealed that the E2NN is able to more accurately fit the robot trajectory to the ground truth compared to the benchmark DRSN. The difference in performance between conventional DRSN and E2NN when tested illustrates the existence of over-fitting in DRSN, especially as they perform similarly in training.

### E.3. Supplementary validation results and discussion of the proposed E2NN performance

**Table E.1:** Comparison of different methods on various metrics.

Method	Metrics			Response time	Parameters
	MSE	Area Difference	Fréchet Distance		
ANN	0.6	42.9	56.7	$3.3 \times 10^{-4}$	66753
E2NN	0.5	8.3	14.3	$1.1 \times 10^{-4}$	56223



**Figure E.2:** Comparison of the performance of the DRSN (above) and E2NN (below).

The difference between the training and final test results can be attributed to the fact that the training and test data are generated from different simulations. Specifically, the training data are generated from “direct-servo” and “inertia”, while the test data are generated from “friction”, which covers a wider range of working conditions and has different actual torque functions. However, E2NN is able to compensate for some of the missing information from “simulated training data” by embedding its corresponding equations.

**Discussion of measured data torque prediction results.** In the context of fitting angular displacement, angular velocity, and angular acceleration to robotic arm

### E.3. Supplementary validation results and discussion of the proposed E2NN performance

joint torque, this paper focuses on the performance of different models on such measured small sample data.

DRSN stands out among these models due to its balanced approach, offering a low maximum error of 0.374 and the lowest Fréchet Distance of 0.160 among the compared methods. Its architectural advantages, like residual concatenation and contraction operations, contribute significantly to this performance, enhancing feature selection and reducing noise, which is crucial for accurate modelling.

In scenarios with limited data, the XGBRegressor demonstrates its strength. With an MSE of 0.00247 and Area Difference of 3.703, it surpasses the Deep MLP model. This highlights the effectiveness of ensemble methods, particularly in small data contexts. The XGBRegressor, through its boosting process, incrementally corrects previous errors, thereby constructing a robust model that is less prone to overfitting – a common concern in small datasets. Its ability to adapt and improve gradually with each additional model in the ensemble is a key factor in its superior performance.

E2NN outperforms previous models like SVM, KNN, Deep MLP, and DRSN. The combination of the Physics-Informed module, trigonometric functions in the residual blocks, and the equation-based structure of the “Liquid” layer enables the E2NN to effectively mimic the inverse dynamics process of the robotic arm under physical constraints. It leads to notable improvements in predictive accuracy and generalization capabilities over models that do not incorporate such domain knowledge. One can see that E2NN can accurately fit the trajectory with high precision and computational efficiency under real friction conditions, while also exhibiting a good fit to the robot manipulator’s motion. The model’s performance is better than the simulation data, which suggests that the impact of friction on the robot manipulators during single-joint motion is less pronounced in the real-world scenario than in the simulation.

**Identification of inverse dynamics parameters.** In this paper, E2NN outputs joint torque, and its embedded physical operator can also serve to estimate the joint parameter  $\beta$ . To evaluate the E2NN’s learning mechanism and its grasp of robot motion dynamics, we extracted the weights corresponding to the embedded operator’s NN layer and compared them with the  $\beta$  values from the generated simulation data. This comparison allowed us to assess whether E2NN has effectively learned the robot motion dynamics.

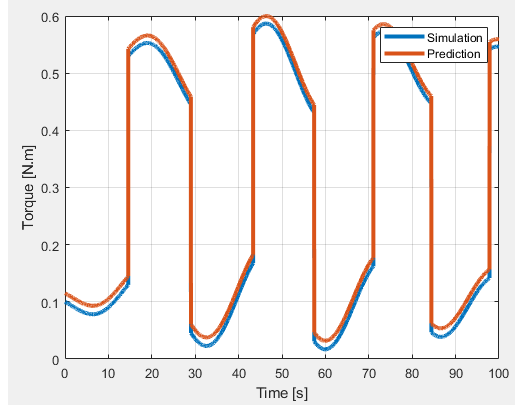
**Table E.2: Results of  $\beta$  estimation for the robotic manipulators torque model.**

$\beta_i$	1	2	3	4	5	6	7	8	9	10
Real	0	0.600	0	0	0	0.010	0.015	0.200	0.100	0.300
Proposed methods	-0.0132	0.617	0.146	0.023	0.0987	0.0134	0.126	0.303	0.116	0.312

The average weights of the E2NN dense layer are used as  $\beta$  values for the reconstructed robot manipulator’s torque trajectory. The fitting performance is evaluated by comparing the reconstructed trajectory with the ground truth trajectory. The obtained results are shown in Table. E.2 and Fig. E.3. The average accuracy of the proposed meth-

## E.4. Computational cost of the Transformer model

ods, measured as the MAE, for the parameter estimation in the robotic manipulators torque model is approximately 0.05433. This value represents the average deviation of the estimated values from the actual values across all parameters. The reconstructed trajectories using E2NN’s weights have a high torque fitting accuracy reaching 97.1% (as its trajectory is shown in Fig. E.3).



**Figure E.3:** Reconstructed torque by using Matlab.

These results indicate that the E2NN can approximately simulate the actual behaviour of the robot manipulator. This accuracy is critical in applications where fine motion control and subtle manipulation capabilities are required. The results successfully demonstrate the capability of the E2NN in predicting the joint torques and estimating the joint parameters  $\beta$  of a robotic manipulator. The key innovation of the E2NN lies in its embedded physical operator, which bridges the gap between theoretical modelling of robotics and practical applications.

## E.4 Computational cost of the Transformer model

In the self-attention mechanism, the computational complexity is primarily related to the sequence length  $N$  and the hidden dimension  $d$ . Each step’s computational complexity can be analyzed as follows: The generation of the Query ( $Q$ ), Key ( $K$ ), and Value ( $V$ ) matrices from the input matrix  $X$  requires a complexity of  $O(N \cdot d^2)$ , since each matrix is produced through matrix multiplication involving weight matrices of size  $d \times d$ . The computation of  $QK^T$  involves a complexity of  $O(N^2 \cdot d)$ , as  $Q$  has a size of  $N \times d$  and  $K^T$  has a size of  $d \times N$ . The softmax operation for normalising the attention scores has a complexity of  $O(N^2)$ ; however, this step is typically not the main computational bottleneck and can often be disregarded in the overall complexity analysis. Additionally, the multiplication of the attention scores matrix with the Value matrix  $V$  has a complexity of  $O(N^2 \cdot d)$ , given that the attention scores matrix is of size  $N \times N$  and  $V$  is of size  $N \times d$ .

## E.5. A detailed derivation of the gating selective state space mechanism

---

Combining the complexities of these steps, the total computational complexity of the self-attention mechanism is  $O(N \cdot d^2) + O(N^2 \cdot d) + O(N^2 \cdot d) = O(N \cdot d^2) + O(2 \cdot N^2 \cdot d)$ . Since  $N^2 \cdot d$  typically dominates over  $N \cdot d^2$ , the overall computational complexity can be simplified to  $O(N^2 \cdot d)$ . For a transformer model with  $l$  layers and an input batch size of  $b$ , the total computational complexity is  $O(l \cdot b \cdot N^2 \cdot d)$ . Specifically, the detailed complexity formula can be expressed as  $l \cdot (24 \cdot b \cdot N \cdot d^2 + 4 \cdot b \cdot N^2 \cdot d)$ , where the first term  $24 \cdot b \cdot N \cdot d^2$  represents the complexity of the feed-forward neural network part, and the second term  $4 \cdot b \cdot N^2 \cdot d$  represents the complexity of the self-attention mechanism.

## E.5 A detailed derivation of the gating selective state space mechanism

Generally, a time series is a sequence of successive observations of the same phenomenon at different times. Statistically, a time series is a realization of a stochastic process. It can be decomposed into either an additive or a multiplicative model, as shown in Eq. (E.5), directly or after a functional transformation.

$$\begin{aligned} \text{Additive Model: } s_t &= Per_t + Sea_t + Cyc_t + Irr_t, \\ \text{Multiplicative Model: } s_t &= Per_t \times Sea_t \times Cyc_t \times Irr_t, \end{aligned} \tag{E.5}$$

Here,  $Per_t$  represents the long-term trend,  $Sea_t$  is the seasonal component,  $Cyc_t$  is the cyclic component, and  $Irr_t$  is the irregular component. The multiplicative model is used when the trend is exponential and the magnitude of seasonal fluctuations increases over time.

Assuming an  $n$ -dimensional measurement vector  $s_t$  and an  $m$ -dimensional state vector  $\mathbf{h}_t$ , the measurement vectors are observable variables, while state vectors characterize the system's dynamics and are generally unobservable. The SSM described by Eq. (E.6), combines these vectors to provide a comprehensive description of the system's dynamics.

$$\begin{cases} \text{State equation:} & \mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{B}\mathbf{u}_t + w(t) \\ \text{Measurement equation:} & \mathbf{Y}_t = \mathbf{C}\mathbf{X}_t + \mathbf{D}\mathbf{u}_t + v(t) \end{cases} \tag{E.6}$$

In SMM, matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  have specific roles:

1.  $\mathbf{A}$  is the State Transition Matrix, describing how the current state influences the next state.
2.  $\mathbf{B}$  is the Input Matrix, describing how external inputs affect the system state.
3.  $\mathbf{C}$  is the Output Matrix, mapping the system state to observable outputs.
4.  $\mathbf{D}$  is the Feedforward Matrix, describing the direct effect of inputs on outputs.
5.  $w_t$  represents State Noise, and  $v_t$  represents Measurement Noise.

## E.5. A detailed derivation of the gating selective state space mechanism

---

6.  $\mathbf{u}_t$  is the control input vector at time  $t$ .

$$\mathbf{A} = \begin{pmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} \mathbf{C}_{1t} \\ \mathbf{C}_{2t} \\ \mathbf{C}_{3t} \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_{1t} \\ \mathbf{D}_{2t} \\ \mathbf{D}_{3t} \end{pmatrix} \quad (\text{E.7})$$

The subscripts  $j = 1, 2, 3$  correspond to the trend, cycle, and seasonal components, respectively. The SM model has the following characteristics:

1. It reflects the system's internal state and the relationship between internal states and external variables.
2. It processes multiple variable time series as vector time series, suitable for multiple input-output variable problems.
3. It requires minimal historical data for describing the system's state, making it time-efficient.

The traditional state-space approach transforms the time series into an SSM, estimates unknown parameters using maximum likelihood, and uses methods like Kalman filtering to compute state vectors. If all matrices vary with time  $t$ , the system is linear time-varying; if not, it is linear time-invariant.

However, integrating SM into modern deep-learning algorithms is challenging for the following reasons:

- **Computational efficiency:** Modern deep learning frameworks are optimized for discrete-time operations.
- **Training algorithms:** Most deep learning training algorithms are designed for discrete-time models.
- **Practical applications:** Many practical datasets are discrete.
- **Model complexity:** Controlling the complexity of the model by selecting an appropriate time step in discretization can be challenging.

Therefore, the discretization process for updating the continuous hidden state is as follows:

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t (\mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{u}(\tau)) d\tau + O \quad (\text{E.8})$$

where  $O$  represents the overall noise level as a constant of integration.

Since  $\mathbf{x}(\tau)$  cannot be obtained for all values in the continuous interval ( $0 \rightarrow t$ ), a discrete ML model cannot compute this integral. Therefore, it is logical to transform this integral into a recurrence relation:



## E.5. A detailed derivation of the gating selective state space mechanism

---

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \sum_{i=0}^k (\mathbf{A}\mathbf{x}(i) + \mathbf{B}\mathbf{u}(i))\Delta t + O \quad (\text{E.9})$$

This form requires modifying the original state equation to eliminate  $\dot{\mathbf{x}}(t)$  in Eq. (E.8). Thus a new function  $\mathbf{x}(t) = e^{-\mathbf{A}t}\mathbf{h}(t)$  is used and Eq. (E.8) is rewritten as Eq. (E.10):

$$\mathbf{h}(t) = e^{\mathbf{A}t}\mathbf{h}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau + Oe^{\mathbf{A}t} \quad (\text{E.10})$$

By slicing the continuous sequence epochs into a finite space with  $T$  as the state interval, Eq. (E.10) can be further transformed into a discrete form:

$$\mathbf{h}(k+1) = e^{\mathbf{A}T}\mathbf{h}(k) + \int_{t_k}^{t_{k+1}} e^{\mathbf{A}(t_{k+1}-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau \quad (\text{E.11})$$

Using zero-order hold for  $\mathbf{u}(t)$ :

$$\int_{t_k}^{t_{k+1}} e^{\mathbf{A}(t_{k+1}-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau = \int_{t_k}^{t_{k+1}} e^{\mathbf{A}(t_{k+1}-\tau)} d\tau \mathbf{B}\mathbf{u}(t_k) \quad (\text{E.12})$$

The result of the integral is:

$$\int_0^T e^{\mathbf{A}\tau} d\tau = \mathbf{A}^{-1}(e^{\mathbf{A}T} - \mathbf{I}) \quad (\text{E.13})$$

Finally, the discrete-time hidden states update equation between time step  $k$  and  $k+1$  for ML model is shown in Eq. E.14:

$$\mathbf{h}(k+1) = e^{\mathbf{A}T}\mathbf{h}(k) + (\mathbf{A}^{-1}(e^{\mathbf{A}T} - \mathbf{I}))\mathbf{B}\mathbf{u}(t_k) + Oe^{\mathbf{A}t} \quad (\text{E.14})$$

Substituting Eq. (E.14) into the update Eq. (E.6) for applying discrete SSM in ML, also replacing the  $T$  as the compressing value of the input  $x$  to incorporate the time-variant.

$$\begin{aligned} \mathbf{h}_k &= \bar{\mathbf{A}}\mathbf{h}_{k-1} + \bar{\mathbf{B}}\mathbf{x}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{h}_k + \mathbf{D}\mathbf{x}_k, \\ \bar{\mathbf{A}} &= e^{\mathbf{A}\Delta}, \\ \bar{\mathbf{B}} &= (e^{\mathbf{A}\Delta} - \mathbf{I})\mathbf{A}^{-1}\mathbf{B}, \\ \bar{\mathbf{O}} &= \mathbf{O}e^{\mathbf{A}\Delta} \end{aligned} \quad (\text{E.15})$$

The iterative expansion of Eq.(E.15), shown in Eq.(5.1), forms the basis for RNN models. Notably, since the  $A$  and  $B$  matrices are independent in RNNs, the hidden state can only be updated step-by-step through a recursive relationship, passing state information between time steps.

$$\begin{aligned}
y_n &= Ch_n \\
&= C \left( \bar{A}h_{n-1} + \bar{B}u_n \right) \\
&= C \left( \bar{A} \left( \bar{A}h_{n-2} + \bar{B}u_{n-1} \right) + \bar{B}u_n \right) \\
&= C \left( \bar{A} \left( \bar{A} \left( \bar{A}h_{n-3} + \bar{B}u_{n-2} \right) + \bar{B}u_{n-1} \right) + \bar{B}u_n \right) \\
&= C \left( \bar{A}^{n-1}h_1 + \bar{A}^{n-1}\bar{B}u_{n-1} + \bar{A}^{n-2}\bar{B}u_{n-2} + \dots + \bar{B}u_n \right) \quad (\text{E.16}) \\
&= C \left( \bar{A}^{n-1} \left( \bar{A}h_0 + \bar{B}u_1 \right) + \bar{A}^{n-2}\bar{B}u_2 + \dots + \bar{B}u_n \right) \\
&= C \left( \bar{A}^n h_0 + \sum_{k=1}^n \bar{A}^{n-k} \bar{B}u_k \right) \\
&= C \bar{A}^n h_0 + \sum_{k=1}^n C \bar{A}^{n-k} \bar{B}u_k
\end{aligned}$$

The hidden state is updated through a recursive relationship, passing state information between time steps. Therefore, during inference, only the current and previous states need to be stored, making the process fast and memory efficient. However, training must be performed sequentially, which limits parallel computing and increases computation time.

A closer look at Eq. (E.15) and Eq. (5.1), reveals that the main time consumption exists in the computation of the superposition of the ABC matrices. In contrast, in the discretized expression Eq. (5.1), the concatenated accumulation of  $A$ ,  $B$ ,  $C$  can be represented as a dot product of the convolution kernel and the inputs shown in Eq. (E.17).

$$\begin{aligned}
\text{kernel } \bar{K} &= (C\bar{B}, C\bar{A}\bar{B}, \dots, C\bar{A}^k\bar{B}, \dots) \\
y &= u * \bar{K} \quad (\text{E.17})
\end{aligned}$$

## E.6 Performance evaluation of End-to-End CNN-SSM model on different datasets

**Bearing prediction results.** Figure E.4 compares the predicted and actual Remaining Useful Life (RUL) values, marked by blue and orange stars, respectively. While the model captures the overall decline trend, the predicted values exhibit significant variation and a wider distribution compared to the smooth, consistent downward trajectory of the actual values. This discrepancy arises from the low-quality inputs the E2E model processes to ensure generality and ease of deployment. Without feature extraction, the model struggles to accurately predict RUL due to noise and low-density information in the raw data.

E.6. Performance evaluation of End-to-End CNN-SSM model on different datasets

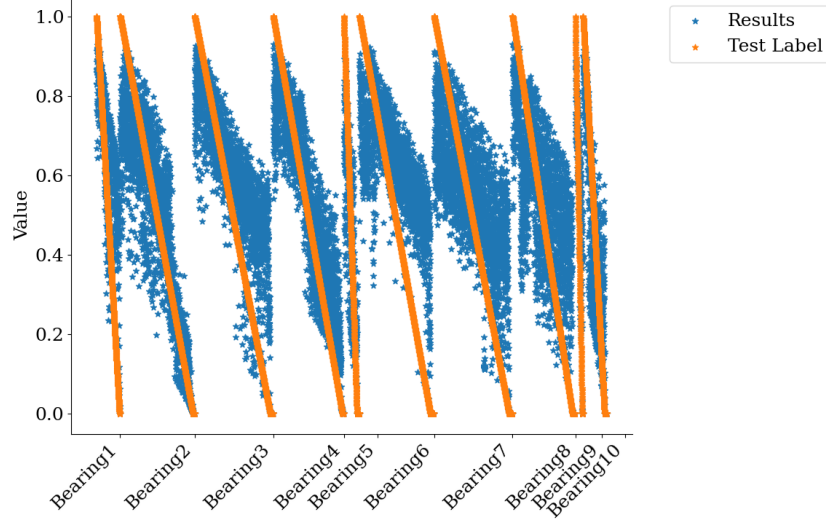


Figure E.4: Qualitative representation of bearing normalised RUL prediction results.

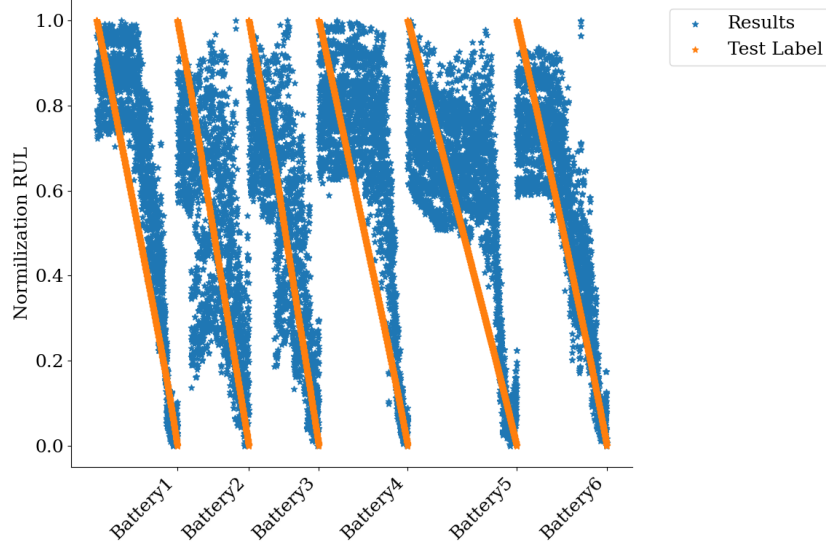


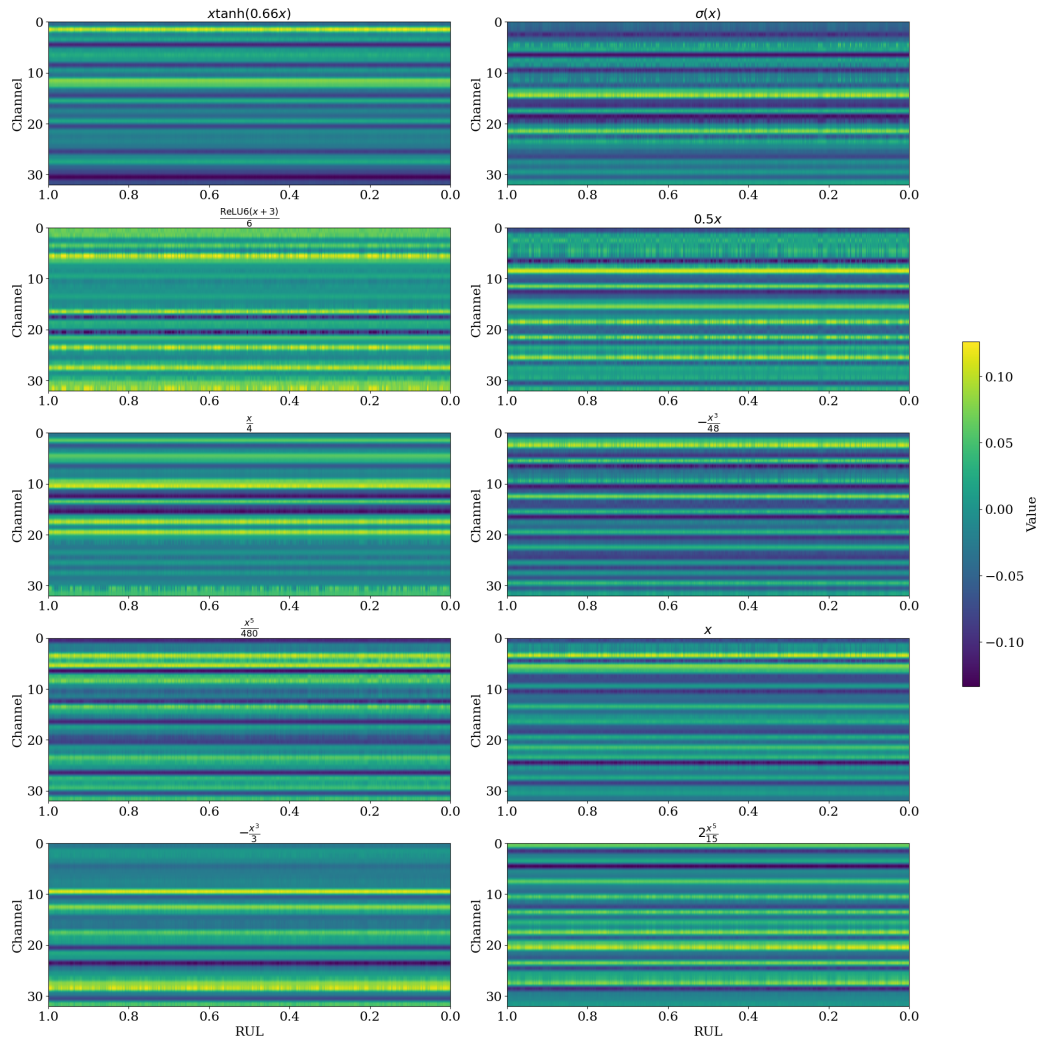
Figure E.5: Qualitative representation of battery normalized RUL prediction results.

## E.6. Performance evaluation of End-to-End CNN-SSM model on different datasets

---

**Battery prediction results.** From the scatter plot Fig. E.5, the trend lines of the predicted and actual values of battery RUL are very close to each other, showing that the model can capture the decaying trend of battery RUL very well. The predicted (blue star dots) and actual (orange star dots) values for each battery pack fall along similar trajectories, indicating that the model can accurately predict the RUL of the batteries at different stages of the life cycle. For example, the predicted and actual values for Battery 3 and Battery 6 fall along the same trend line, indicating that the model performs well for these battery packs.

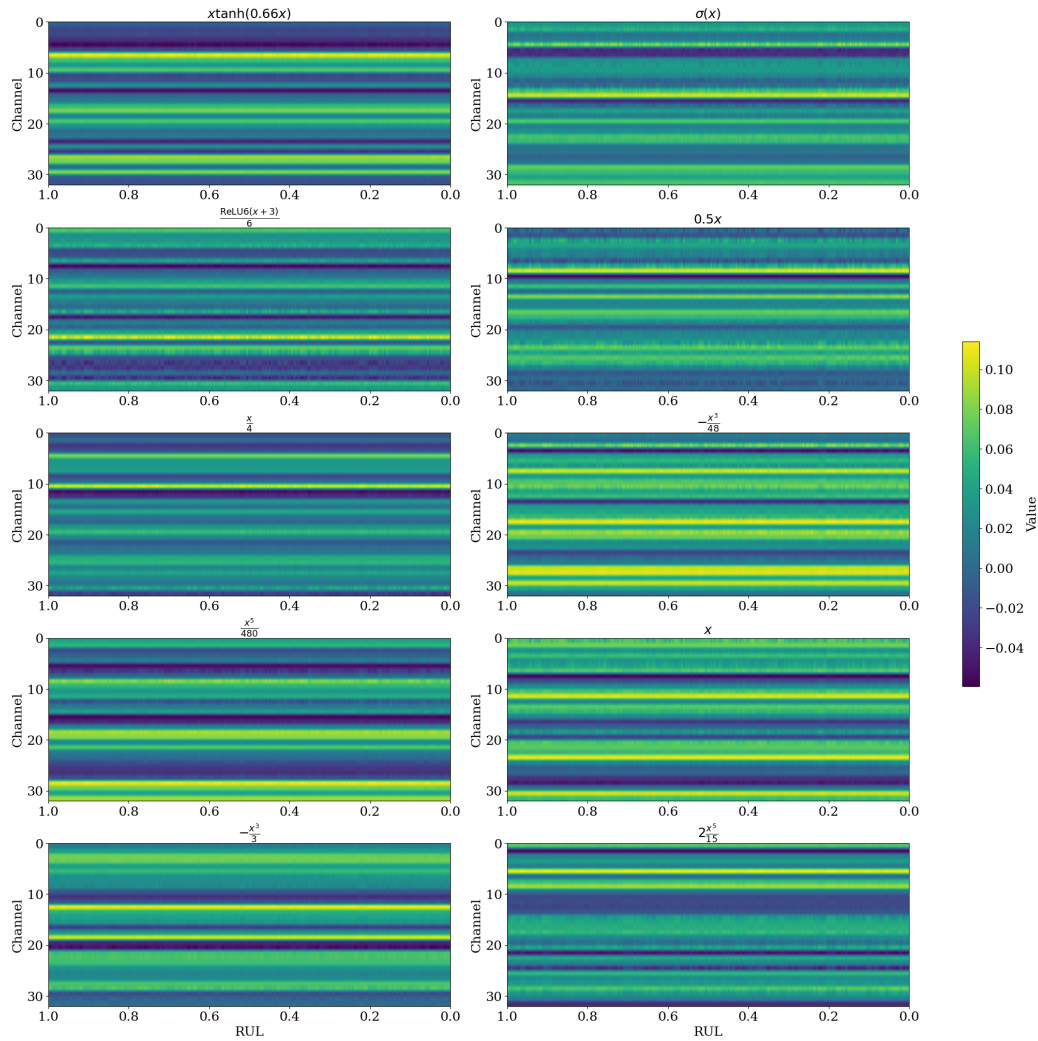
## E.6. Performance evaluation of End-to-End CNN-SSM model on different datasets



**Figure E.6:** Operator weights change when actively discovering knowledge from **current** signals during the 1st battery degradation.

The vertical axis represents the feature channel after initial data embedding and operator processing in the high-dimensional space. The horizontal axis represents the remaining lifetime decay process, with colors indicating the weights assigned to the operators.

## E.6. Performance evaluation of End-to-End CNN-SSM model on different datasets



**Figure E.7:** Operator weights change when actively discovering knowledge from **temperature** signal during the 1st battery degradation.

The vertical axis represents the feature channel after initial data embedding and operator processing in the high-dimensional space. The horizontal axis represents the remaining lifetime decay process, with colors indicating the weights assigned to the operators.

APPENDIX F  
**Abstract**

---

**Résumé** Cette thèse aborde le défi des « données éparses et des connaissances rares » dans le développement d'un modèle générique de pronostic et de gestion de la santé (PHM). Elle met en lumière l'efficacité des modèles hybrides combinant la modélisation basée sur la physique (PBM) et l'apprentissage automatique (ML), notamment l'apprentissage automatique informé par la physique (PIML) et l'apprentissage auto-supervisé (SSL) pour apprendre à partir de données non étiquetées. La thèse apporte ainsi des contributions significatives aux théories PIML et SSL et à leurs applications pratiques dans le PHM.

La première contribution est une solution générique d'architecture et stratégie d'apprentissage pour le PIML. Diverses approches sont analysées et la théorie mimétique est proposée pour concevoir des neurones et connexions flexibles et physiquement cohérents, aboutissant au Réseau Neuronal Mimétique des Éléments Finis du Rotor (RFEMNN). Le RFEMNN reconnaît efficacement les défauts à travers diverses structures de rotor. Pour améliorer la capacité de diagnostic du RFEMNN avec peu de données, une stratégie d'apprentissage par renforcement alignée avec la physique est proposée. Une architecture générique PIML avec des branches PI et basées sur les données est développée, impliquant un processus en trois étapes : pré-formation de la branche basée sur les données, formation de la branche PI, et formation conjointe. Cette méthode assure des performances supérieures aux modèles basés sur les données dans un contexte de données éparses. De plus, le modèle CNN dilaté utilisant cette approche prédit efficacement la RUL des batteries lithium-ion avec des données de petits cycles. La deuxième contribution est une stratégie SSL pour l'apprentissage à partir de données non étiquetées, introduisant un modèle Siamese CNN-LSTM avec une fonction de perte contrastive personnalisée. Ce modèle extrait des représentations robustes en maximisant les différences dans les mêmes échantillons présentés dans des ordres séquentiels variés. Des tâches en aval sont proposées comme objectifs intermédiaires pour aligner les représentations avec les exigences en aval. Le modèle Siamese CNN-LSTM excelle à prédire la RUL sur le dataset PRONOSTIA et reste stable même avec une augmentation de la rareté des données d'apprentissage.

La contribution finale étend les concepts de PIML pour la découverte active des connaissances sur des données non étiquetées et intègre le SSL dans la formation PIML en trois étapes. Une nouvelle structure PI liquide et un modèle PI-CNN-Selective state space model (CNN-SSM) sont développés. Liquid PI introduit des neurones à portes et des connexions liquides qui s'adaptent dynamiquement, acquérant des connaissances physiques grâce à une recherche optimisée. Appliquée dans le suivi du couple des manipulateurs robotisés, cette approche découvre des connaissances en utilisant des opérateurs physiques de base et des poids dynamiques. Le Liquid PI CNN-SSM traite des séquences d'entrée de longueur variable sans prétraitement du signal, optimisant les ressources en



nécessitant seulement 600 KB pour gérer 23,9 GB de données. Il atteint des performances de pointe dans des tâches de pronostic mixtes, y compris la dégradation des roulements, l'usure des outils de coupe, le vieillissement des batteries et la fatigue des tubes CFRP.

Les travaux futurs appliqueront des lois d'échelle spécifiques au PHM et utiliseront de vastes ensembles de données synthétiques et industrielles pour construire un macro-modèle. Ce modèle pourrait intégrer des capacités de diagnostic et de pronostic avec un traitement de séquence infinie, transformant les méthodologies et les solutions de PHM.

**Mots clés :** Gestion de la santé et des pronostics, Données éparses, Connaissances rares, PIML, SSL, Modèle générique.

**Abstract** This thesis addresses the critical challenge of “sparse data and scarce knowledge” in developing a generic Prognostics and Health Management (PHM) model. A comprehensive literature review highlights the efficacy of hybrid models combining physics-based modeling with machine learning, focusing on Physics-Informed Machine Learning (PIML) and Self-Supervised Learning (SSL) for enhanced learning from unlabeled data. Thereby, this thesis contributes to advancing both PIML and SSL theories and their practical applications in PHM.

The first contribution is developing a generic architectural and learning strategy solution for PIML. Various informed approaches are analyzed, and the mimetic theory is proposed to design flexible, physically consistent neurons and inter-layer connections. This novel approach leads to the development of the Rotor Finite Elements Mimetic Neural Network (RFEMNN), which mimics rotor finite element-based dynamics to adjust weight distribution and data flow within the neural network. RFEMNN effectively localizes and recognizes compound faults across multiple rotor structures and conditions. To enhance RFEMNN’s few-shot diagnostic capability, constraint projection theory and a reinforcement learning strategy are proposed, aligning the learning process with physics. A generic PIML architecture with parallel, independent PI and data-driven branches is proposed, involving a three-stage training process: pre-training the data-driven branch, freezing it to train the PI branch, and joint training of both branches. This method combines optimized local branches into a comprehensive global model, ensuring the PIML model’s performance exceeds original data-driven models under sparse data context. Moreover, the solid electrolyte interphase growth-informed Dilated CNN model using this approach showcases its superiority, surpassing leading models in predicting lithium-ion battery RUL with small-cycle data.

The second contribution is developing an innovative SSL strategy for unlabeled data learning, introducing a Siamese CNN-LSTM model with a custom contrastive loss function. This model extracts robust feature representations by maximizing differences in the same samples presented in varied sequential orders. Variants of downstream tasks are proposed as intermediate objectives in SSL pretext learning, integrating downstream structures into the pre-training model to align representations with downstream requirements. Under this strategy, the proposed Siamese CNN-LSTM excels at predicting RUL on the PRONOSTIA-bearing dataset and remains stable even as training data sparsity increases.

The final contribution extends PIML concepts for active knowledge discovery on unlabeled data and integrates SSL into the second phase of PIML’s three-step training, utilizing both labeled and unlabeled data. A novel Liquid PI structure and an end-to-end Liquid PI-CNN-Selective state space model (CNN-SSM) are developed. The Liquid PI design introduces gated neurons and liquid interlayer connections that adapt dynamically, acquiring physics knowledge through an

optimized search within a predefined operator pool. Demonstrated in torque monitoring of robot manipulators, this approach efficiently discovers knowledge using basic physical operators and dynamic weights from unlabeled data. The Liquid PI CNN-SSM processes variable-length input sequences without signal preprocessing, optimizing resources by requiring only 600 KB to handle 23.9 GB of data. It achieves state-of-the-art performance in mixed prognostic tasks, including bearing degradation, tool wear, battery aging, and CFRP tube fatigue, showcasing the originality and versatility of the proposed approach.

Future work will apply PHM-specific scaling laws and train on extensive synthetic and industry datasets to build a cross-modal macro-model. It could integrate diagnostic-prognostic capabilities with infinite sequence length processing, continuing to transform PHM methodologies and solutions.

**Keywords:** PHM, Sparse data, Scarce knowledge, PIML, SSL, Generic model.

Tarbes National Engineering School, Toulouse University, 47 Avenue d'Azereix,  
BP 1629 - 65016, Tarbes, France.

**Titre :** Amélioration du diagnostic et du pronostic dans des conditions de données rares et de connaissances limitées par l'apprentissage automatique informé par la physique et auto-supervisé

**Mots clés :** Pronostic et Gestion de la santé, Données éparées, Connaissances rares, PIML, SSL, Modèle générique

**Résumé :** Cette thèse aborde le défi des « données éparées et des connaissances rares » dans le développement d'un modèle générique de pronostic et de gestion de la santé (PHM). Elle met en lumière l'efficacité des modèles hybrides combinant la modélisation basée sur la physique (PBM) et l'apprentissage automatique (ML), notamment l'apprentissage automatique informé par la physique (PIML) et l'apprentissage auto-supervisé (SSL) pour apprendre à partir de données non étiquetées. La thèse apporte ainsi des contributions significatives aux théories PIML et SSL et à leurs applications pratiques dans le PHM.

La première contribution est une solution générique d'architecture et de stratégie d'apprentissage pour le PIML. Diverses approches sont analysées et la théorie mimétique est proposée pour concevoir des neurones et connexions flexibles et physiquement cohérents, aboutissant au Réseau Neuronal Mimétique des Éléments Finis du Rotor (RFEMNN). Le RFEMNN reconnaît efficacement les défauts à travers diverses structures de rotor. Pour améliorer la capacité de diagnostic du RFEMNN avec peu de données, une stratégie d'apprentissage par renforcement alignée avec la physique est proposée. Une architecture générique PIML avec des branches PI et basées sur les données est développée, impliquant un processus en trois étapes : pré-formation de la branche basée sur les données, formation de la branche PI, et formation conjointe. Cette méthode assure des performances supérieures aux modèles basés sur les données dans un contexte de données éparées. De plus, le modèle CNN dilaté utilisant cette approche prédit efficacement la RUL des batteries lithium-ion avec des données de petits cycles.

La deuxième contribution est une stratégie SSL pour l'apprentissage à partir de données non étiquetées, introduisant un modèle Siamese CNN-LSTM avec une fonction de perte contrastive personnalisée. Ce modèle extrait des représentations robustes en maximisant les différences dans les mêmes échantillons présentés dans des ordres séquentiels variés. Des tâches en aval sont proposées comme objectifs intermédiaires pour aligner les représentations avec les exigences en aval. Le modèle Siamese CNN-LSTM excelle à prédire la RUL sur le dataset PRONOSTIA et reste stable même avec une augmentation de la rareté des données d'apprentissage.

La contribution finale étend les concepts de PIML pour la découverte active des connaissances sur des données non étiquetées et intègre le SSL dans la formation PIML en trois étapes. Une nouvelle structure PI liquide et un modèle PI-CNN-Selective state space model (CNN-SSM) sont développés. Liquid PI introduit des neurones à portes et des connexions liquides qui s'adaptent dynamiquement, acquérant des connaissances physiques grâce à une recherche optimisée. Appliquée dans le suivi du couple des manipulateurs robotisés, cette approche découvre des connaissances en utilisant des opérateurs physiques de base et des poids dynamiques. Le Liquid PI CNN-SSM traite des séquences d'entrée de longueur variable sans prétraitement du signal, optimisant les ressources en nécessitant seulement 600 KB pour gérer 23,9 GB de données. Il atteint des performances de pointe dans des tâches de pronostic mixtes, y compris la dégradation des roulements, l'usure des outils de coupe, le vieillissement des batteries et la fatigue des tubes CFRP.

Les travaux futurs appliqueront des lois d'échelle spécifiques au PHM et utiliseront de vastes ensembles de données synthétiques et industrielles pour construire un macro-modèle. Ce modèle pourrait intégrer des capacités de diagnostic et de pronostic avec un traitement de séquence infinie, transformant les méthodologies et les solutions de PHM.

**Title:** Improving diagnostics and prognostics in sparse data and scarce knowledge conditions by physics-informed and self-supervised machine learning

**Key words:** Prognostics and health management, Sparse data, Scarce knowledge, PIML, SSL, Generic model

**Abstract:** This thesis addresses the critical challenge of "sparse data and scarce knowledge" in developing a generic Prognostics and Health Management (PHM) model. A comprehensive literature review highlights the efficacy of hybrid models combining physics-based modeling with machine learning, focusing on Physics-Informed Machine Learning (PIML) and Self-Supervised Learning (SSL) for enhanced learning from unlabeled data. Thereby, this thesis contributes to advancing both PIML and SSL theories and their practical applications in PHM.

The first contribution is developing a generic architectural and learning strategy solution for PIML. Various informed approaches are analyzed, and the mimetic theory is proposed to design flexible, physically consistent neurons and interlayer connections. This novel approach leads to the development of the Rotor Finite Elements Mimetic Neural Network (RFEMNN), which mimics rotor finite element-based dynamics to adjust weight distribution and data flow within the neural network. RFEMNN effectively localizes and recognizes compound faults across multiple rotor structures and conditions. To enhance RFEMNN's few-shot diagnostic capability, constraint projection theory and a reinforcement learning strategy are proposed, aligning the learning process with physics. A generic PIML architecture with parallel, independent PI and data-driven branches is proposed, involving a three-stage training process: pre-training the data-driven branch, freezing it to train the PI branch, and joint training of both branches. This method combines optimized local branches into a comprehensive global model, ensuring the PIML model's performance exceeds original data-driven models under sparse data context. Moreover, the solid electrolyte interphase growth-informed Dilated CNN model using this approach showcases its superiority, surpassing leading models in predicting lithium-ion battery RUL with small-cycle data. The second contribution is developing an innovative SSL strategy for unlabeled data learning, introducing a Siamese CNN-LSTM model with a custom contrastive loss function. This model extracts robust feature representations by maximizing differences in the same samples presented in varied sequential orders. Variants of downstream tasks are proposed as intermediate objectives in SSL pretext learning, integrating downstream structures into the pre-training model to align representations with downstream requirements. Under this strategy, the proposed Siamese CNN-LSTM excels at predicting RUL on PRONOSTIA-bearing dataset and remains stable even as training data sparsity increases.

The final contribution extends PIML concepts for active knowledge discovery on unlabeled data and integrates SSL into the second phase of PIML's three-step training, utilizing both labeled and unlabeled data. A novel Liquid PI structure and an end-to-end Liquid PI-CNN-Selective state space model (CNN-SSM) are developed. The Liquid PI design introduces gated neurons and liquid interlayer connections that adapt dynamically, acquiring physics knowledge through an optimized search within a predefined operator pool. Demonstrated in torque monitoring of robot manipulators, this approach efficiently discovers knowledge using basic physical operators and dynamic weights from unlabeled data. The Liquid PI CNN-SSM processes variable-length input sequences without signal preprocessing, optimizing resources by requiring only 600 KB to handle 23.9 GB of data. It achieves state-of-the-art performance in mixed prognostic tasks, including bearing degradation, tool wear, battery aging, and CFRP tube fatigue, showcasing the originality and versatility of the proposed approach.

Future work will apply PHM-specific scaling laws and train on extensive synthetic and industry datasets to build a cross-modal macro-model. It could integrate diagnostic-prognostic capabilities with infinite sequence length processing, continuing to transform PHM methodologies and solutions.