UNIVERSITY OF CALIFORNIA,
IRVINE

Interactive Causality Enabled Adaptive Machine Learning

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Yutian Ren

Dissertation Committee:
Professor G. P. Li, Chair
Professor Pramod P. Khargonekar
Professor Charless C. Fowlkes

2023

DEDICATION

To my father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my PhD advisor, Prof. G. P. Li, for his unwavering guidance, support, and mentorship throughout my doctoral journey. His expertise, patience, and encouragement have been invaluable in shaping my research and personal growth. Prof. Li's profound insights, constructive feedback, and dedication to academic excellence have constantly inspired and motivated me to push the boundaries of knowledge. His mentorship has been instrumental in refining my research skills and fostering a deeper understanding of the subject matter.

Beyond their academic brilliance, Prof. Li's kindness, approachability, and genuine concern for my well-being have made the academic journey an enriching and enjoyable experience. His unwavering belief in my abilities has given me the confidence to overcome challenges and strive for excellence. I owe a significant portion of my academic achievements to Prof. G. P. Li, and I will forever cherish the knowledge and wisdom imparted during our interactions.

I am also deeply grateful to Prof. Pramod Khargonekar and Prof. Charless Fowlkes, who served as members of my PhD Committee, and Prof. Mohammad Al Faruque and Prof. Yasser Shoukry, who served as members of my PhD Qualifying Exam Committee. Their valuable insights, constructive feedback, and scholarly expertise have played a pivotal role in enriching the quality of my research work. Their thoughtful guidance and thoughtful critiques have consistently pushed me to explore new perspectives and refine my ideas.

I would like to take this opportunity to express my heartfelt gratitude to the exceptional individuals who played pivotal roles in the successful accomplishment of the Smart Connected Worker project. First and foremost, I am deeply thankful to my esteemed lab mentors, Dr. Richard Donovan, and Dr. Mike Klopfer. Their unwavering guidance, expertise, and encouragement have been instrumental in shaping the direction of my research and providing valuable insights throughout the project. Their mentorship has not only honed my technical skills but has also fostered a deep appreciation for innovation and critical thinking in approaching complex problems. I am also immensely grateful to all my labmates who contributed to the project with their collaboration, support, and camaraderie. The collaborative environment of the lab has been a constant source of motivation and inspiration, propelling the project to new heights.

In addition to the esteemed individuals mentioned earlier, I would like to extend my heartfelt gratitude to my beloved family and friends. Their unwavering support, encouragement, and selfless love have been the pillars of strength throughout my academic journey.

To my family, I am deeply grateful for their constant belief in my abilities and for always standing by me, cheering me on, and offering words of encouragement during the most challenging times. Their unwavering support has been a source of motivation, reminding me of the importance of pursuing my passion and academic goals.

To my friends, I am thankful for their camaraderie, understanding, and the many moments of

joy and laughter we shared. Their presence in my life has provided a much-needed balance amidst the academic rigor, reminding me of the importance of cherishing friendships and maintaining a positive outlook.

I would like to express my sincere gratitude to Henry Samueli Endowed Fellowship, DOE CESMII (Grant No. DE-EE0007613), and the Broadcom Foundation for their invaluable funding support throughout my research journey. Chapter 4 of this dissertation is a reprint of the material as it appears in [116], used with permission from Elsevier. The co-author listed in this publication is G. P. Li. Chapter 5 of this dissertation is a reprint of the material as it appears in [117], used with permission from IEEE. The co-author listed in this publication is G. P. Li.

In conclusion, I want to express my deepest appreciation to everyone for being an integral part of this transformative journey. Your belief in my potential, unwavering support, and boundless love have made all the difference. This dissertation is a testament to the collective efforts and encouragement of all the remarkable individuals in my life.

VITA

Yutian Ren

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2023**
University of California, Irvine *Irvine, CA*

Bachelor of Engineering in Electronic Science and Technology **2018**
Southeast University *Nanjing, China*

RESEARCH EXPERIENCE

Graduate Student Researcher **2018–2023**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant for EECS 70A-LA **Winter 2023**
Teaching Assistant for EECS 70B-LB **Spring 2023**
University of California, Irvine *Irvine, California*

REFEREED JOURNAL PUBLICATIONS

An Interactive and Adaptive Learning Cyber Physical Human System for Manufacturing With a Case Study in Worker Machine Interactions **2022**
IEEE Transactions on Industrial Informatics

A Contextual Sensor System for Non-Intrusive machine status and energy monitoring **2022**
Journal of Manufacturing Systems

Smart Connected Worker Edge Platform for Smart Manufacturing: Part 2—Implementation and On-site Deployment Case Study **2022**
Journal of Advanced Manufacturing and Processing

Smart Connected Worker Edge Platform for Smart Manufacturing: Part 1: Architecture and Platform Design **2022**
Journal of Advanced Manufacturing and Processing

Machine Learning-based Real-time Monitoring System of Manufacturing Workflow for Smart Connected Worker to Improve Energy Efficiency **2021**
Journal of Manufacturing Systems

Applications of Information Channels to Physics-Informed Neural Networks for WiFi Signal Propagation Simulation at the Edge of the Industrial Internet of Things **2021**
Neurocomputing

REFEREED CONFERENCE PUBLICATIONS

Intelligent Sleep Control at Standby State for Energy Saving in Semiconductor Manufacturing Processes Involving Mechanical Vacuum Pumps **Jun 2023**
Industrial Energy Technology Conference

ABSTRACT OF THE DISSERTATION

Interactive Causality Enabled Adaptive Machine Learning

By

Yutian Ren

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2023

Professor G. P. Li, Chair

The capability to adapt to dynamic environments and changes in data distribution is essential for the development of adaptive artificial intelligence (AI) systems that can operate effectively in the real world. Typically, adaptive learning requires an AI agent to autonomously gather the required information (*i.e.*, training data) from the deployed environment to adapt to the local dynamic changes. This process falls within the broad area of semi-supervised learning, where partial training data is not manually labeled. This is especially critical for applications at the edge of cyber-physical systems, where available datasets are rather limited and resources for deep learning from big data are not available. While there exist effective semi-supervised learning methods based on feature smoothness assumptions, they become less robust in dynamic environments where there are shifts (*e.g.*, concept drift) in data distribution. Recently, causality has been explored to address deployment domain shifts. This is based on the argument that causality, especially the causal direction, is consistent across different domains, making invariant features highly significant in adaptive learning.

This dissertation presents an interactive causality (IC) methodology, which utilizes directional and temporal causal events to facilitate the automated self-labeling of data. Interactive causality refers to the temporal state transitions of causes and effects during interactive activities, which are captured through additional observing channels. The methodology cap-

italizes on the domain-invariant causal relationships to capture information leakage during interactions from an additional observing channel, and is fortified by the rich domain knowledge in cyber-physical system contexts. Differing from big data ML approaches, the IC leverages the existing knowledge and experiences to create a knowledge graph with embedded temporal causality among interactive nodes and directly look at the interactivity among these nodes for adaptive learning. A theoretical foundation of the interactive causality driven self-labeling method is discussed and compared with other traditional semi-supervised learning algorithms. The proof is derived from the theory of dynamical system which represents the time-varying environments and the time-series data. A simulation using physics engine and a real-world example in semiconductor manufacturing are provided to demonstrate the effectiveness of the proposed method for adaptive learning and knowledge expansion. Overall, the experimental results successfully demonstrated the efficacy of our proposed adaptive learning in reducing manual data labeling efforts and robust domain adaptation.

Chapter 1

Introduction

1.1 Background

This generation's Artificial Intelligence (AI) empowered by data-driven modeling and learning has reformed diverse fields with the advanced and autonomous intelligence and incubated many promising visions for the future. Novel and powerful deep learning (DL) algorithms nowadays, compared with the ones ten years ago, have significantly advanced in many of the commonly seen tasks on various data types. The emergence of ChatGPT and other generative models has improved people's in-office productivity by distilling vast amount of information in natural languages for users. Other than Information Technology (IT) industry, AI-empowered intelligence has begun its democratization in traditional sectors to improve productivity and reduce cost such as in agriculture and manufacturing.

The promotion of AI in traditional fields is not as fast as in IT industry. There are many reasons. The uncompleted transformation of digitization and informatics in these fields restricts the acquisition and archive of task-oriented data used for DL training. Some novel AI applications in these fields need additional dimensions of data beyond what has been

established and collected in the era of Industry 3.0. The specialized domain knowledge in the traditional fields requires AI engineers to detail the understanding of problems and application scenarios. However, the lack of IT expertise raises the cost of problem formulation, data collection and annotation, model development, and model and system maintenance. Over the years before the age of digitization, traditional industries have developed their ways to accumulate and share domain knowledge, which can be summarized as the field of industrial engineering and operation research, but lack a standard way to share knowledge to AI experts for AI problem formulation. These main reasons, lack of data and specialized domain knowledge, need solutions to pave the road for pervasive AI adoption in traditional fields.

Recently, the exploration and development of few-shot learning, transfer learning, unsupervised and semi-supervised learning, and self-supervised learning aim to lower the barrier of AI adoption by reducing the needs of manually annotated training data or enhancing the domain adaptation and robustness of AI models in different application domains. These methods step on algorithmic development and have shown promising results in research fields. Some of these methods have also demonstrated effectiveness in traditional industries, *e.g.*, manufacturing. For example, few-shot learning is suitable for vision-based defect detection in manufacturing where only limited samples are accessible. Transfer learning based on pretrained feature extraction models can be used to adjust and adapt large models to specific domains and applications. Most of the current methods highly depend on rich and invariant data features that are common across domains. Drastic cross-domain variations, such as concept drift, can make these methods less effective. In addition, while some of the current solutions can reduce the needs of training on large amount of labeled samples, manually labeled data are still required and the models lack a way of automatic evolution and adaptation to local environmental changes.

Cyber-Physical Systems (CPS) represent a class of systems where the physical world and

computational elements are deeply intertwined and interact seamlessly to achieve specific objectives. A CPS is a complex integration of physical processes, sensors, actuators, and computational algorithms that work together to control and monitor physical entities and processes. These systems have the ability to sense, process, and actuate the physical world in real-time. CPS can be found in various domains, including manufacturing, transportation, healthcare, energy, agriculture, smart cities, robotics, and more. CPS sets a spacious stage for AI applications in traditional industries, especially using small-scale ML models rather than large models due to requirement of rapid response, energy efficiency, and also the intrinsic properties for ML training. In CPS applications, large amount of datasets is usually unavailable and ML tasks are usually less complicated than human-like generic vision or language models. Hence, large models can be a overkill and perform even worse on relatively simple ML tasks with limited datasets compared to traditional data processing methods. In addition, the ever-changing scenarios in CPS require the deployed ML models be able to autonomously adapt to the changes.

On the other hand, the needs for AI in traditional industries are different from consumer electronics. For example, while manufacturing industry has been investing major efforts for transformation to IT-enhanced production, the needed infrastructure and data for AI application development do not align with the current IT infrastructure in manufacturing. The maturity of AI is also a problem. Ad-hoc data processing and understanding of specific domain knowledge are intensively required. This high customization of AI models for different specific problems significantly complicates AI development in traditional industries with less generalizability. Overall, these problems motivate the author to think about a different paradigm of AI development targeting at the pain points.

The problem that this dissertation focuses on is that most of the current ML strategies require a manually labeled dataset in pre- and post-deployment stages for initial training and post-deployment drift adaptation, which significant hinders the usage of data-driven

AI in traditional fields with less data science expertise. This dissertation approaches the problem from a system perspective and designs an adaptive machine learning system such that ML models can achieve autonomous learning without the needs of labor-intense manual data collection and annotation. In the next sections, we will review the basic concepts of related topics including data distribution shift, causality, ontology and knowledge graph, and adaptive machine learning. These reviewed topics are close to this dissertation's concentration and can provide readers with certain contexts to help on the understanding of this dissertation. In Section 1.6, the contribution of this dissertation is outlined.

1.2 Data Distribution Shift

In ML production environments, data distribution shift is one of the major reasons that causes deployed ML system failures in the form of less accurate prediction. Typically, a ML model can learn a underlying data distribution from training dataset and conduct prediction based on the learned distribution. However, in real-world deployment scenarios, the underlying distribution in real world data can be different from that of the training dataset, and the real world data distribution can be non-stationary due to unknown factors. These problems will degrade the performance of deployed ML models and thus there is a need to collect and label new datasets for retraining models to counter the data shifts. In general, there are three distinct types of data distribution shifts: covariate shift, label shift, and concept drift. Covariate shift occurs when the distribution of input data changes, but the conditional probability of a label given an input remains constant. Label shift occurs when the output distribution changes, but the input distribution remains the same for a given output. Concept drift means when the input distribution remains unchanged, but the conditional distribution of the output given an input shifts. In other words, with concept drift there will be a different output for the same input.

Data distribution shift has been a long-standing problem and many researchers have proposed and demonstrated solutions. There are two aspects of this problem: drift detection and drift adaptation. The objective of drift detection is to effectively monitor the data distribution and identify harmful shifts to reduce the needs of frequent model retraining. After drift identification, the adaptive learning will play a role to adapt existing ML models to the drift.

In general, drift detection is based on statistical methods including three categories as summarized by [91]: error rate based methods, data distribution based methods, and mulitple hypothesis test based methods. These methods help on understanding the drift: when it happens and the severity of drift. For instance, a classic error-rate based method, called Drift Detection Method (DDM), applies a trained classifier to predict labels and compares the predicted labels with available true labels within a window which generates an error rate used with thresholding for drift alarms [40]. The data distribution based methods focus on data itself by setting up a distance metric to quantify the similarity between the historical and new data distributions. Multiple hypothesis test based methods differ from previous methods in that multiple hypothesis test configured in parallel or hierarchical ways. In terms of drift adaptation, retraining models is essential to adapt models to the changed data and thus a new dataset is needed. Currently, the drift adaptation methods focus on different retraining ways, especially for the problem of catastrophic forgetting that will cause a machine learning model to forget previously learned information after retraining. Therefore, careful tuning of models and specialized retraining techniques are necessary to avoid this problem.

1.3 Causality

Causality is a fundamental concept in various fields of research, including philosophy, economy, science, and statistics. Over the years, extensive research has been conducted to under-

stand the nature of causality, its implications, and methods for causal inference. Causality refers to the relationship between cause and effect, where a cause produces an effect. Establishing causal relationships is essential for understanding how different factors or variables influence each other and for making accurate predictions or interventions. Many research topics has been highlighted in the field of causality, such as causal discovery, causal inference, counterfactual analysis, and causal machine learning. From statistical perspective, there are two widely used causality modeling techniques, Granger Causality and structural causal models.

Granger causality (GC) is a statistical concept and methodology used to assess the causal relationship between two time series variables. It was developed by Nobel laureate Clive Granger in 1969 and has become widely used in econometrics and other fields for studying causal relationships in time-dependent data [52]. Granger causality is based on the principle that if a time series variable X "Granger-causes" another variable Y, then the past values of X contain information that helps predict the future values of Y, beyond what can be predicted using only the past values of Y itself.

Structural causal model (SCM), also known as a structural equation model (SEM) or a causal graphical model, was first proposed by Judea Pearl in late 1980s [107]. It is a mathematical framework used to represent and study causal relationships among variables and provides a formalized approach to understanding how variables interact with each other and the underlying mechanisms that generate the observed data by using Bayesian network theory. SCM often utilizes causal graphical models, such as directed acyclic graphs (DAGs), to visually depict the causal relationships between variables. SCM defines a do-calculus which provides a set of formal rules for manipulating causal expressions involving interventions, counterfactuals, and observational data. It allows people to make causal inferences and estimate causal effects from observed data and knowledge about the underlying causal structure. Compared with GC, SCM lacks the view of temporal information but utilizes probabilistic graphical

models to process many causal variables.

Causality differs from correlation from several aspects. Causality involves a temporal order, cause-and-effect mechanisms, interventions, counterfactual reasoning, and considerations of confounding factors. Correlation, on the other hand, describes statistical associations between variables without specifying the direction, causality, or temporal precedence. Establishing causality requires more rigorous analysis and evidence than establishing correlation. In order to determine causality, researchers often conduct experiments or use methods like randomized controlled trials, where they manipulate one variable while keeping others constant, to establish a cause-and-effect relationship. Such randomized controlled trials have been widely used in medicine fields to examine the causal effects of novel treatments. There are several data-driven causal discovery algorithms introduced to efficiently discover causal relations and build causal graphs among many variables based on Bayesian network and various conditional independence tests. However, such statistical causality may no faithfully represent the physical cause and effect mechanism due to the existence of unknown confounders. As such, solid identification of causation is always a topic pursued by researchers.

1.4 Ontology and Knowledge Graph

An ontology model is a conceptual representation of knowledge in a specific domain. It defines the concepts, entities, relationships, and properties within a domain and organizes them in a structured and hierarchical manner. Ontology is widely used in various fields, including artificial intelligence, knowledge representation, semantic web, and information systems. In simpler terms, an ontology serves as a framework for presenting the characteristics of a particular subject area and illustrating their connections. It accomplishes this by establishing a collection of concepts and categories that embody and depict the subject. In every academic discipline or field, ontologies can be developed to manage complexity and structure data into

meaningful information and knowledge. Ontology has been applied in many fields, such as semantic web [60], various machine learning applications [9, 58, 112], Industry 4.0 [150, 133], IoT [37], and cybersecurity [134]. These applications have demonstrated the effectiveness and potential of ontologies to effectively capture the domain knowledge, enable efficient knowledge sharing and integration, and facilitate reasoning and inference capabilities in a domain.

A knowledge graph (KG) is a structured representation of knowledge that captures relationships between entities, concepts, and facts in a domain. It is a graph-based model that organizes information in the form of interconnected nodes and edges, where nodes represent entities or concepts, and edges represent relationships or associations between them. Knowledge graphs are designed to capture and encode rich semantic information, allowing for efficient storage, retrieval, and analysis of data. They enable advanced knowledge discovery, reasoning, and inference capabilities, making them valuable tools for various applications such as search engines, recommendation systems, and question-answering systems. Conceptually different from ontology models, a knowledge graph is a specific implementation or instantiation of a knowledge representation system that utilizes a graph structure to capture and represent information. While an ontology model provides a high-level conceptual framework, a knowledge graph is a concrete instantiation of that model, representing the actual data and relationships within a domain. Similar to ontology, KG has been widely used in many applications. Especially, dynamic knowledge graph recently becomes popular as it is designed to evolve and adapt to changes in the underlying data and knowledge it represents. Dynamic knowledge graphs have been researched in many fields, such as digital twin [2], event forecasting [29], and large language models [7]. KG is basically a way to represent knowledge and can be utilized in various manners. It can be solely used as a knowledge representation for computers to understand the logic among nodes. Alternatively, KG can be used as embedding in algorithm learning as a way to propagate information over graphical connections. The dynamicity of KG provides an additional temporal dimension to represent

time-varying relations such as interactions, which is useful in the scope of this dissertation.

1.5 Adaptive Machine Learning

Adaptive machine learning refers to the ability of machine learning models and algorithms to adapt and learn from new data or changing environments. It allows the models to continuously update their knowledge and improve their performance over time. Due to its necessity and practicability, many research efforts have been made in this direction. Adaptive learning is usually discussed with other similar AI concepts including continual learning, lifelong learning, transfer learning, and online learning.

Continual learning and lifelong learning are basically the same concept referring to the ability of a learning system or model to continuously acquire and integrate new knowledge and skills over time while retaining previously learned information. It involves adapting to changing environments, handling evolving data distributions, and accommodating new tasks or concepts without significantly forgetting or overwriting previously learned ones. In continual learning, the learning system is exposed to a stream of data or a sequence of tasks, and it must update its knowledge incrementally to incorporate new information. The goal is to achieve a cumulative learning process, where the system's performance improves or remains stable over time as it encounters new experiences.

Transfer learning refers to a machine learning technique where knowledge gained from training on one task is leveraged to improve learning and performance on a different but related task. It involves transferring learned representations or knowledge from a source task to a target task. In transfer learning, a pretrained model, often trained on a large and diverse dataset, serves as a starting point. This model has already learned general patterns, features, or representations that are useful for various tasks. Instead of training a model from scratch

on the target task or domain, transfer learning allows reusing parts of the pretrained model or its learned representations to expedite and enhance the learning process on the target task.

Online learning refers to the process of training machine learning models on data that arrives in a sequential manner, often in real-time, and updating the model's parameters as new data becomes available. Unlike batch learning, where models are trained on a fixed dataset, online machine learning adapts and learns from streaming data continuously. In online machine learning, the model is trained incrementally as new data samples arrive, and predictions are made in real-time or near real-time. The model's parameters are updated dynamically, reflecting the changing patterns and characteristics of the incoming data. This iterative learning process allows the model to adapt to evolving data distributions, handle concept drift (changes in the target variable or underlying relationships), and capture temporal dependencies.

In the context of this dissertation, adaptive learning puts a focus on allowing models to automatically modify their behavior, structure, or parameters in response to new data or data distribution shifts, without requiring manual intervention. At the same time, it has the potential to continuously evolve to reach higher detection accuracy in non-shifted domains and class incremental learning. Adaptive learning can be generally categorized into semi-supervised approach. Besides semi-supervised approaches, unsupervised domain adaptation is also widely used. Unsupervised Domain Adaptation (UDA) [43] proposes a model architecture to jointly optimize a feature extractor and two discriminative classifiers to address the problem of domain shift or distribution mismatch between the training data and the target data. Since then, many novel UDA architectures are developed [89, 119, 72].

1.6 Contributions and Dissertation Outline

This dissertation proposes a novel adaptive machine learning system based on the contextual causal relationships embedded in interactive activities. It differs from traditional algorithmic development to achieve autonomous domain adaptation. We focus on system-level design incorporating the causal contexts that a machine learning task is involved in to provide natural labels and associate data streams via learnable causal time lags. In addition, a holistic interactive causality methodology is proposed to model and extract contextual causation from domain knowledge, conduct adaptive learning, and expand knowledge graphs. It highlights this idea different from traditional machine learning methods where machine learning is utilized to extract the correlation among many data sources. We propose to use the existing domain knowledge to create a knowledge graph to represent the causality underneath interactivity among nodes. These interactive nodes can be directly utilized for adaptive machine learning.

In Chapter 2, the core idea of self-labeling for adaptive machine learning is introduced and illustrated along with a theoretical proof using dynamical systems theory and a simulated experiment using physics engine. The proof and experiment utilize simple causal relations and demonstrate that the causality and interaction time based self-labeling is more robust than traditional semi-supervised learning for countering data distribution shifts. This chapter lays the theoretical foundation of the feasibility and superiority of self-labeling. In Chapter 3, a holistic methodology including knowledge modeling for a domain is described. It provides an answer to where to find the existing causality, how to model it, and how to gain new knowledge given a knowledge graph. The knowledge modeling utilizes the concept of ontology and knowledge graph to mimic how humans build up perception for a new domain. Moreover, a more comprehensive self-labeling scheme is provided for four basic causal structures in statistical causal graphs which can be used to analyze more complex causal graphs. A simulation with a complex causal graph is provided and the results further demonstrate

the effectiveness of self-labeling.

Chapter 4 and Chapter 5 are about a real-world case study in semiconductor manufacturing to demonstrate the capability of the proposed interactive causality methodology. In this study, we aim to develop an adaptive learning system that can recognize worker-machine interactions by using videos as inputs. Two types of machines, a manually operated one (PlasmaTherm) and a PLC controlled one (E-beam), are selected as the testbed. Based on the existing knowledge in machines' standard operating procedures (SOP), the causality between workers' actions and machine's responses are extracted and represented as a Finite State Machine model. An additional observing channel, which is machines' responses in the form of energy consumption, is utilized to observe the causal effect of worker-machine interactions. In Chapter 4, a contextual sensor system is developed on these two machines to detect the state transition of each machine component and capture corresponding contexts. In Chapter 5, an Interactive Cyber Physical Human System (ICPHS) is proposed and describes a three-stage methodology incorporating different human roles to develop an adaptive model for worker-machine interactions. The experiments are conducted in a clean-room facility when users operate machines to autonomously adapt an skeleton-based action recognition model (graph convolutional networks) and demonstrate in a real world environment the effectiveness of interactive causality based self-labeling. In Chapter 6, several future directions are discussed from three aspects: theory, methodology, and applications. It is envisioned that broad CPS applications in various fields can be developed based on this dissertation's work.

Chapter 2

A Self-Labeling Method for Adaptive Machine Learning by Interactive Causality

2.1 Introduction

Machine learning (ML) equips substantial applications with predictive intelligence. However, for most of the ML using supervised models for reliable performance, the training set collection and annotation consumes considerable time and labor efforts. While it is generally agreed that, similar to infant learning, certain level of supervision is beneficial, a large dataset with complete annotation for training is not practical nor anticipated in future AI applications.

Recent progress in semi-supervised learning and self-supervised learning has reduced annotation needs while preserving the superior performance in many classic ML tasks [154, 31, 32, 10]. Current semi-supervised methods assume feature similarity or distribution smooth-

ness across labeled and unlabeled data and rely on feature-rich data [109], which constrains their usage with non-ideal datasets. Data distribution shifts such as concept drift and covariate shift can worsen semi-supervised methods' performance after deployment in dynamic environments [41]. Despite self-supervised learning's advantages in learning data representation at the pretext stage, manual generation of data labels for downstream tasks is still required[101]. These challenges motivate researching alternative methods to address them.

Recent studies have explored using causality to aid ML with domain adaptation to distribution shifts [157, 117], inspired by converse cases where ML is used for causality identification. A plausible and compelling argument is that while data distributions can differ in different domains due to unknown environmental dynamics, causal relationships and the causal direction in particular, are invariant across domains [120]. Motivated by the intuitive significance of invariant features, we propose a novel and effective way of leveraging invariant causality to address post-deployment domain shifts via automatic label generation. This is remarkably useful in interaction scenarios associated with causal mechanisms. In static ML tasks such as offline object recognition, interactions do not play an explicit role. However, when ML models are deployed for real-time decision making tasks such as autonomous driving, interactions between cars and pedestrians when yielding or changing lanes generate rich data with underlying causality that can be processed for intention recognition. Causal effects, where a change in one object's properties can elicit a response in another, can be mutual or unilateral, transmitting force, energy, or information to trigger observable effects. Meanwhile, the time interval between cause and effect contributes informatively to understanding causal relationships [48, 18, 34], but receives little attention in current causality-aided ML research.

In this study, we exploit interaction scenarios where multiple objects, domains, or humans interact, maintaining unambiguous causal relationships, which we refer to as interactive causality. We propose a self-labeling method to automate post-deployment data annotation

based on temporal relationships of causal events. Our method focuses on post-deployment data shifts where multimodality facilitates the utilization of additional effect-side observation channels for label inference. The self-labeling method learns the time interval between asynchronous causal and effect events, and uses observable effect data to self-label cause data. This self-labeled data is used to adaptively retrain an ML model of predicting effects from causes under unknown data shifts. The self-labeling method rests on the assumption that the temporal relationship between causes and effects is more consistent and less subject to domain shifts relative to input feature similarity as hypothesized in semi-supervised learning. We utilize 1-d dynamical systems to prove that the proposed method consistently outperforms traditional semi-supervised learning methods under specified conditions. A computer simulation models physical interactions upon which the self-labeling method is evaluated and exceeds several benchmarks. Finally, we discuss the connection between the proposed method and relevant fields. The abbreviation SSL below will refer to semi-supervised learning. Related code is available at `https://github.com/yutianRen/slb_interactive_causality`.

2.2 Related Work

Our work relates to two broad areas of research: (a) automatic data labeling methods and (b) causality-inspired machine learning techniques for domain adaption. We review recent works and backgrounds in each topic. A comparison is provided in Section 2.2

Automatic Data Labeling Methods. There are generally four widely-used measures for automatic labeling, namely self-supervised learning, pseudo labels, delayed labels, and domain knowledge. Pseudo labels and its variants use trained ML models or clustering methods to generate labels for unlabeled data that is then used to retrain models, or more recently, to jointly optimize target learning and label generation [79, 22, 11, 151, 161]. As with other semi-supervised methods, pseudo labels rely heavily on feature similarity between

Paper	Method	Advantages	Limitations
[101, 32, 10]	Self-supervised	Good for representation learning at pretext stages.	Need manual labels for downstream tasks.
[79, 22, 11, 151, 161, 17]	Pseudo labels	Strong theoretical assumption in quasi static environments.	Tend to fail with dynamic distribution shifts.
[49, 54, 108, 99, 127]	Delayed labels	Focus on real data streaming scenarios with delays.	Ignore the physical meaning of delays.
[56, 28, 130, 24, 8]	Domain knowledge	Robust and suitable for individual cases.	Lack a systematic method to generalize.
[51, 50, 155, 156, 95]	Causality inspired	Build on strong invariant causality for knowledge transfer.	Overlook temporal relationships in causality.

Table 2.1: Comparison of related methods.

labeled and unlabeled data and the distinctiveness of features across different classes. For example, Asano *et al.*, [11] added an equipartition constraint to maximize mutual information between data indices and labels for pseudo labels generation. Yan *et al.*, [151] improved pseudo-labels by ensembling predicted probabilities of multiple randomly augmented versions of the same sample for source-free unsupervised domain adaptation. Zhou *et al.*, [161] enhanced reliability of self labeling for contrastive learning by weighing on estimated feature similarity from query and regular one-hot labels. Delayed labels refers to cases where label feedback comes after the input data in data streams [49, 54, 108, 99]. The label latency here coincides with the causal time interval we defined in our self-labeling method to be discussed in later sections. Most existing works only acknowledge the delays and attempt to reduce the delay's impact on model learning or evaluation [53], but ignore the physical meaning of the delay itself, which our study aims to address. Domain knowledge includes logical relations of data, ontology, and knowledge bases, etc. [113, 100], and is typically converted to constraints applied during model training. For example, Gupta *et al.*, [56] automatically labeled drivers' yield intentions by using the resultant car positions from changing lane behaviors to infer preceding driving actions. Causality also falls within the scope of domain knowledge. We envision that the causal relation of interactive objects, particularly causal directions, can be extracted from existing knowledge.

Causality Inspired ML. Progress in statistical causality, such as Granger Causality (GC) and Structural Causal Models (SCM), have formalized causality testing, representation, and analysis with mathematical tools. Recently, ML algorithms have been used in conjunction with statistical causal representations for the causal analysis, such as in multi-domain causal structural learning [44], causal imitation learning [118], causal discovery [64], and causal in-

ference by graphical models [122, 76, 85]. Schölkopf [120] looked conversely into how causality can be used in ML, especially semi-supervised learning, to enhance robustness by leveraging cross-domain invariant causal mechanisms. Since then, Gong *et al.*, [51, 50] modeled the data generation process as a causal mechanism to learn conditional transferable components for domain adaptation. Zhang *et al.*, [155] incorporated causal relationship into DNN design by manipulating data features from causal variables to enhance DNN robustness. Stewart in [130] presented an example of using logical constraints from domain knowledge to model outputs and loss function. Most research in this field has been built on the assumption that the generation of cause data and the causal mechanism ($P(effect|cause)$) are independent, overlooking the temporality of causal relationships [83]. We approach causality from a unique perspective where time is equally critical and informative.

2.3 Methodology and Theory

The data annotation consists of two major steps: 1) selection of data samples to be labeled; 2) generation of labels for the selected data samples. When labeling image datasets, most time researchers will not require step 1 as images are pre-selected. However, for sensors (*e.g.*, cameras) capturing streaming data in dynamic environments, both steps are required for annotation. In the proposed self-labeling paradigm, we automate both steps without human intervention.

The self-labeling method aims at assisting ML tasks. ML tasks refer to the regular pattern recognition tasks accomplished by supervised machine learning models. The proposed self-labeling method involves an interaction scenario where participating objects interact and induce effects. A 2-object interaction scenario is used as an example to illustrate the idea as shown in Fig. 2.1. The causal relation between object o_1 and o_2 is unidirectional and known *a priori*. The ML task is to train a model that digests the cause data of o_1 to infer the effect

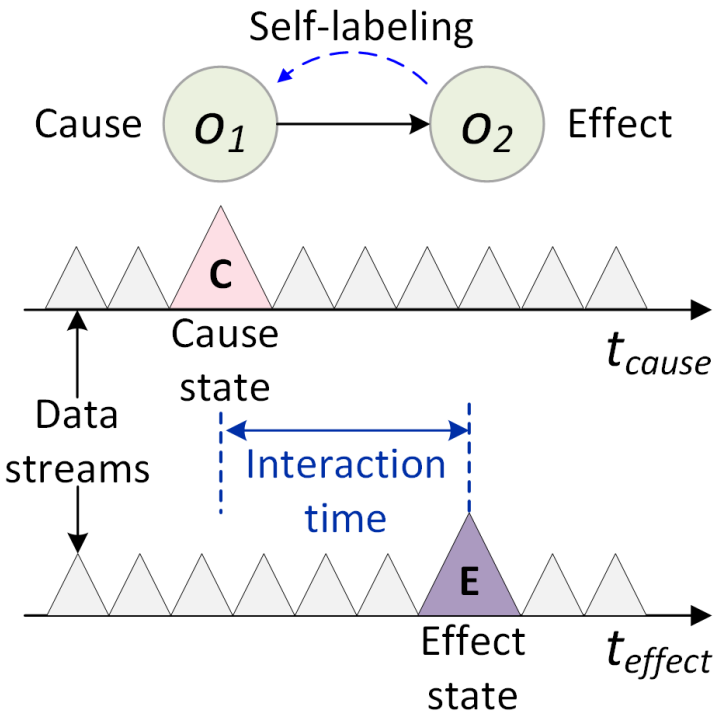

Figure 2.1: An illustration of causality-based self-labeling and the definition of interaction time.

of o_2. Two streams of sensor data from the cause and effect sides are available during model deployment. If we view the causality from an ML perspective, the causal data are the input features, and the effect data are the output labels. The interaction time is defined as the time lag between corresponding cause and effect states.

The self-labeling procedure is illustrated in Fig. 2.2. Before model deployment, following the procedure in Fig. 2.2(a) the interaction time between the data streams of two objects can be identified. Causal relationships can come from existing knowledge or causal modeling. An auxiliary interaction time model (ITM) infers the interaction time using the effect data. The ITM can be trained using supervised or unsupervised methods and can be an ML, statistical, mathematical, or physical model. As there are two data models, we designate the primary functional ML model as the task model and the other as the ITM for distinguishing purposes. Optionally, the task model can be pretrained during the derivation of supervision

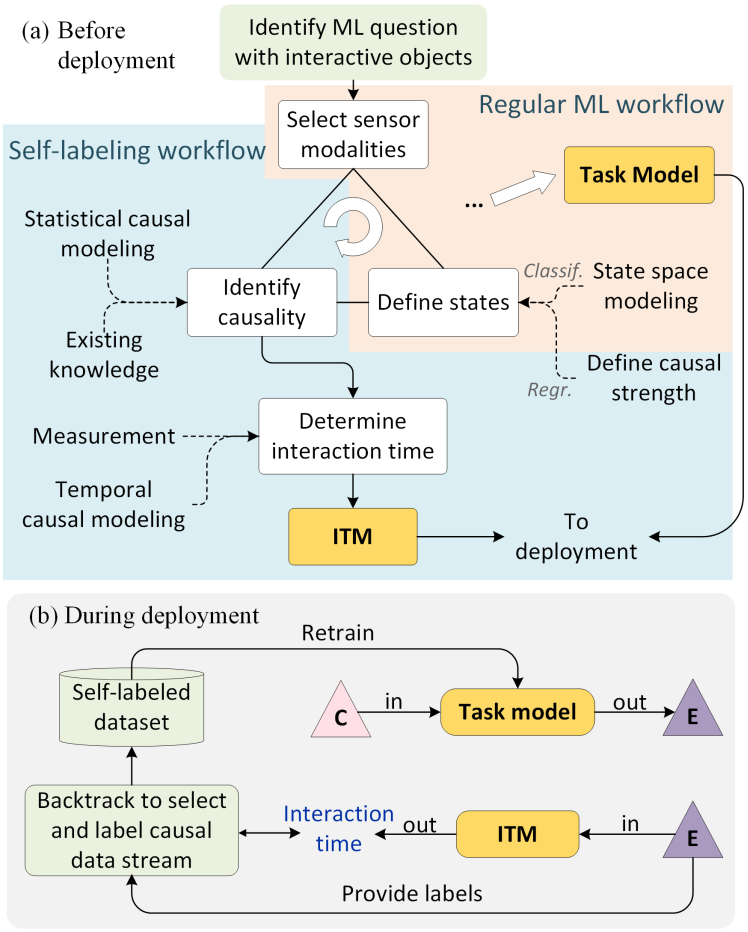

Figure 2.2: Proposed self-labeling workflow before and during deployment.

for the ITM with concurrent labeling for task model. During deployment, the self-labeling is accomplished with three sub-steps: 1) when effect data is received, the ITM infers the interaction time; 2) the effect data serves as the label or can be simply processed to generate the label; 3) starting from the timestamp of the effect data, the system will backtrack the period of inferred interaction time to select the corresponding segment of cause data to be labeled. Then, we can derive multiple self-labeled data-label pairs and retrain the task model to improve its performance.

There are several advantages to the self-labeling method. The self-labeling method belongs to the semi-supervised category, and can considerably reduce manual annotation efforts. After the deployment of the task model, it can achieve continual learning to correct for data

distribution drift. In this sense, the task model is able to evolve to automatically capture and label data from deployment environments, leading to increased real-world performance. Moreover, using causal data to predict effect data is meaningful because of their temporal precedence. Once the causal data is received, the expected effect can be predicted for prompt decision making.

In addition, we propose the following conditions for the self-labeling method: 1) a situation where object interactions happen; 2) a known or derived causal relationship during interactions; 3) the interaction time needs to be dependent on the effect data for the ITM; 4) the effect data is easier to process than the cause data. The last is not a necessary condition but can be viewed the criterion for when applying self-labeling is beneficial and selecting the effect observer as an alternative for observing the cause. Condition 3 critically determines the feasibility of self-labeling and also guides the effect observer selection.

2.3.1 Proof by Dynamical Systems

As self-labeling aids in modeling time-evolving systems, we use dynamical systems (DS) to demonstrate that the self-labeling method consistently outperforms traditional SSL that relies on the distribution smoothness to infer labels in resolving concept drift problems. DS use differential or difference equations to describe system states evolving with time. Many real-world systems can be modeled as DS if the system state changes with time. In the literature [77], the interaction of two DS is modeled as coupled differential equations. We consider a simplified case of two interacted 1-d DS in the proof. They are represented as

$$\dot{x} = f(x) \tag{2.1}$$

$$\dot{y} = y + h(x) \tag{2.2}$$

where x and y are scalar system states and serve as cause and effect respectively. $f(\cdot)$ defines a vector field, and $h(\cdot)$ is the coupling function. When there is an unknown perturbation in system x, the cause side will show a corresponding disturbance, changing the cause-effect relationship. In this example, the perturbation simulates concept drift in ML and is not considered as noise as in control theory. Instead, we regard it as the unknown factor changing the learned relationship between inputs and outputs to validate the learning performance. With system perturbation, the two systems become

$$\dot{x} = f(x) + d(x) \tag{2.3}$$

$$\dot{y} = y + h(x) \tag{2.4}$$

where $d(x)$ represents the perturbation. We define the perturbation as being related to system state rather than being an independent value (as a function of t). This is reasonable as we only consider it to be a factor to change the input-output relation of an ML model after training rather than real noise. We will discuss the $d(t)$ case in Section 2.4.

System x and y have initial and final values represented by x_1, x_2, y_1, y_2, respectively, where subscript 1 is initial value and 2 is final value. The systems propagate from initial to final values during a time period that is defined as the interaction time. We define x_1 as the cause state and y_2 as the effect state in x-y interaction. The ML task is to learn a mapping between cause x_1 and effect y_2. We will then follow the proposed self-labeling method to derive the self-labeled x_1-y_2 relation under perturbation and compare with conventional SSL and fully supervised ways.

The derivation of self-labeled (SLB) x_1-y_2 relation will follow the steps: 1) without perturbation, derive the relation between interaction time t_{if} and effect state y_2 used for inferring interaction time from effect state; 2) under perturbation, use the true effect y_2 to infer the interaction time, and select the corresponding state x_1 as the self-labeled x_{slb}; 3) under per-

turbation, derive the relation between x_{slb} and y_2. The above steps generally follow Fig. 2.2 but can be simplified. Since we need to derive the relation of $y2$ and x_{slb}, we can combine the t_{if} and y_2 relation derived from unperturbed case and the t_{if} and x_{slb} relation derived from perturbed case and cancel out t_{if}.

In the unperturbed case, we need to use the effect y_2 to infer the interaction time t_{if}. Eqs. (2.1) and (2.2) can be solved with initial values to derive

$$x(t) = A^{-1}(t + A_{x_1}) \tag{2.5}$$

$$y(t) = e^t \int_0^t e^{-\tau} \cdot h(A^{-1}(\tau + A_{x_1}))\, d\tau + e^t y_1 \tag{2.6}$$

where $A(x) = \int^x \frac{1}{f(\xi)}\, d\xi$ (constant is not needed in the integral) and is locally invertible on $[x_1, x_2]$. Subscript A_{x_1} represents $A(x_1)$. x_1 in Eq. (2.6) needs to be substituted with x_2 since x_1 is unknown during inference, but x_2 can be regarded as a parameter. Then, letting $y(t) = y_2$, we can find the relation between t_{if} and y_2 to be used to infer the interaction time as

$$y_2 = e^{t_{if}} \int_0^{t_{if}} e^{-\tau} \cdot h(A^{-1}(\tau + A_{x_2} - t_{if}))\, d\tau + e^{t_{if}} y_1. \tag{2.7}$$

With perturbation, Eq. (2.7) is used to infer t_{if} given y_2. Next, we can solve Eqs. (2.3) and (2.4) to derive the evolution function

$$x(t) = B^{-1}(t + B_{x_1}) \tag{2.8}$$

$$y(t) = e^t \int_0^t e^{-\tau} \cdot h(B^{-1}(\tau + B_{x_1}))\, d\tau + e^t y_1 \tag{2.9}$$

where $B(x) = \int^x \frac{1}{f(\xi)+d(\xi)}\, d\xi$ and is locally invertible on $[x_1, x_2]$.

From Eq. (2.8), we can derive the true interaction time needed for the evolution from x_1

to x_2 under perturbation, which is $t_{true} = B_{x_2} - B_{x_1}$. Given t_{if} and t_{true}, the self-labeled $x_{slb} = B^{-1}(t_{true} - t_{if} + B_{x_1})$, and then we can derive the relation between t_{if} and x_{slb} as

$$t_{if} = B_{x_2} - B_{x_{slb}}. \tag{2.10}$$

Now we can use Eq. (2.10) to substitute t_{if} in Eq. (2.7) and derive the y_2 and x_{slb} relation under perturbation as

$$y_{2slb} = e^{B_{x_2} - B_{x_{slb}}} \cdot \left(\int_0^{B_{x_2} - B_{x_{slb}}} e^{-\tau} h(A^{-1}(\tau + A_{x_2} - B_{x_2} + B_{x_{slb}}))\, d\tau + y_1 \right) \tag{2.11}$$

which is the input-output relation learned by the ML task model using our self-labeling method under perturbation.

Eq. (2.11) needs to compare with traditional SSL and fully supervised (FS) method. Most traditional SSL relies on the feature similarity of input data to assign labels to unlabeled data. In this example of interacting dynamical systems, traditional SSL methods learn a x_1-y_2 relation in an unperturbed environment during the supervised stage. In a perturbed environment, the learned x_1-y_2 relation is leveraged to infer pseudo labels given unlabeled perturbed x_1. Therefore, traditional SSL methods can theoretically only learn the unperturbed x_1-y_2 relation. The FS method referred to here is the ground truth relation between perturbed x_1 and y_2 by training on data-labels pairs, and can be derived from Eqs. (2.3) and (2.4). By directly solving the original and perturbed DS, we can derive

$$y_{2trad} = e^{A_{x_2} - A_{x_1}} \cdot \left(\int_0^{A_{x_2} - A_{x_1}} e^{-\tau} h(A^{-1}(\tau + A_{x_1}))\, d\tau + y_1 \right) \tag{2.12}$$

$$y_{2fs} = e^{B_{x_2} - B_{x_1}} \cdot \left(\int_0^{B_{x_2} - B_{x_1}} e^{-\tau} h(B^{-1}(\tau + B_{x_1}))\, d\tau + y_1 \right) \tag{2.13}$$

ID	$f(x)$	$d(x)$	$f(x)+d(x)$	Relation
1	+	+	+	$fwd > trad > slb > fs$
2	+	-	+	$fs > slb > trad > fwd$
3	+	-	-	$trad > fwd > fs > slb$
4	-	+	+	$slb > fs > fwd > trad$
5	-	+	-	$fwd > trad > slb > fs$
6	-	-	-	$fs > slb > trad > fwd$

Table 2.2: Theoretical comparisons of the methods.

where subscript fs and $trad$ represent FS and traditional SSL methods respectively.

y_{2slb}, y_{2trad}, and y_{2fs} need to be compared to evaluate their relative performance. This comparison can be done by taking derivatives to study their variation as when $x_1 = x_2$, $y_{2slb} = y_{2fs} = y_{2trad}$. More details can be found in the supplementary material.

Given a simplified case where $h(\cdot)$ is an identity map and x and y are positive systems, the relations of $(y_{2slb}, y_{2trad}, y_{2fs})$ and corresponding conditions are shown in Table 2.2. In general, the assumption of positive systems is reasonable in many real-world systems. It is observed that under certain conditions, the proposed SLB method is always better than the traditional SSL method, as long as the perturbation does not reverse the direction of the vector field that drives x. If h is not an identity map, the analysis needs to consider the properties of h. When h satisfies the conditions: 1) locally $h(x) \geq 0$ and $h(x)$ monotonically increases, or 2) locally $h(x) \leq 0$ and $h(x)$ monotonically decreases, the results in Table 2.2 are still determined. All the conditions here only need to be valid locally. For negative systems, the relations are a mirror image of Table 2.2 as shown in supplementary Table 2.6.

With the proof showing the advantageous retrospective self-labeling, an emerging question is the feasibility of reversely using causal data to infer interaction time for self-labeling effect data. This cause-based self-labeling is also analyzed, and represented with the subscript fwd. In this case, the inferred interaction time from x_1 is $t_{if} = A_{x_2} - A_{x_1}$. With perturbation, the self-labeling process will start from the timestamp when x_1 is received to infer forward,

self-labeling y_2 from the effect data stream. By substituting t in Eq. (2.9) with t_{if} here, we can derive

$$y_{2fwd} = e^{A_{x_2} - A_{x_1}} \cdot \left(\int_0^{A_{x_2} - A_{x_1}} e^{-\tau} h(B^{-1}(\tau + B_{x_1}))\, d\tau + y_1 \right) \tag{2.14}$$

which is compared under the same conditions in Table 2.2. It is observed that under conditions 3 and 4, where SLB and trad relation is undetermined, fwd is always better than trad.

2.3.2 An Example of DS

Given the proof in general forms, we provide an example of interacted DS. If $h(\cdot)$ is an identity map, examples of x_1-y_2 relation under conditions 1 to 4 in Table 2.2 are shown in Fig. 2.3, where $x_2 = 100$ and $y_1 = 10$. It can be observed that y_{2slb} is always closer to the ground truth than y_{2trad} in the given range in the first row, and in the second row, the forward self-labeling outperforms traditional SSL.

2.3.3 Connection to Discrete DS and GC

The above proof uses continuous-time DS. In practice, many systems are modeled in an ideal discrete form of $x(k+1) = f(x(k))$. When having two interacting DS, a form of coupling is $y(k+1) = g(y(k), x(k))$ [19]. In a simple linear coupling case, the system y becomes $y(k+1) = y(k) + x(k)$. While the above proof uses continuous DS, the conclusion still holds for discrete DS. By quantizing the time dimension, continuous DS can be easily converted to discrete DS. Thus, self-labeling can become closer to reality where object properties are often digitized and classified into finite states. Finite state machines or its variants can serve as object states modeling tools.

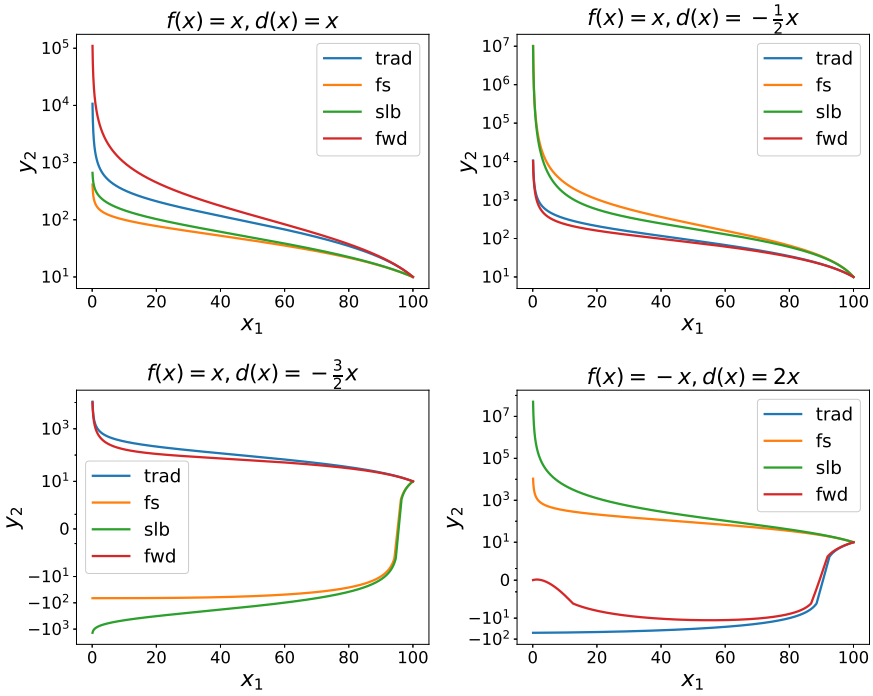

Figure 2.3: An illustration of x_1 and y_2 (in *log* scale) relation of two interacting DS derived from different methods.

Granger Causality (GC) [20] formally defines a statistical test of causal relations between two random variables represented by two time-series data. GC is predicated on the statement that the cause occurs before the effect. The standard GC [104] in a linear auto-regressive model is

$$Y_n = a_0 + \sum_{k=1}^{L} b_{1k} Y_{n-k} + \sum_{k=1}^{L} b_{2k} X_{n-k} + \xi_n \tag{2.15}$$

where ξ_n is uncorrelated noise and n is discrete step. It defines X "Granger Cause" Y. The GC formula is similar to coupled discrete DS where X and Y are two systems and the order L is 1. From the GC aspect, the causal and effect data can be quantized into distinct states, similar to discrete DS, for self-labeling. The self-labeling can also follow the form of GC where the order can be greater than 1. In this sense, more sequential causal states are involved and the effect state can self-label a sequence of data from the cause side.

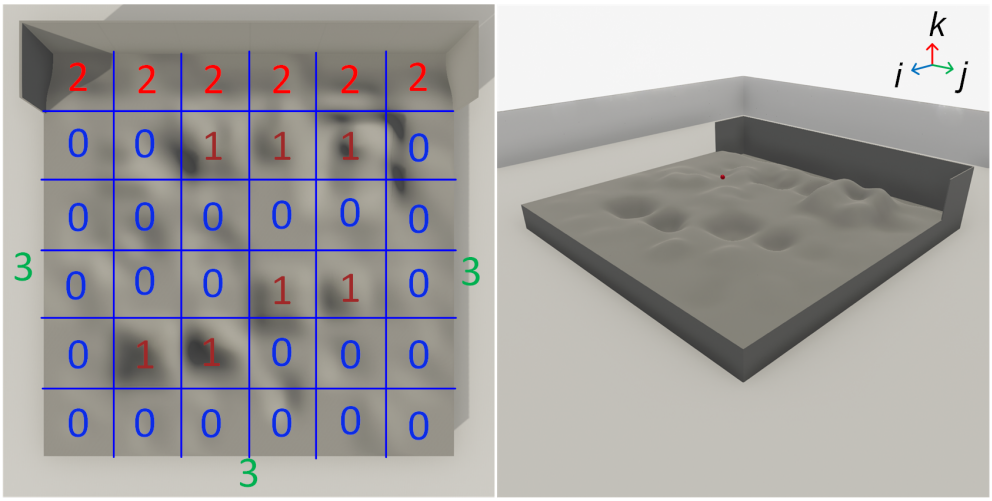

Figure 2.4: The designed landscape used in the simulation. Left side is the land block categorization and right side is an example of ball falling.

2.4 Experiment Results and Discussion

In this section, we design a computer simulation of object interactions to demonstrate the proposed self-labeling method using the TDW platform [42] based on Unity. The simulation drops a ball at a random location and observes its interactions with a finite ground surface region. The classical mechanical interactions between the ball and the ground include gravity, collision, and friction. A complex landscape with hills, bumps, holes, walls, and uneven surfaces is used in the simulation as shown in Fig. 2.4. The landscape is partitioned into 6×6 equal-sized blocks which are divided into 4 classes depending on each block's features. There are 23 blocks in class 0 (flat with minimal slope), 7 blocks in class 1 (indentation that traps balls), 6 blocks in class 2 (region contains wall), and the remainder of the simulated area excluding the landscape belongs to class 3. It is noted that a ball falling from a random location onto the land interacts with differing local topography. The ball can bounce and roll on the landscape and settle on or off the landscape. The ball's initial position is used as the cause data as it determines the potential energy and the possible trajectories of each ball. The effect data is defined as the ball's trajectory upon contact with land and the

Parameter	Description	Default value
k_0	Initial height range for sampling	$[10, 15]$
v_i, v_j	Ball's i/j axis movement speed before falling	0.05, 0.05
bounciness	The bounciness of the land in $[0, 1]$,	0.75
friction	The friction coefficient of the land in $[0, 1]$	0.25
w_i, w_j, w_k	Wind force vector	$(0.5, 0.5, -0.5)$

Table 2.3: Changeable Parameters in the Simulation

category of the region it settles in is used as the effect label. The ML task is to train a model that ingests the ball's initial position and predicts its final location category. The added perturbation is a wind applied to the ball randomly in the air. The friction and bounciness of each land block are adjustable to alter interactions with balls. The simulation also includes a mechanism where the number of balls accumulated on a land block scaled by predetermined linear coefficients will dynamically change the block's properties, increasing the complexity of the system. The changeable parameters in the simulation are summarized in Table 2.3.

In this experiment, we simulate two data streams that independently sense the ball's positions in the air to collect cause data and on the land for effect data. The class of the ball's final position and its rebound number can be derived by processing the effect data stream. In the task model, the land category is used as the label, while the 3-d coordinates of the final position and number of rebounds are used as the input for the ITM.

The ITM uses a gradient boosting decision tree (GBDT) with 500 estimators and 0.1 dropout rate as the regression model [73]. The task model is a multi-layer perceptron (MLP) of size $(32, 64, 128, 256, 128, 64, 32)$ with a ReLU after each linear layer implemented in PyTorch, optimized using SGD with 0.0005 weight decay and 0.01 learning rate [102]. The batch size is 128 with 600 epochs. The ITM and task model are chosen specifically for this simulation scenario using conventional ML development methods.

Dataset. The simulation generates a single ball and drops it from a random position sampled

from a 3-d uniform distribution where $i \in [-6, 6]$, $j \in [-6, 6]$, $k \in [10, 15]$. The dataset generated by the simulation is inherently imbalanced with class distribution 2.1:4.9:3.4:4.6 in the unperturbed case and 1.1:4.7:3.2:6.0 in the perturbed case with default simulation parameters. Therefore, a resampling is applied to balance classes, taking 1500 samples per class and 6000 samples overall.

Nested k-fold cross validation is applied to reduce data selection bias. We partition 6000 samples into 3 outer folds with 2000 samples each, selecting one outer fold as the test set and the remainder as training and validation sets. The remaining 4000 samples are partitioned into 5 inner folds with 800 samples each. One inner fold is used to train the ITM and pretrain the MLP. Then, 500 to 2500 samples from the four unused inner folds are used incrementally as self-labeled datasets to mimic drift adaptation, and the final 700 samples serve as the validation set. The outer and inner folds are rotated and averaged for model evaluation. When training other SSL models for comparison, the self-labeled datasets are used as unlabeled data.

As the interaction time inferred by the ITM can be longer than the ground truth, the initial position to be self-labeled may not exist in the cause data stream. A mechanism is added to resolve this issue by making the ball move horizontally before falling so that there are corresponding ball positions to be self-labeled. The horizontal movement is controlled by two coefficients in Table 2.3 with random direction. With default parameters, the average R^2 score on test sets of the trained ITM is 0.84 and mean absolute error is 36.4, resulting in an average horizontal offset of 1.7, which is reasonable, as the average inaccuracy in the inferred interaction time equates to an offset of almost one block from the ground truth initial position.

First, we consider the experimental results of the unperturbed case shown in Table 2.4 generated with the default parameters in Table 2.3. The self-labeling method is compared with several recent semi-supervised models implemented in TorchSSL [154] and USB [142].

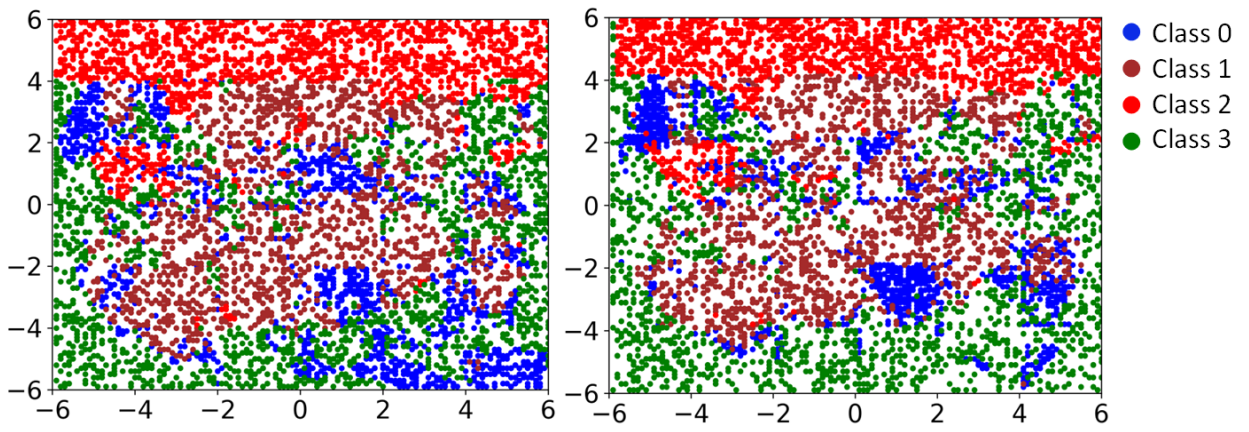

Figure 2.5: The data distributions of input and output labels. The plot shows the initial horizontal positions of balls and the corresponding labels with wind perturbation (left) and without (right).

As these methods were tested on image recognition datasets in the original papers, we adapt these methods in this work to our dataset where the input data is a vector $[i, j, k]$. From Table 2.4, it can be observed that the self-labeling method maintains comparable performance as other SSL methods across 5 unlabeled dataset sizes without domain shift. The self-labeling method also shows an increasing accuracy trend with more self-labeled data similar to the FS method, while other SSL methods do not benefit from enlarging the unlabeled dataset.

The results with perturbation (wind) applied are shown in Table 2.5. The wind is applied at a random time during initial 60 frames of ball falling. It is apparent from the comparison in Fig. 2.5 between the left and right sides that the input-output mapping changes, simulating the concept shift. For other SSL methods, the training data combines a labeled unperturbed dataset and an unlabeled perturbed dataset with data samples identical to the pretraining and self-labeled set used in the self-labeling method. With such perturbation, SSL methods based solely on feature similarity and smoothness show no significant improvement. Contrarily, the self-labeling method maintains higher accuracy and gradually improves in accuracy with more data, indicating potential for lifelong adaptive learning. This experiment validates the theoretic proof in Section 2.3.1. Moreover, this experiment uses 3-d data with complex

	500	1000	1500	2000	2500
PseudoLabel [79]	73.8	74.0	74.1	74.1	74.0
MixMatch [15]	68.3	68.0	68.0	68.1	68.3
FixMatch [124]	74.3	74.2	74.0	74.4	74.0
FlexMatch [154]	74.1	74.7	74.8	74.4	74.6
PseudoLabelFlex [154]	74.2	74.2	74.3	74.2	74.0
SimMatch [158]	72.8	72.5	72.8	72.7	72.7
SoftMatch [23]	74.7	75.0	74.7	74.5	75.0
FreeMatch [143]	74.3	74.6	74.7	74.7	74.7
SLB (no pretrain)	62.9	65.9	67.7	69.3	70.0
FS (no pretrain)	67.9	74.2	77.1	78.7	79.8
SLB ($v = 0.05$)	72.7	73.3	74.5	74.9	74.8
SLB ($v = 0.1$)	73.2	73.2	74.4	74.8	75.3
SLB ($v = 0.15$)	72.8	73.0	75.0	75.3	75.6
FS	75.7	77.3	79.7	80.3	81.0

Table 2.4: Model Accuracy (%) trained on unperturbed dataset

	500	1000	1500	2000	2500
PseudoLabel [79]	69.1	69.1	69.1	69.1	69.0
MixMatch [15]	66.1	65.9	66.0	66.3	65.8
FixMatch [124]	68.9	69.1	69.0	68.9	69.0
FlexMatch [154]	69.4	69.3	69.4	69.6	69.7
PseudoLabelFlex [154]	69.2	69.5	69.4	69.4	69.8
SimMatch [158]	68.4	68.0	68.3	68.3	68.2
SoftMatch [23]	69.2	69.5	69.6	69.5	69.7
FreeMatch [143]	69.5	69.6	69.5	69.4	69.5
SLB (no pretrain)	64.2	67.4	69.4	71.2	72.3
FS (no pretrain)	71.1	76.1	78.0	79.2	79.9
SLB ($v = 0.05$)	70.8	72.4	73.5	74.3	74.4
SLB ($v = 0.1$)	71.1	72.2	73.3	74.2	74.5
SLB ($v = 0.15$)	71.4	73.0	73.8	74.2	74.8
FS	74.4	76.3	77.7	78.9	79.4

Table 2.5: Model Accuracy (%) adapted on perturbed dataset

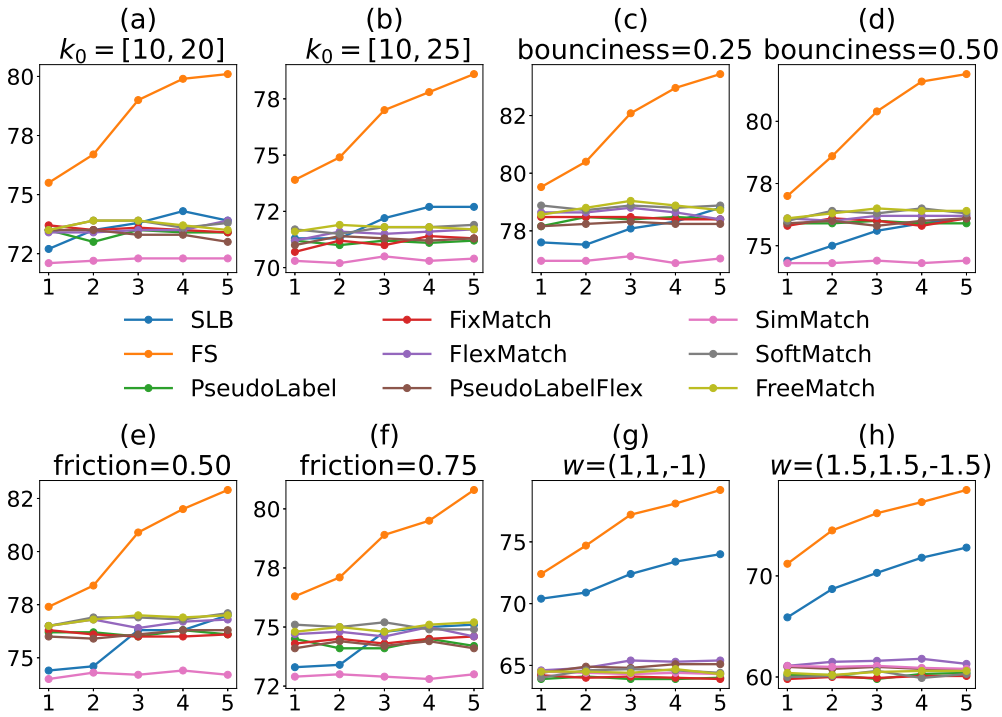

Figure 2.6: Test results varying other simulation parameters. The changed parameter is listed in each plot's title while the rest are unchanged. y-axis is accuracy (%) and x-axis is the incremental number of SLB set. Each increment is 500 samples. (a)-(f) are unperturbed.

internal interaction mechanisms, indicating that the self-labeling method is not limited to the 1-d case shown in the proof.

Additionally, we test the SLB method without pretraining the task model and with different v_x and v_y ($v_x = v_y$) in Tables 2.4 and 2.5. Without pretraining, self-labeling has a greater impact on accuracy. When v, the penalty for incorrect interaction time inference, is increased, the SLB method's performance still exceeds other SSL methods in perturbed cases even with $v = 0.15$, causing an average horizontal shift of 5.0, more than 2 blocks of distance.

For further validation, more thorough experiments are conducted with different simulation parameters as shown in Fig. 2.6 to demonstrate that regardless of simulation parameters, the self-labeling method maintains superior performance, congruent with our theory. With more intense perturbations, as shown in Fig. 2.6(g) and (h), the performance of other SSL methods drops about 10% or more compared to the unperturbed case in Table 2.4, while the

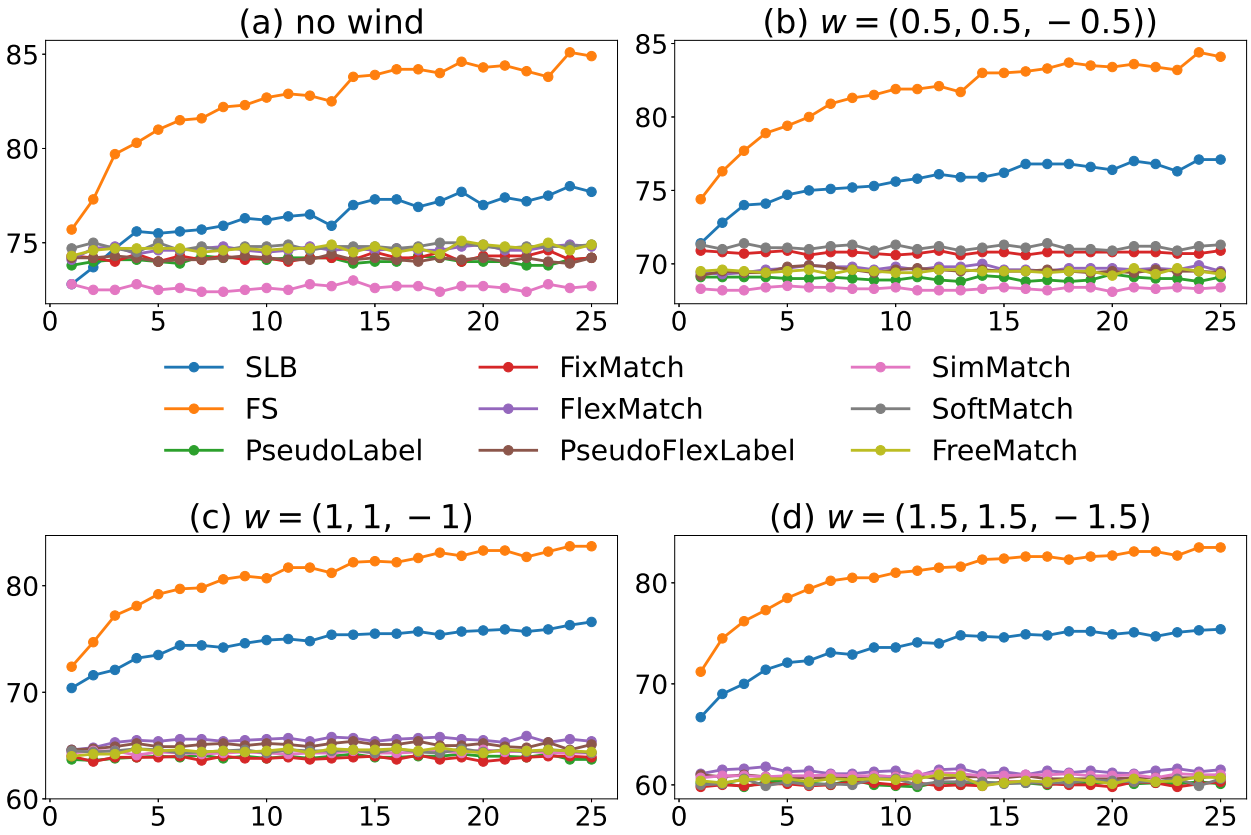

Figure 2.7: Test results with 25 increments of self-labeled datasets with 500 samples in each increment. y-axis is accuracy (%) and x-axis is the incremental number. (a) is tested without wind while (b) to (d) are tested with different wind magnitudes. The rest simulation parameters are default.

self-labeling method maintains a similar accuracy to the original domain and shows potential to improve in accuracy with more data.

Fig. 2.7 provides experimental results with 25 increments (up to 12500 samples) of self-labeled data to further validate accuracy trends in four cases: the unperturbed case and three perturbed cases with different wind magnitudes. Comparing Fig. 2.7(a) and Table 2.4, the accuracy of the self-labeling method continues to improve with additional increments of self-labeled data and rises to outperform other SSL methods. This observation further confirms the merit of self-labeling over SSL methods, even in applications without domain shifts. These results reveal the potential of the proposed self-labeling method to achieve autonomous adaptive learning.

2.5 Discussion

Dependent or Independent Perturbation. Most dynamical system analyses in literature treat perturbations as a function of time due to the assumption of independence. In the proof, we define $d(x)$ to keep $\dot{x}$ homogeneous. From the ML perspective, the perturbation term simulates the distribution difference between training data and real data, *i.e.*, concept drift. We use the same perturbation nomenclature as DS theory, but with a distinct physical meaning. The independence of perturbation does not affect the simulation of concept drift, given that the input-output relation is changed. For example, d can be in various forms such as constants, piece-wise functions, or impulse functions, where the changepoint conditioned on t can be converted to x since the boundary of interaction is defined. Supplementary Fig. 2.9 shows an example of $d(t)$ where the relative relations of the four methods still hold, although more comprehensive proofs can be accomplished in the future.

Extension to n-d DS. The theoretical analysis of self-labeling in this study applies only to 1-d cases, whereas the experiment demonstrates its effectiveness with high-dimensional data. An example of interacted 2-d DS is given in supplementary Fig. 2.10. An n-d DS is intrinsically composed of coupled 1-d state variables. Depending on the boundary definition of the two DS and their synchronization, two interacted 1-d DS can also be viewed as a 2-d internally coupled system. Due to increased complexity and ambiguous definition of interaction boundary [128, 129, 93], further research is needed to extend to n-d DS interactions.

In Real ML Application. In the theoretical analysis, we assume that an ML model can ideally learn a function given inputs and outputs. In practice, however, trained ML models may not generalize well with respect to the input range, which manifests a practical problem in that self-labeling methods can suffer from biased input data ranges. In Fig. 2.3(a), the self-labeled causal data range changes slightly from $[0.1, 100]$ to $[0.22, 100]$. Given another example with $\dot{x} = x, \dot{y} = 2x$, and $d(x) = 1$, x_{slb} range changes to $[3.7, 100]$. This change

in data range can marginally impair self-labeling performance in practice. Considering its superior performance over most of the regions, the self-labeling method still owns its merit.

Connection to Control/RL. In control systems and Reinforcement Learning (RL), *e.g.*, robot learning, DS and interactions are also widely discussed. RL leverages interactions between control agents and their environments to allow agents to learn from interactive trials with designed reward functions. Our method and RL-based control utilize both interactions and feedback in the form of either effects or rewards caused by the interactions in the model learning. In RL for robot-object interactions, for example, the objective is to adapt a new robot control strategy such that the learning output will improve robot behaviors. In our self-labeling method, the system stands away and observes interactions from two channels, but will not interfere with interactions governed by their own dynamics. Our objective is to adapt robust ML models for the recognition of cause or effect states without imposing any control over agents, which delineates the interactions in self-labeling from RL.

Why Causality. Our system builds on causation rather than correlation for several reasons. Causality, especially causal direction, is more consistent across domains than correlation. Correlation is strongly associated with probability, while causality possesses greater physical regularity. From a physics perspective, in the Minkowski space-time model, causality is preserved in the timelike light cone regardless of the observers' reference frames [16], making its invariance more theoretically sound than that of correlation. The directionality of causation more explicitly characterizes state relations and time lags in that cause always precedes effect, a principle leveraged in this work.

Multi-Variable Causality. In the proof and experiment, we consider the causality of two variables. Scenarios with more than two variables complicate the causal structure, inducing fork, collider, or confounder cases. With a collider, each cause can have a different interaction time [18], thus more ITMs can be trained to infer each interaction time. Multiple effects can also jointly infer interaction time and generate labels with a fork. Moreover, variables can be

$$\frac{dy_{2trad}}{dx_1} = -A'_{x_1} e^{A_{x_2}-A_{x_1}} \Big(\int_0^{A_{x_2}-A_{x_1}} e^{-\tau} h(A^{-1}(\tau + A_{x_1}))\, d\tau + y_1 \Big) + e^{A_{x_2}-A_{x_1}} \Big(e^{-A_{x_2}+A_{x_1}} h(A^{-1}_{A_{x_2}})(-A'_{x_1}) + \int_0^{A_{x_2}-A_{x_1}} \frac{d(e^{-\tau} h(A^{-1}(\tau + A_{x_1})))}{dx_1}\, d\tau \Big) \quad (2.16)$$

$$\int_0^{A_{x_2}-A_{x_1}} \frac{d(e^{-\tau} h(A^{-1}(\tau + A_{x_1})))}{dx_1}\, d\tau) = \int_0^{A_{x_2}-A_{x_1}} e^{-\tau}\, d\tau A'_{x_1} \frac{d(h(A^{-1}(\tau + A_{x_1})))}{d(\tau + A_{x_1})} \quad (2.17)$$

$$\frac{dy_{2trad}}{dx_1} = -y_1 A'_{x_1} e^{A_{x_2}-A_{x_1}} - A'_{x_1} h(x_2) + A'_{x_1} e^{A_{x_2}-A_{x_1}} \int_0^{A_{x_2}-A_{x_1}} e^{-\tau} \Big[\frac{d(h(A^{-1}(\tau + A_{x_1})))}{d(\tau + A_{x_1})} - h(A^{-1}(\tau + A_{x_1})) \Big]\, d\tau \quad (2.18)$$

separated for analysis if their corresponding state transitions can be derived and smoothed on a temporal scale.

2.6 Appendix

2.6.1 Detailed Derivation of Theoretical Proof

In this section, we will give detailed steps of the derivation in getting the results of Table 2.2.

First, we will use y_{2trad} as an example to clarify the steps in calculating its derivative. Given Eq. (2.12) to derive $\frac{dy_{2trad}}{dx_1}$, first we can directly take its derivative to get Eq. (2.16). The last integral term inside the bracket can be converted to by using chain rule as Eq. (2.17). Then, we can substitute Eq. (2.17) in Eq. (2.16) and reorganize the equation to derive Eq. (2.18)

The last integral term of Eq. (2.18) is equal to

$$e^{-\tau}h(A^{-1}(\tau + A_{x_1}))\Big|_0^{A_{x_2}-A_{x_1}}. \tag{2.19}$$

Hence, Eq. (2.18) will become

$$\frac{dy_{2trad}}{dx_1} = -A'_{x_1}e^{A_{x_2}-A_{x_1}}(y_1 + h(x_1)) \tag{2.20}$$

Similarly, Eqs. (2.21) and (2.22) are derived. It can be observed that when $x_1 = x_2$, $y_{2slb} = y_{2fs}$. If we view x_{slb} same as x_1, then the only difference is the last term inside bracket which will be compared in later sections.

$$\frac{dy_{2fs}}{dx_1} = -B'_{x_1}e^{B_{x_2}-B_{x_1}}(y_1 + h(x_1)) \tag{2.21}$$

$$\frac{dy_{2slb}}{dx_{slb}} = -B'_{x_{slb}}e^{B_{x_2}-B_{x_{slb}}}(y_1 + h(A^{-1}(A_{x_2} - B_{x_2} + B_{x_{slb}}))). \tag{2.22}$$

Additionally, to compare (y_{2trad}, y_{2slb}) we can rewrite Eqs. (2.11) and (2.12) by using integration by substitutions to make the terms under integral same, and get

$$y_{2trad} = e^{A_{x_2}}\int_{A_{x_1}}^{A_{x_2}} e^{-u}h(A^{-1}(u))\,du + e^{A_{x_2}-A_{x_1}}y_1 \tag{2.23}$$

$$y_{2slb} = e^{A_{x_2}}\int_{A_{x_2}-B_{x_2}+B_{x_{slb}}}^{A_{x_2}} e^{-u}h(A^{-1}(u))\,du + e^{B_{x_2}-B_{x_{slb}}}y_1. \tag{2.24}$$

In the following we will use the condition 3 in Table 2.2 as an example to go through the comparison of the relative relations of $y_{2trad}, y_{2slb}, y_{2fs}, y_{2fwd}$. We will still assume that h is an identity map and positive systems.

Compare Trad and FS. Initially, with the conditions, we can derive a series of function properties: $B(x) \leq 0, A(x) \geq 0,\ A(x) \uparrow, B(x) \downarrow, B^{-1}(\tau) \downarrow, B^{-1}(x) \geq 0$. Here $\uparrow$ and $\downarrow$ represent locally monotonically increase and decrease respectively. Eqs. (2.20) and (2.21) can be compared to derive the relative relation between y_{2trad} and y_{2fs} since at the boundary of $x_1 = x_2$, $y_{2trad} = y_{2gt}$. It can be easily derived that $A_{x_2} - A_{x_1} \geq B_{x_2} - B_{x_1}$ by comparing the slopes of A_{x_1} and B_{x_1}.

For Eq. (2.20), since $A'_{x_1} \geq 0$, $\frac{dy_{2trad}}{dx_1} \leq 0$, which means that $y_{2trad} \downarrow$. For Eq. (2.21), since $B'_{x_1} \leq 0$, $\frac{dy_{2fs}}{dx_1} \geq 0$, which means that $y_{2fs} \uparrow$. When $x_1 = x_2$, $y_{2trad} = y_{2fs}$, thus $y_{2trad} \geq y_{2fs}$ in the given range.

Compare Fwd and FS. y_{2fwd} and y_{2fs} can be compared by using Eqs. (2.13) and (2.14). Since $A_{x_2} - A_{x_1} \geq B_{x_2} - B_{x_1}$, the integral range of y_{2fs} is smaller than that of y_{2fwd}. The exponential term of y_{2fs} is also smaller than that of y_{2fwd}. Therefore, we can get $y_{2fwd} \geq y_{2fs}$.

Compare Trad and Fwd. Given Eqs. (2.12) and (2.14) to compare y_{2trad} and y_{2fwd}, we can observe that the only difference is the integrand. Thus only the integrand in these two functions need to be analyzed under defined conditions. As $A^{-1}(x) \uparrow$ and $B^{-1}(x) \downarrow$, and when $\tau = 0$, $A^{-1}(A_{x_1}) = B^{-1}(B_{x_1}) = x_1$, we can derive that when τ getting larger, A^{-1} goes up and B^{-1} goes down. Thus for the same bounds of integration, $y_{2trad} \geq y_{2fwd}$

Compare Trad and SLB. Given Eqs. (2.23) and (2.24) to compare y_{2trad} and y_{2slb}, the differences are the integration bounds and the last exponential term. x_{slb} can be treated equally as x_1 during comparison. Since $A_{x_2} - B_{x_2} + B_{x_1} \geq A_{x_1}$, the integral bounds and the last exponential term of y_{2trad} are greater than those of y_{2slb}. Given $A^{-1} \geq 0$, we can derive $y_{2trad} \geq y_{2slb}$.

Compare SLB and FS. y_{2slb} and y_{2fs} can be compared by Eqs. (2.21) and (2.22). The only difference is the last term in bracket. Under condition 3, $A_{x_2} - B_{x_2} + B_{x_1} \geq A_{x_1}$ and $A^{-1} \geq 0$. Thus for the bracket term, SLB is greater than FS. Since the two equations are

positive as $B^{'} \leq 0$, thus Eq. (2.22) is greater than Eq. (2.21), which means that $y_{2fs} \geq y_{2slb}$.

With the above examples of the detailed comparison under condition 3 in Table 2.2, other comparison can be accomplished in a similar way. Note that when comparing integrals, the sign of integrands needs to be taken into account.

Equations of the examples. The equations for the case of where $f(x) = x, d(x) = x$ in Fig. 2.3 are:

$$y_{2slb} = x_2 \cdot log(\sqrt{\frac{x_2}{x_1}}) + y_1\sqrt{\frac{x_2}{x_1}} \tag{2.25}$$

$$y_{2trad} = x_2 \cdot log(\frac{x_2}{x_1}) + \frac{x_2}{x_1}y_1 \tag{2.26}$$

$$y_{2fs} = x_2 - \sqrt{x_1 x_2} + y_1\sqrt{\frac{x_2}{x_1}} \tag{2.27}$$

$$y_{2fwd} = \frac{x_2}{x_1}(x_2 - x_1 + y_1) \tag{2.28}$$

The equations for the case of where $f(x) = x, d(x) = -\frac{x}{2}$ in Fig. 2.3 are:

$$y_{2slb} = x_2 log\frac{x_2^2}{\sqrt{x_1}} + y_1\frac{x_2^2}{\sqrt{x_1}} \tag{2.29}$$

$$y_{2trad} = x_2 \cdot log(\frac{x_2}{x_1}) + \frac{x_2}{x_1}y_1 \tag{2.30}$$

$$y_{2fs} = \frac{x_2^2}{\sqrt{x_1}}y_1 + 2\frac{x_2^2}{x_1} - 2x_2 \tag{2.31}$$

$$y_{2fwd} = y_1\frac{x_2}{x_1} + 2x_2 - 2\sqrt{x_1 x_2} \tag{2.32}$$

An example is given with $f(x) = x, d(x) = -\frac{3x}{2}$ to illustrate their relative relations. In this

example, we can derive

$$y_{2slb} = x_2 log(\frac{x_1}{x_2})^2 + (\frac{x_1}{x_2})^2 + y_1 \tag{2.33}$$

$$y_{2trad} = x_2 \cdot log(\frac{x_2}{x_1}) + \frac{x_2}{x_1} y_1 \tag{2.34}$$

$$y_{2fs} = -\frac{2}{3}x_2 + \frac{2x_1}{3}(\frac{x_1}{x_2})^2 + (\frac{x_1}{x_2})^2 y_1 \tag{2.35}$$

$$y_{2fwd} = -\frac{2x_2}{3}(\frac{x_2}{x_1})^{-\frac{3}{2}} + \frac{2}{3}x_2 + \frac{x_2}{x_1} y_1 \tag{2.36}$$

For the case of where $f(x) = -x, d(x) = 2x$, the equations are:

$$y_{2slb} = \frac{x_2}{x_1}(-\frac{x_1}{2} + \frac{x_2^2}{2x_1} + y_1) \tag{2.37}$$

$$y_{2trad} = (y_1 + x_1/2)\frac{x_1}{x_2} - x_2/2 \tag{2.38}$$

$$y_{2fs} = x_2 log\frac{x_2}{x_1} + y_1\frac{x_2}{x_1} \tag{2.39}$$

$$y_{2fwd} = \frac{x_1}{x_2}(x_1 log\frac{x_1}{x_2} + y_1) \tag{2.40}$$

2.6.2 Negative Systems

When x and y are negative systems, the relative relations of the four methods are summarized in Table 2.6 with $h(\cdot)$ as an identity map. It can be observed that for negative systems, the conclusion that SLB is better than trad is strictly satisfied only under condition 9 and 10, which is a mirror image of the conclusions with positive systems. While for other conditions in Table 2.6 the relative performance of SLB and trad over FS is obscure, it can be observed that the forward self-labeling method obviously performs better than traditional SSL.

In addition, the positivity of system x and y do not need to be consistent. x and y can have different signs. In such cases, the relative relations of the four methods will depend on the values of x and y. As in real world systems the system states are positive in many cases, the

ID	$f(x)$	$d(x)$	$f(x)+d(x)$	Relation
7	+	+	+	$slb > fs > fwd > trad$
8	+	-	+	$trad > fwd > fs > slb$
9	+	-	-	$fs > slb > trad > fwd$
10	-	+	+	$fwd > trad > slb > fs$
11	-	+	-	$slb > fs > fwd > trad$
12	-	-	-	$trad > fwd > fs > slb$

Table 2.6: Theoretical comparisons of the methods given negative systems.

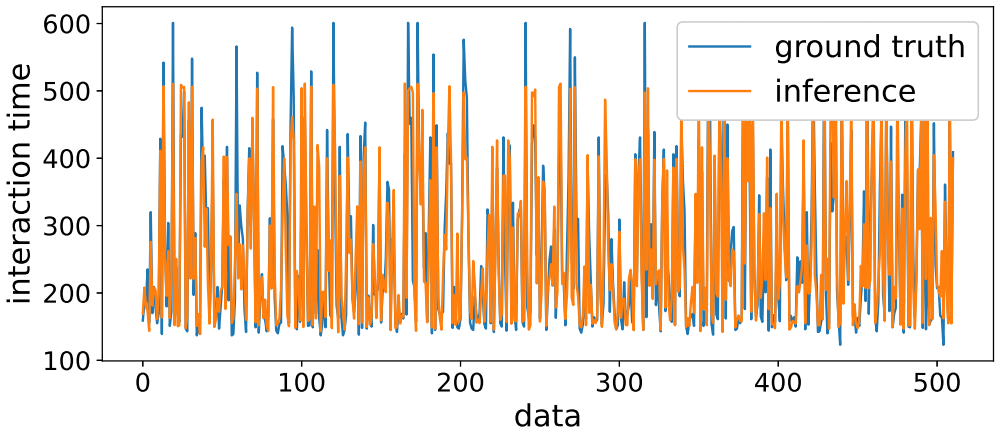

Figure 2.8: An example of the ITM performance on dataset using the default simulation parameters.

merit of self-labeling can play an important role in designing adaptive and continual ML.

2.6.3 Additional Examples and Experiment Illustration

Fig. 2.8 displays the ITM inference performance on the test set with default simulation parameters in Section 4.4.

In Fig. 2.9, a DS example with $d(t)$ is given where $f(x) = x, d(t) = at$, and $\dot{y} = y + x$. It can be seen that the relative relations of the methods still hold regardless of the independence of perturbation term. Due to the challenge in analytically solving non-homogeneous differential equations, a rigorous proof of using $d(t)$ will need further research.

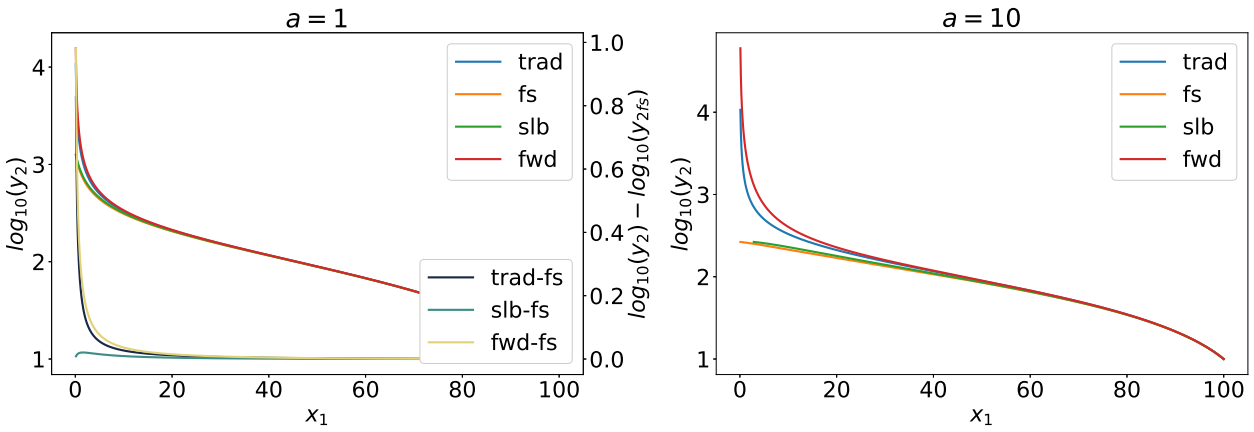

Figure 2.9: Examples of interacted DS when the perturbation is $d(t) = at$. For easier view, in the left the differences between FS and others are plotted.

Fig. 2.10 provides a visualization of two 2-d interacted dynamical systems. It plots the input-output relations derived from traditional SSL, FS, and SLB methods, where the propagation of system $\boldsymbol{x} = (x_1, x_2)^T$ and $\boldsymbol{y} = (y_1, y_2)^T$ are defined as

$$\boldsymbol{x}' = \begin{pmatrix} 1 & 2 \\ & \\ 1 & 2 \end{pmatrix} \boldsymbol{x} + \boldsymbol{d} \tag{2.41}$$

$$\boldsymbol{y}' = \begin{pmatrix} 1 & 1 \\ & \\ 1 & 1 \end{pmatrix} \boldsymbol{y} + \boldsymbol{x}. \tag{2.42}$$

The perturbation vector $\boldsymbol{d}$ is defined as $(0, 0)^T$ in unperturbed case and $(1, 1)^T$ in perturbed case. The boundary of interactions are defined. In this example, the initial values of $\boldsymbol{x}$ and $\boldsymbol{y}$ are $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$. The final values are $\boldsymbol{x_f}$ and $\boldsymbol{y_f}$. The cause and effect in this interactive system are $\boldsymbol{x_i}$ and $\boldsymbol{y_f}$ respectively. We define $\boldsymbol{y_i} = (1, 1)^T$ and $x_{1f} = 10$ while x_{2f} can be derived given these constraints. As shown in Fig. 2.10, the SLB method still owns its advantage in this example compared over other traditional SSL methods in high dimensional data.

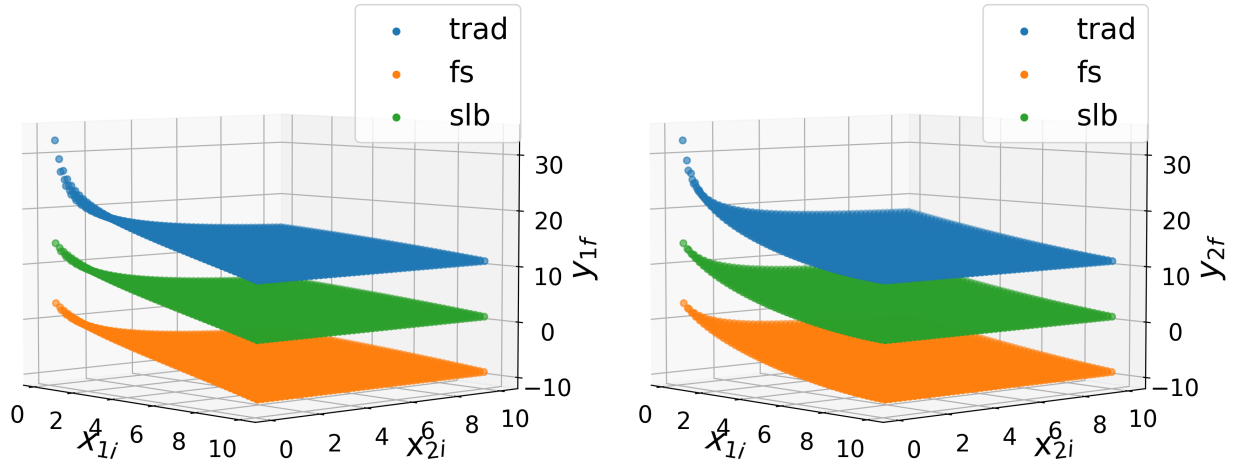

Figure 2.10: An example of 2D interacted DS. For easier view, the differences among the three methods are increased by 10.

Chapter 3

Interactive Causality Methodology and Applications in Complex Causal Structures

3.1 Introduction

Interactive causality based self-labeling has been elaborated and demonstrated in Chapter 2. The core idea of self-labeling is to use causation and learnable causal time lag to associate data streams for autonomous data annotation, which can be used to adapt machine learning models to local environments. Compared with traditional semi-supervised methods, the self-labeling targets at realistic scenarios with streaming data and is more theoretically sound for countering domain shifts without the need of manual data collection and annotation.

In the self-labeling derivation in Chapter 2, we made an assumption that there is an identified causal relationship between two variables to assist the selection of additional observing channel. A remaining question is how to find or identify a causal relationship given a machine

learning task. Causality owns a stronger definition than correlation with less stochasticity and thus requires more strict verification. For example in clinical research, the randomized controlled trial is a main method to verify causal effects of medical treatments. In statistics, several causal discovery methods have been developed to test conditional independence among variables [159, 75, 86, 65], which empowers data-driven methods to explore causality only from observational data and simplifies causal discovery. While these methods are useful, obtaining causality solely for the purpose of applying self-labeling can be expensive. Domain knowledge is a rich source for understanding a problem and its contexts. Typically domain knowledge is summarized by domain experts and contains rich causation. From the perspective of ML development for real-world problems, domain knowledge is also essential for understanding ML problems and selecting relevant data features. Therefore, we propose to extract existing causality from domain knowledge and build a complete methodology to enable the development of self-labeling assisted applications. Additionally, how to gain new knowledge in the form of causation is of great interests to assist anomaly understanding and adaptation. Newly acquired knowledge can be represented as additional nodes connected to an existing knowledge graph and used for application development.

To answer these questions and complete the interactive causality methodology, in this chapter we propose to extract and model causality from existing human knowledge in a domain. A knowledge modeling framework is adopted for a domain using the concept of ontology and knowledge graphs with embedded causality among interactive nodes. This knowledge modeling framework emulates how humans establish perception for a domain from static to dynamic relationships. It provides a pathway to build and model causal knowledge to obtain additional observing channels for self-labeling. In addition, a workflow to acquire new knowledge and expand knowledge graph is introduced utilizing the concept of interactive causality. We use existing techniques including unsupervised pattern recognition and data-driven causal discovery algorithms to identify new knowledge. It is confirmed that the interactive causality can also assist knowledge acquisition leveraging temporal causal events

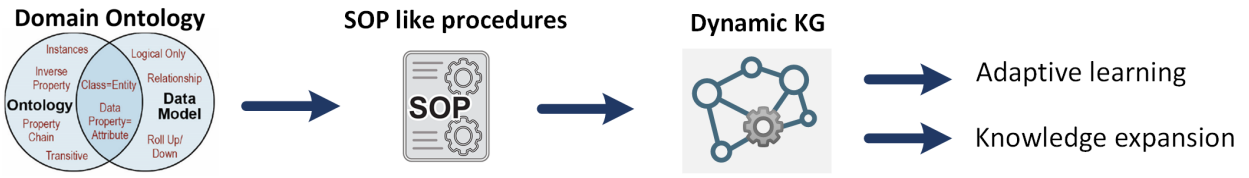

Figure 3.1: A pipeline of knowledge modeling in a domain.

and provide an alternative way to grow knowledge graph.

Additionally, Chapter 2 only provides theory and simulation in a simple causal structure with one cause and one effect variable. In this chapter we explore how self-labeling can be applied to complex causal structures with multiple variables. The combination and manipulation of individual interaction time is discussed in four basic cases which can be extended to complex structures composed of basic ones. A simulation utilizing a physics engine is conducted to show that the self-labeling is still applicable and effective with a complex causal graph.

3.2 Interactive Causality Methodology

The proposed interactive causality (IC) methodology consists of several steps for two objectives: adaptive machine learning and knowledge expansion. Both objectives step on the idea of IC and the importance and physical meaning of interaction time. As the self-labeling has been illustrated in Chapter 2, this section will focus on the knowledge modeling for causality extraction and knowledge graph expansion.

3.2.1 Knowledge Modeling Framework

Knowledge modeling emulates in general how humans build up knowledge towards a new domain for accomplishing tasks as shown in Fig. 3.1. Initially given a new domain, we can build up a static perception of the entities, attributes, and relations in this domain. Usually,

this static perception can be transformed into a static ontology model. To accomplish a certain task in this domain, a person can utilize the static perception and develop a sequence of steps, which can be modeled as a knowledge graph incorporating dynamic and temporal interactions among multiple entities. Similar to daily cooking, a person unaware of cooking can initially build up a static perception of a kitchen in terms of entities, attributes (*e.g.*, functions of utensils), and connections. To cook a dish, the apprentice develops a sequence of steps by applying the static perception, which formulates a standard operating procedure (SOP) for a certain recipe incorporating dynamic and temporal interactions among multiple entities. This human learning and knowledge acquisition paradigm can be abstracted into multiple layers of perception and combined into the AI knowledge (*i.e.*, data and label) learning to substantially reduce the cost of knowledge transfer. Ontology models represent entities, properties of entities, and relationships between entities in a domain. In manufacturing, the knowledge for a machine initially can be represented as a static ontology model that describes static properties of its functions, components, connections, control logic, process parameters, and their relationships. The static ontology model of a machine can be utilized by engineers to develop multiple SOPs for specific processes or recipes encompassing dynamic information. This dynamic SOP can be modeled into multiple interconnected dynamic causal knowledge graphs (DCKG) representing the state transitions and underlying causality of interactive components including machines, materials, humans, environments, and cyber space.

Fig. 3.2 provides a simple example of knowledge modeling in semiconductor manufacturing. We can build general ontology models for materials, machines, and workers to describe their static properties. In this example, the knowledge of a worker's capability includes their sensing and actuation functions, the knowledge of a machine includes its components and connection, and the knowledge of materials (silicon wafers) is the state transformation or chemical reactions. As required by the SOP, a worker can twist a gas cylinder or type on a computer to operate the machine. Thus, these interactions will build up connections among

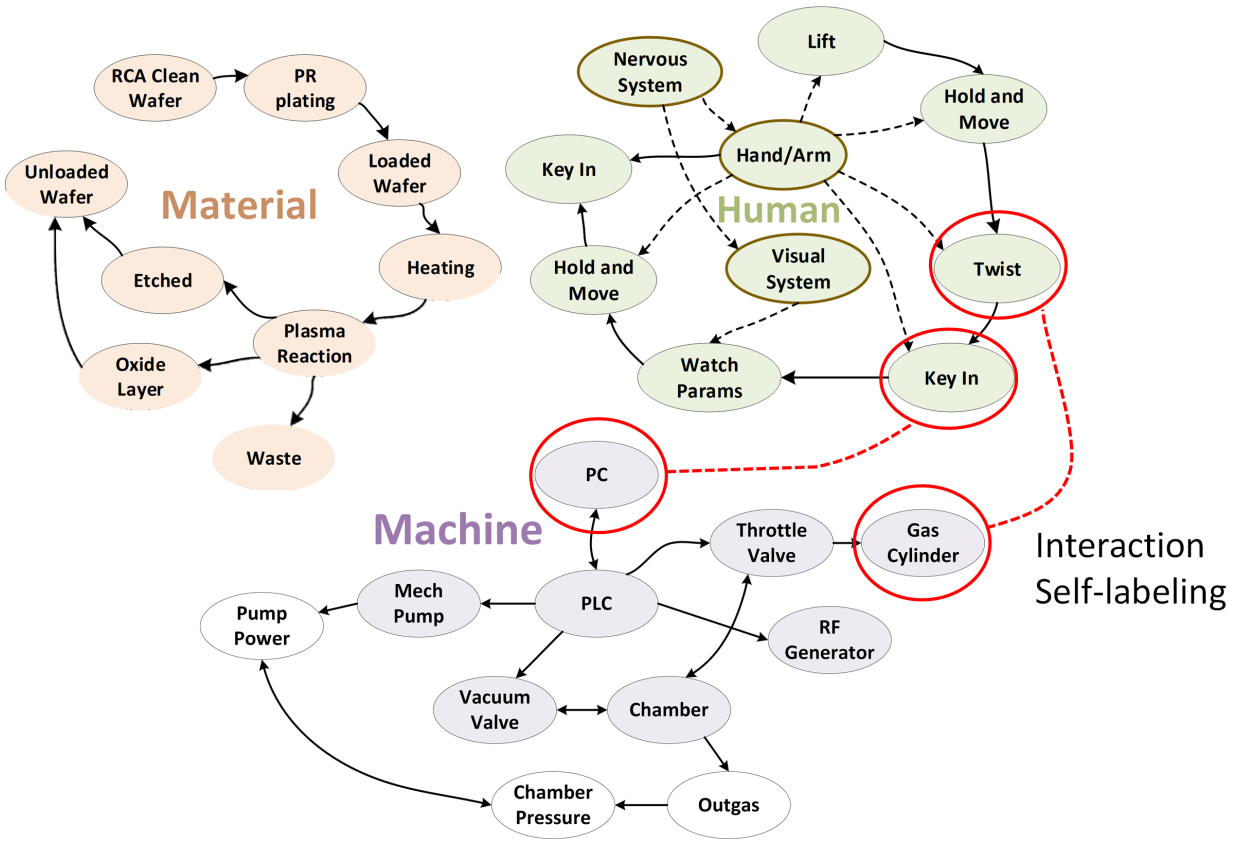

Figure 3.2: An example of three knowledge graphs for materials, machines, and workers in PECVD semiconductor manufacturing with interactive nodes.

the nodes in the worker's ontology model and in the machine's ontology model as indicated by the red lines in Fig. 3.2. The connections also represent the underlying causality embedded in such interactions across graphs. These interaction nodes across different knowledge graphs build up connections between events and states that can lead to automatic association of data streams via learnable interaction time for following self-labeling.

3.2.2 Knowledge Graph Expansion

The proposed knowledge modeling workflow only represents existing knowledge. An emerging question is how to gain new knowledge by using the interactive causality. We extend interactive causality to add new nodes by using some existing technologies including unsu-

pervised pattern recognition and causal discovery. A proposed workflow is shown in Fig. 3.3.

First, the data distribution shift (*i.e.*, concept drift) caused by an unknown factor needs to be identified by IC using interaction time as a sampling window. The concept drift can be observed by detecting the inconsistency between inputs and outputs. In this case, given the same input, the effect signal might be different or the interaction time may be different induced by an unknown cause. Intuitively, humans can sense this type of anomaly by using the causal time interval. For example, when we use a remote control to turn on a TV, we will hit the button, wait for a few seconds, and expect screen wake-up or some sound. If after a few seconds the expected effects do not show up, we will realize this is an anomaly and trace back to think about what happened earlier. In this example, humans can use the causal time lag as a way to find out the inconsistency between causes and effects. This process can also be replicated into a workflow as shown in Fig. 3.3 (a). To match with human intuition, the forward inference of interaction time by using cause data is utilized. The inferred interaction time can be intentionally extended to wait for the perturbed effect fully showing up. Then, an effect sample from the effect data stream can be selected based on the extended interaction time to build a self-labeled data pair. Next, a data distribution measure between the perturbed and the original dataset can be applied to find out the distribution distance. To confirm existence of an anomaly, the distance shift needs to be greater than a threshold. The threshold comes from an observation that the distances among a number of sampled datasets in the unperturbed original domains are not 0 due to randomness and there will be a variance which can be used as the threshold. By extending the inferred interaction time, the perturbed effects can fully show up and generate a higher distribution shift than the predefined threshold for anomaly confirmation. The logic behind this technique is that the interaction time is related to the magnitude of effect signals. With perturbation the interaction time changes and thus the expected effect state may show up at an extended or shortened time.

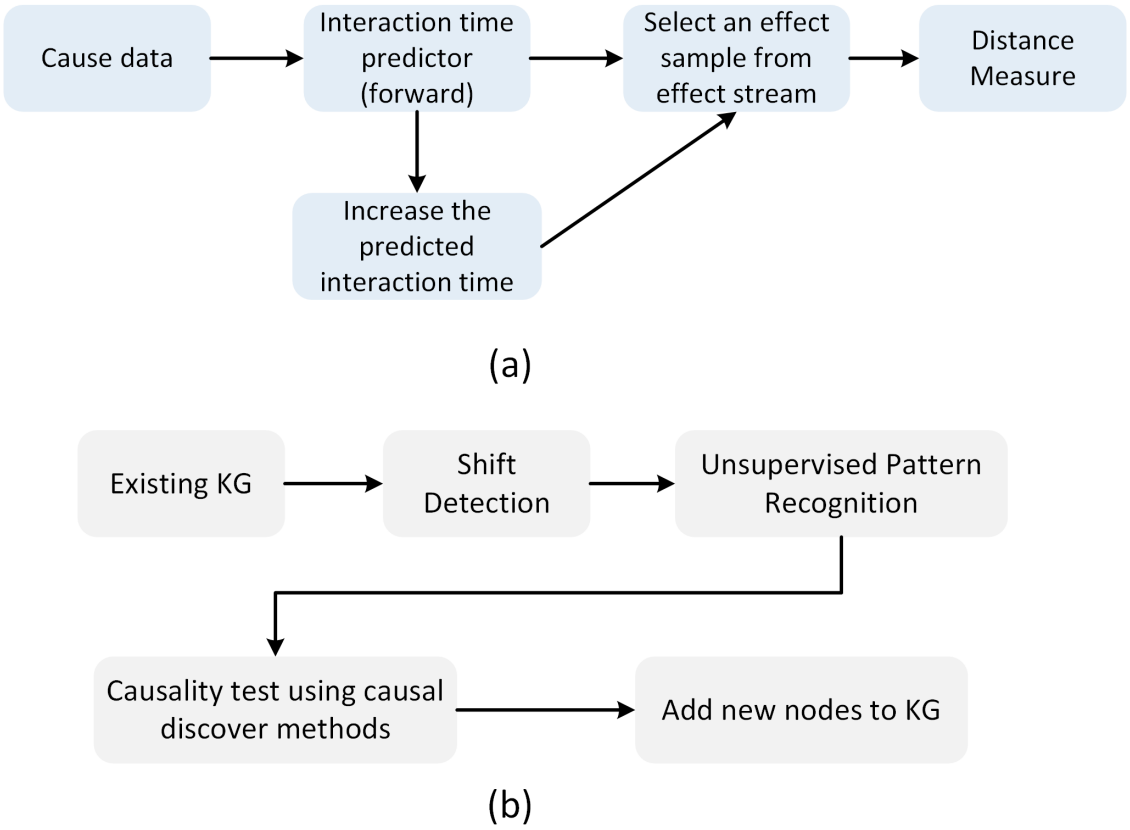

Figure 3.3: (a) a workflow to identify data distribution shifts by using interaction time as a sampling window. (b) a workflow to expand knowledge graph after data shift identification.

With the domain shift identified, another workflow shown in Fig. 3.3 (b) is introduced with the help of interactive causality to identify the unknown factor causing the shift and expand knowledge graph. After confirmation of data shifts, unsupervised pattern recognition algorithms, such as changepoint detection or dynamic time warping [1], can be applied on all the available data streams in the application environment. If some repetitive patterns are found, causal discovery algorithms, such as PC algorithm [78], can be utilized to test the causality between found patterns and perturbed effects. Finally, new nodes representing the found additional causes are added to the KG.

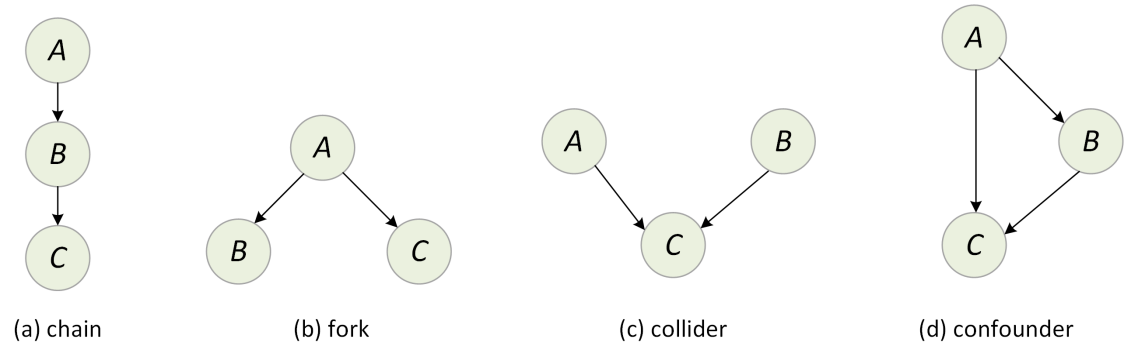

Figure 3.4: Four basic types of causal structures defined in graphical causal model.

3.3 Self-labeling in Different Causal Structures

In Chapter 2, a basic self-labeling method on a simple single cause and single effect causal structure is illustrated as a foundation. According to the statistical and graphical causal theory (*i.e.*, structural causal model), there are four basic causal structures, namely chain, collider, fork, and confounder. Assuming there are three variables A, B, and C, Fig. 3.4 illustrates the four causal structures. We will explore how self-labeling can be applied in these basic causal structures and extend to more complex causal graphs.

Chains can be represented as a sequence of nodes with arrows indicating the direction of causality from causes to effects. According to Fig. 3.4 (a), B acts as a mediator by facilitating the influence of A on C, which will not occur directly. Forks occur when a single cause produces multiple effects as shown in Fig. 3.4 (b) where A is the common cause for effect B and C. Colliders are situations where multiple causes (A and B) converge to produce a single outcome (C), as illustrated by Fig. 3.4 (c). A confounder is a variable that is associated with both the independent variable and the dependent variable in a causal model. In other words, it is a third variable that affects both the cause and the effect, making it difficult to establish a direct causal relationship between the two. Confounders complicate the causal analysis and causal effect inference. These four structures cover most of the commonly seen cases of causal relationships, thus will be discussed in this section in terms of their applications in

self labeling.

When there are multiple variables in causality, an emerging question is how to coordinate the relationships of causal time delays (*i.e.*, interaction time) of each pair of variables for self-labeling. Additionally, the undetermined logic relations among variables (*e.g.*, AND/OR/XOR) will further complicate the relational analysis for self-labeling. The causal relationships referred to in this section are the function space that maps cause variables to effect variables, *e.g.*, different logic relations of A and B in a collider to generate effects. The analysis of interaction time combination in the following does not assume specific causal relations.

Given a chain structure, the combined interaction time from A to C can be represented as

$$t_{AC} = t_{AB} + t_{BC} \tag{3.1}$$

due to the fact that the causal effect is sequentially transmitted. Thus, the interaction time of two pairs of variables in a chain can be directly combined for the self-labeling between A and C. The causal logic condition among these three variables will not affect the self-labeling as the causal effect is passed via each node.

With multiple effects in a fork structure, the effects can either individually or jointly label the cause depending on the causal logic relations and availability of effect observers. For example, if the logic relation is OR, either effect can be utilized as the observer for cause variable. Given multiple effects in Fig. 3.4, the combination of individual interaction time is represented as

$$t_{AC} = max(t_{AB}, t_{BC}). \tag{3.2}$$

The combination is taken by using maximum so that the latest effect can be captured and

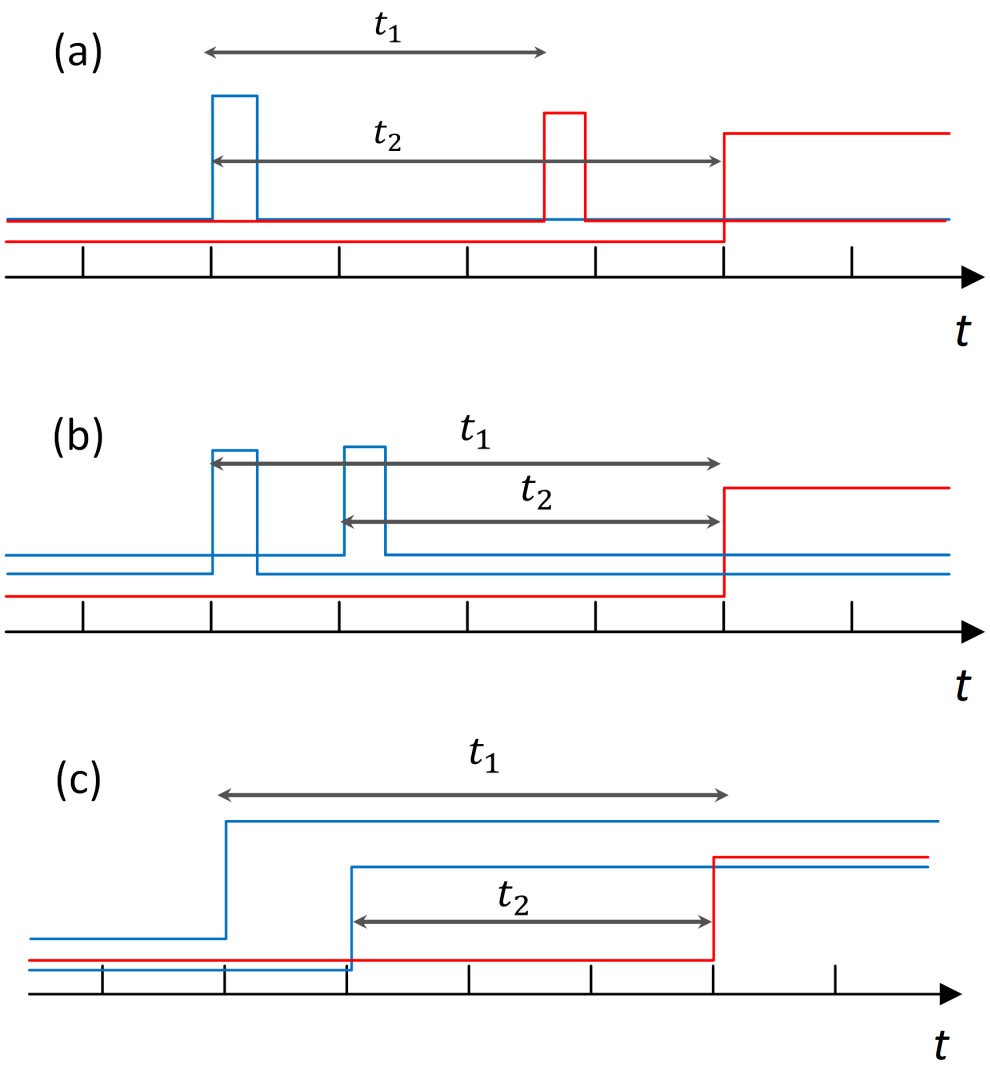

Figure 3.5: An illustration of different interaction time combinations in (a) a fork structure, (b) a collider with transient states, and (c) a collider with steady states. Blue lines represent the state variation of cause variables and red lines represent the state variation of effect variables.

utilized for self-labeling. Fig. 3.5 (a) depicts an example of a fork with a steady and a transient effect states where the combined interaction time should be $max(t_1, t_2)$.

In a collider, multiple cause variables jointly affect an effect variable. Regardless of the causal relationships, the state change of causes can be represented in steady or transient states as shown in Fig. 3.5 (b) and (c), and the interaction time combination is different. In the steady state case, the cause variables change states and remain steady until effect state

change. The combination of interaction time in the steady state case can be represented as

$$t_{AC} = min(t_{AB}, t_{BC}). \tag{3.3}$$

Taking the minimum guarantees that the selected cause states will always include the informative data segments. In the transient state case, the cause variables change to active states for a short time and change back or to other inactive states before effects showing up. The self-labeling method is required to capture the state and state transitions of each causal variable. Hence the transient of each cause variable needs to be captured in this case. An individual ITM for each cause-effect pair is needed to infer the individual interaction time used for self-labeling individual cause. In terms of different logic relations among the two causes, each cause variable can be regarded as a dimension of data used in the learning. For example, if the logic relation between A and B is an OR function, only the self-labeled segment of A may contain meaningful information while that of B may not. However, the data segments of A and B can be fed into task models as two dimensions of data sources for learning. Thus task models can still distill discriminative features to learn an OR mapping between the combination of data sources A and B and joint effect C. Similar learning strategies can be adopted for other logic relations.

In the confounding structure, the confounder A affects B and C. If the self-labeled variable pair is B and C, the confounder A will work as an additional cause. For the self-labeled pair of A and C, variable B works as an intermediate cause that will impact the end effect. In this case from the perspective of C to look back, A and B are regarded as two dependent causes. Compared with the collider case, the viewpoint from end effect to its causes does not change, thus the self-labeling scheme will remain same as the collider case.

In a more complex causal graph with multiple variables, the self-labeling scheme for the four basic cases can be used as a tool to analyze the interaction time calculus by disentangling a

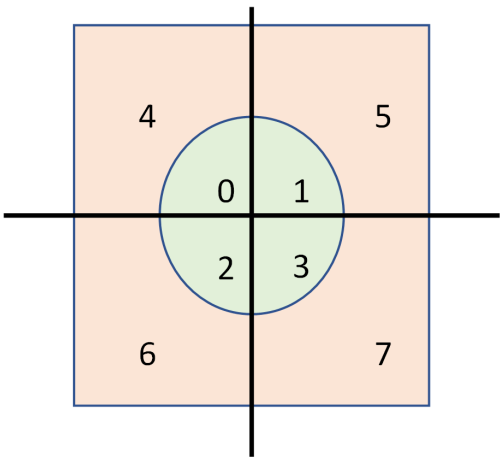

Figure 3.6: An illustration about the categorization of final effect distance vector in the simulation.

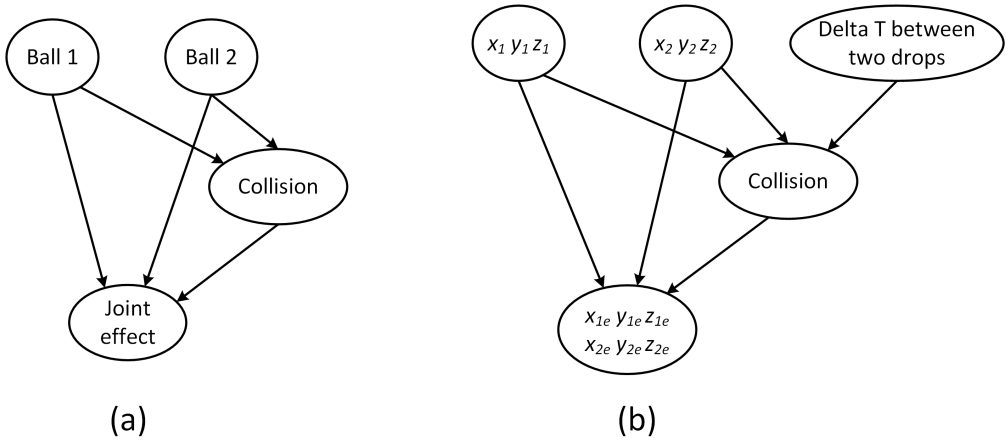

Figure 3.7: (a) A multi-variable causal graph used to represent the causality in the simulated experiment. (b) A detailed causal graph with each node represented by numerical variables.

complex graph into the four basic structures. This will be explored in the future.

3.4 Simulated Experimental Results

In this section, a simulated experiment is provided to demonstrate the interactive causality methodology for adaptive machine learning in a complex causal structure and for knowledge graph expansion.

3.4.1 Adaptive learning

To demonstrate the effectiveness of self-labeling in a complex causal graph, a simulation with multiple causes is designed and evaluated. TDW with PhysX engine is used as the simulator [42]. In this simulation, two balls are dropped at random positions and at different time onto a flat surface of size 150×150. The two balls will fall onto the surface, move, and finally settle down or reach the preset maximum simulation duration. Due to the relative positions and the different dropping time of the two balls, collisions may happen between the two balls to change their trajectories. We intentionally set the initial positions (in an area of size 20×20) of the two balls to let them collide at a 50% chance. The final effect is transformed into a joint effect representing a vector pointing from ball 1's final position to ball 2's final position after they settle down on the land. To make the ML task still a classification problem, this joint effect is categorized into 8 classes according to the direction and amplitude of the distance vector. Fig. 3.6 shows a plane of the distance vector which is partitioned into eight regions depending on angle and magnitude. The perturbation to simulate concept drift is a wind applied randomly to change balls' trajectories. To penalize inaccurate interaction time inference longer than ground truth, we set the two balls to first move horizontally with a velocity of 0.0025 before dropping. The penalty will let the balls deviate from their initial dropping positions to account for the inaccuracy in interaction time inference. The horizontal moving velocity is selected as a parameter for providing a reasonable penalty. Its default value of 0.0025 will change the colliding balls' initial positions by around 15% and prohibit them from collision.

Given the simulation settings, a causal graph can be derived as a lumped representation of causality between the two cause variables and one effect variable as shown in Fig. 3.7(a). In this causal graph, we can find out two typical causal structures. The variable ball 1, ball 2, and final effect form a collider structure. With the collision variable, there is also a confounder structure embedded in the graph. In Fig. 3.7(b), a detailed causal graph with

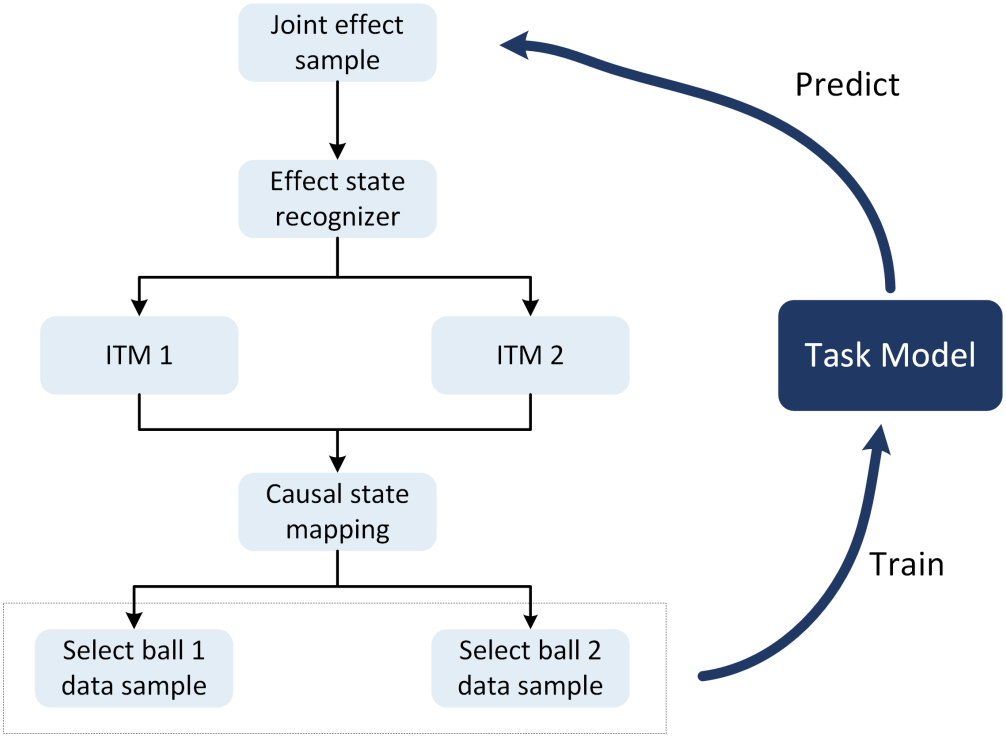

Figure 3.8: A self-labeling workflow for the multi-variable simulation.

specific variables are provided for this simulation. The machine learning task model is to use the two balls' initial properties to infer a class of the final joint effect. Thus self-labeling for the two cause events needs to be accomplished by using the joint effect. As this causal graph has two causes and each cause state is transient, two independent interaction time models are needed for each causal pair. An interesting observation is that the collision variable will not be involved in the self-labeling as its root causes are reachable in the graph. The holistic self-labeling workflow for this simulation is described in Fig. 3.8. Two ITMs individually ingest effect data and infer the interaction time for each cause to select and self-label a cause state from each cause data stream. Then the selected cause states are combined as a self-labeled sample used for retraining the task model.

Nested k-fold validation is stilled applied to evaluate the performance. 360 samples are used as the increments in self-labeled dataset and there are 25 increments. Test set has 1500 samples and validation set has 600 samples. Pretraining set has 600 samples. The task

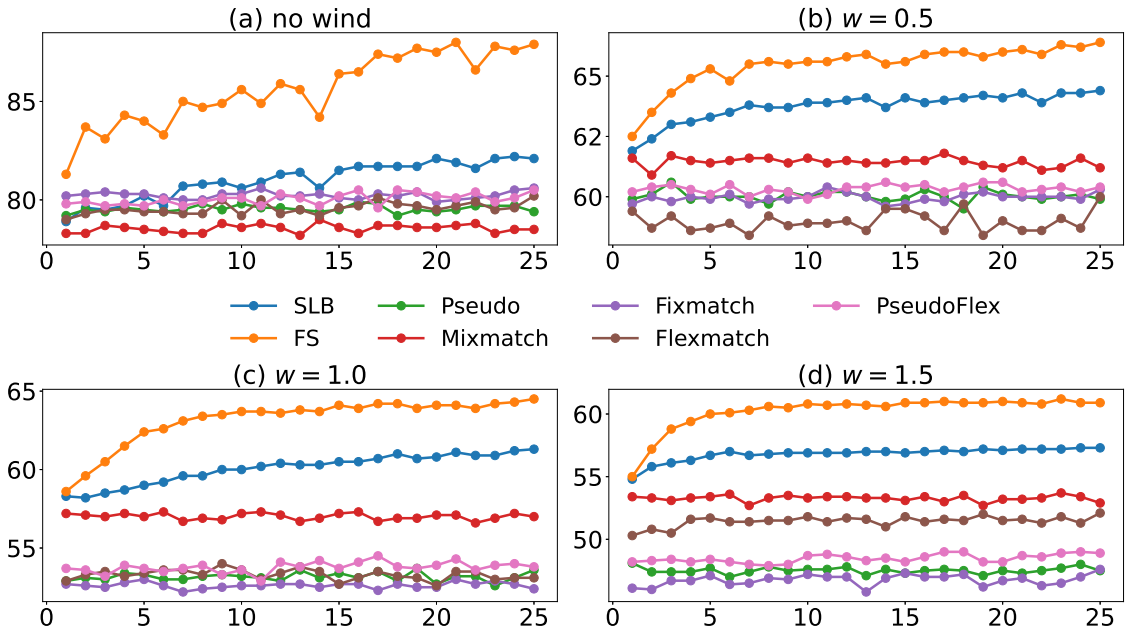

Figure 3.9: Model learning results with different wind magnitudes or without wind. Y axis is the accuracy in percentage. X axis is the number of increments of the self-labeled datasets.

model is a multi-layer perceptron (MLP) of size (32, 64, 128, 256, 512, 128, 64, 32) with a ReLU and a batchnorm layer after each linear layer implemented in PyTorch, optimized by AdamW using 0.0005 weight decay and 0.001 learning rate. The batch size is 64 with 600 epochs. Two XGBoost models are used as the ITMs for each cause (ITM 1 for ball 1 and ITM 2 for ball 2). Using the default simulation parameters and evaluated with the perturbed dataset, the R2 score for ITM 1 is 0.817 and its MAE is 31.2, and the R2 score for ITM 2 is 0.884 and its MAE is 24.7. The learning results are averaged over three random seeds for dataset splits.

Fig. 3.9 shows the results compared to five other semi-supervised methods. It can be observed that in the unperturbed case, the performance of self-labeling gradually exceeds other methods with more self-labeled samples. In the perturbed cases with different wind magnitudes, the self-labeling always outperforms other traditional SSL methods, which further demonstrates its superiority in data shift adaptation given a complex causal structure. To further validate the effectiveness, we change the penalty parameter and rerun experiments

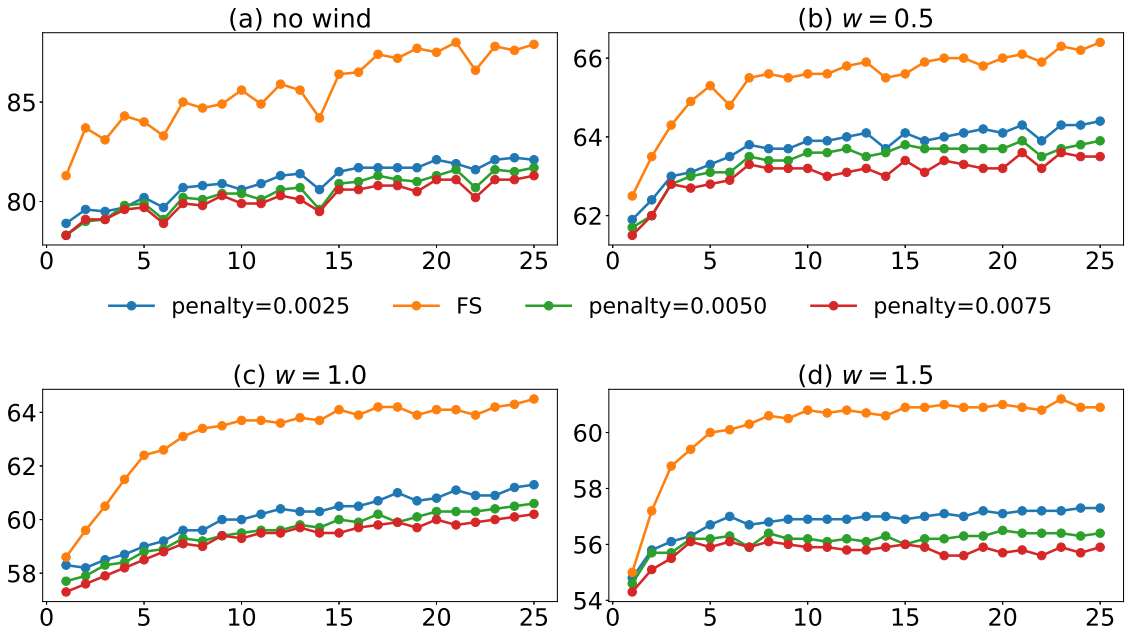

Figure 3.10: Model learning results with different penalty parameters (horizontal drift velocity) in different wind cases. Y axis is the accuracy in percentage. X axis is the number of increments of the self-labeled datasets.

with the results shown in Fig. 3.10. It can be observed that with much higher penalty parameter, the performance of self-labeling degrades but still outperforms other traditional semi-supervised approaches referring to Fig. 3.9.

In the theoretical derivation, the effect state recognizer is assumed to maintain 100% accuracy which is relatively impractical to achieve in real-world applications. The impact of inaccurate effect state recognizer will be quantitatively tested using the multi-cause simulation data. Fig. 3.11 shows the experiment result where label noise from the effect state recognizer is injected and controlled in the perturbed case with 0.5 wind magnitude. Three levels of label noise, 2.5%, 5%, and 7.5%, are tested. It can be observed that while the accuracy of the four cases fluctuates, the overall trend meets expectation that high label noise will degrade the self-labeling performance. However, the performance degradation is not intense with an average drop of 0.23% at 7.5% noise level, which demonstrates the robustness of the self-labeling against inaccurate effect state recognizer. This experiment confirms the applicability of self-labeling in real-world applications with error margins for the effect state

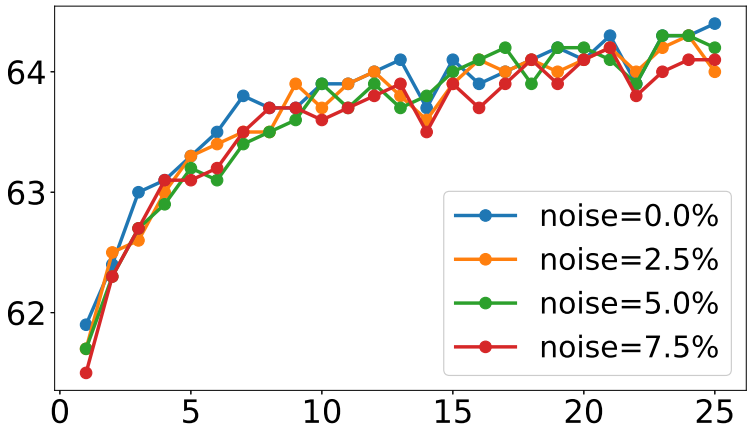

Figure 3.11: Experiment results with different label noise levels in the perturbed case with 0.5 wind magnitude. Y axis is the accuracy in percentage. X axis is the number of increments of the self-labeled datasets. Noise level is the ratio between noisy labels and total labels.

recognizer.

3.4.2 Knowledge Graph Expansion

The demonstration of KG expansion utilizes the multi-cause simulation data. The simulation was run ten times to generate 10 datasets in the original case without perturbation. A regularized approximation of Wasserstein distance is applied to quantify the drift of joint distribution of input and output as the distance measure [47, 39]. We measure the distances among each pair of the 10 generated datasets. A mean (μ_d) and a standard deviation (σ_d) of the distribution distances are calculated. The threshold is defined by using the mean and standard deviation in the form of $threshold = \mu_d + a\sigma_d$, where a determines the number of σ_d. In this experiment, we find out $\mu_d = 1.63$ and $\sigma_d = 0.19$. We define that $a = 2.5$ and thus the threshold is derived as 2.105. The coefficient a can be selected depending on the actual applications following the three sigma rule.

Fig. 3.12 shows the distance between the original and perturbed distribution with different extended interaction time. The interaction time is inferred from a cause sample and then extended with a step of 30 frames. It can be observed that the distance gradually increases

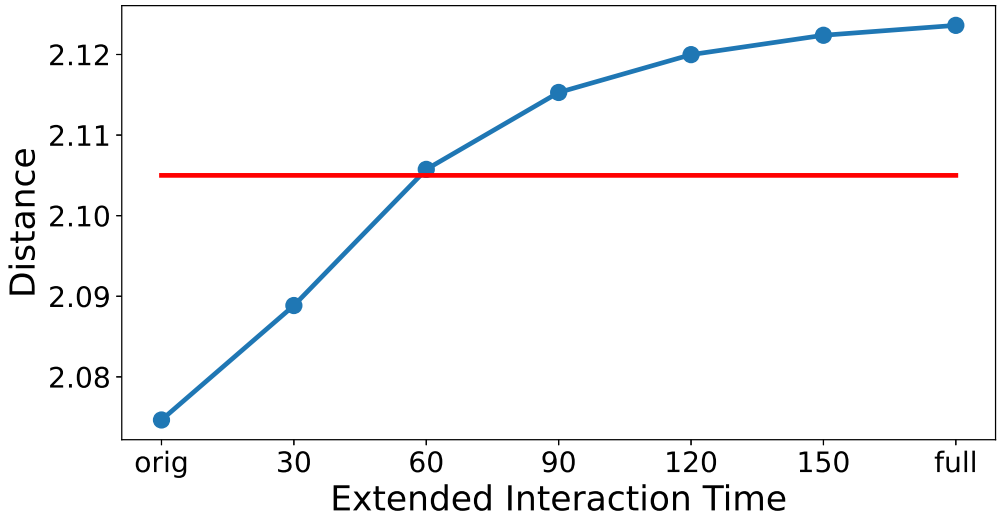

Figure 3.12: Distribution distances between original and perturbed distances with different extended interaction time. The red line is the defined threshold.

and exceeds the threshold, after which the data shift can be confirmed. The reason why the distance increases is that the perturbed effects in this simulation will show up with a longer interaction time than usual. In other words, the affected balls will gain velocity from wind and keep moving for a longer time. Therefore, the extension of interaction time allows the magnitude of perturbed effects to manifest, causing a larger distribution distance. This experiment demonstrates the feasibility of using IC to simulate how humans capture such anomalies via the variation of causal time lags.

After data shift is confirmed by using the interaction time as a sampling window, we follow the procedures in Fig. 3.3(b) to expand the knowledge graph by extracting consistent patterns and validate causal links. Unsupervised pattern recognition needs to be applied to all the available data streams. As a proof of concept, in this experiment the available data streams are only the wind magnitude and the positions of two balls. Thus a peak detector is applied to find the consistent spikes in wind signal. Note that in real applications, there can be many available data streams with complicated features and thus more advanced algorithms are needed to identify consistent patterns. After the extraction of wind patterns, the Peter-Clark (PC) algorithm is utilized as the causal discovery method to test the causality between the

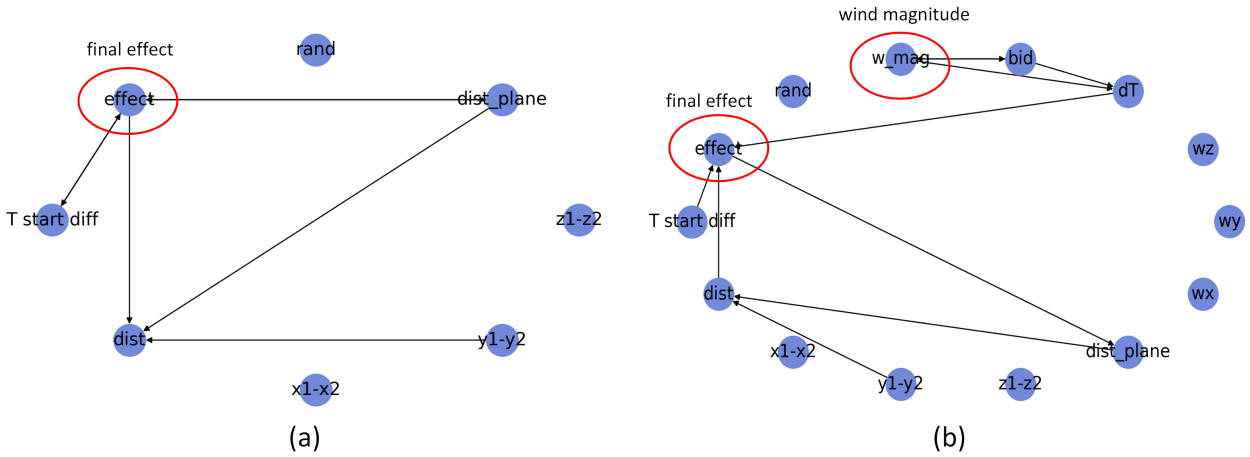

Figure 3.13: (a) shows the causal graph inferred by PC algorithm without wind. (b) is the causal graph generated by PC algorithm with wind, where the node representing wind magnitude is connected to the existing graph and the final effect.

found pattern and the perturbed joint effects. Fig. 3.13 shows the resultant causal graph with and without wind by PC algorithm. It can be observed that the PC algorithm successfully connect the nodes of winds to the existing network and the final effect, which demonstrates the feasibility of of knowledge graph expansion by using the interactive causality.

3.5 Discussion

In this chapter, we present the entire interactive causality (IC) methodology for knowledge modeling to extract causality and for adaptive learning in complex causal structures. IC method includes multiple stages for knowledge modeling and applications. It can be applied for adaptive machine learning to enhance model adaptation and assist new knowledge discovery. Additional simulated experiments to quantify the impact of inaccurate effect state recognizer are conducted. It is confirmed that the self-labeling has enough tolerance and is robust to inaccuracy of the two auxiliary model. Theoretically, quantification of inaccurate ITM and effect state recognizer can be accomplished by adding an error factor into the equations but is challenging to reach conclusion solely based on analytical equations. It is

envisioned that more strict evaluation of ITM and effect state recognizer can be completed in the future.

Chapter 4

Contextual Intelligent System in Advanced Semiconductor Manufacturing

4.1 Introduction

In smart manufacturing (SM), cyber physical systems (CPS) call for the enhancement of context-awareness of manufacturing machines and factory operations by contextualizing the sensed signals, detected events, and recognized surroundings so that it is capable of providing actionable intelligence to improve operational integrity, energy productivity, and machine prognostics and health management (PHM) [92, 136]. To transform the data into actionable intelligence, two popular frameworks have been proposed and conceptually implemented. One is to leverage the data generated in current manufacturing systems and transfer them to computing services for contextual machine learning (ML) [144, 138]. Another is to design a context-aware system with manufacturing engineers by incorporating additional IoT sensors

into available information from manufacturing execution systems for ML [5, 81]. However, both frameworks have not leveraged the contextual sensing capability of workers in real time on factory floors to complement the data generated by IoT sensors and existing manufacturing systems. Thus, an alternative contextual system design capable of incorporating the intelligence of shop floor workers, *i.e.*, human senses and knowledge/experience, in real time is worthwhile exploring.

While systems allowing workers on the floor to input information via computers have been developed, they have not been successfully integrated into existing manufacturing systems due to natural language inputs not compatible with data from machines. Additionally, workers are required to proactively input readable information, but it is not well accepted by workers due to sociological reasons according to a questionnaire [70]. These barriers motivate us to search an alternative way of connecting workers. In fact, workers are naturally connected to manufacturing systems through their active and reactive interactions with machines [167]. Workers' active and reactive interactions contain meaningful context values in the form of causes and effects. For example, workers as causes by following standard operation procedures (SOP) actively operate machines and machines change states as effects. On the other hand, machines behaving abnormally as causes result in workers reactively responding to machine operation conditions. By understanding worker interactions in the active and reactive aspects, the contextual information regarding regular operation and anomalies can be extracted.

To understand the worker machine interactions (WMI), a methodology to reliably to capture and confirm workers' intended inputs via gesture recognition is proposed in this work for capturing the time and location of happenings on the floor, which can commensurate real time data from existing manufacturing systems. The interaction data captured on the floor can then be used in ML for developing classifications of manufacturing conditions such as normal operation, operator errors, and machine warnings. This set of manufacturing

classifications can further support an existing manufacturing system in dynamically adjusting its execution commands for the machine fault prevention, workflow optimization, and energy productivity improvement.

The development of the WMI recognition system is based on the concept of well-established causality between workers and machines rather than supervised ML routines of manual data labeling and model training. A causally correlated Finite State Machine (FSM) model is established in this study to model the timing and causal behaviors of machines and workers during manufacturing processes. The design of FSM can leverage not only existing knowledge and experience from workers but also documented standard operating procedures (SOP) and machine operation manuals to extract the known causes and effects. The WMI recognition developed from FSM at normal operation conditions offers a class of various human gestures representing the contextual information of active healthy interactions. The anomaly detection of floor operation can then be determined when the worker reactive gestures fall out of the norms or the machine operates out of its functional states.

Conventionally, the reliable understanding of WMI requires advanced ML models with well-labeled dataset for training [148]. With the causality between workers and machines identified, the confirmation from the machine side as causes or effects provides an adaptive way of capturing WMI contexts as training data. Thus, initially a reliable method of observing machine states to automatically capture the data of causally related worker interactions becomes more applicable as the first step. The captured WMI contexts can then be used for the ML training of WMI recognition. For situation awareness of machine operation in the cause-and-effect method, machine states with their corresponding observable quantified contents are needed. For example, power signatures of individual components at states of active, idle, and off in each machine can be measured and presented as a cause of deviation from norms for workers to react.

To illustrate the concept of the causality-inspired contextual WMI recognition system, as

the first step, this paper builds a contextual sensor system with a security video camera for capturing WMI contexts in identifying causes of interaction and a power meter for observing the effects of interaction for establishing norms of worker gestures. Conversely, it can also use the same power meter via an energy desegregation technique in identifying the power levels and states of machine individual components as causes to observe effects of worker responses via video cameras. This study serves as the first step towards the causality-inspired contextual WMI recognition system. This paper presents a case study of semiconductor fabrication processes to demonstrate the capability to capture the sequence of contextual machine events with WMI contexts. In addition to the use of FSM modeling for the timing and causal behaviors of workers and machines, a novel energy disaggregation technique by exploring the logic states of machine components and their corresponding working principles is researched for analyzing power signals with fast-varying pulses caused by bang-bang control at a low sampling frequency, resulting in identification of power signature of individual components at various state. Finally, the contextual awareness of anomaly detection of workers and machines is illustrated.

4.2 Related Work

4.2.1 Context-aware Manufacturing Systems and CPS

Context-aware manufacturing systems have become a vibrant research area recently. Several studies focus on system-level designs by using IoT-based multi-sensor fusion and ML to achieve context-awareness [12, 3, 141, 137]. For example, Alexopoulos et al. utilizing massive sensor data designed a context-aware information distribution system that has visibility of shop floor processes and provides relevant recommendation information to relevant people [4]. In addition, the existing knowledge from humans can be provided to assist the design of a

Ref.	Year	Monitoring	PLC/Intrusive	Submetering	Technology Description	Context-aware
[55]	2020	Yes	Yes	N/A	MTConnect+Petri Net	No
[140]	2020	Yes	Yes	N/A	Combined with Digital Twin	No
[30]	2020	Yes	No	N/A	Computer vision based panel recognition	No
[105]	2016	Yes	Yes	No	Energy disaggregation with PLC control variables	No
[25]	2018	Yes	No	No	Frequency spectrum signal analysis	No
[126]	2019	Yes	No	No	Kalman Filter	No
[125]	2020	Yes	No	Yes	Supervised machine learning	No
Ours	-	Yes	No	No	Knowledge enhanced unsupervised way	Yes

Table 4.1: A comparison with some previous work

context-aware system. Horváth conceptualized a context-driven and knowledge-driven CPS modeling and system design methodology [61]. Emmanouilidis et al. proposed a conceptual context-based framework for maintenance management that integrates expert knowledge to a classification model where humans can identify unknown data or conditions and subsequently include the unseen information into a knowledge pool for future uses [38]. Wang et al. leveraged the known contextual information about a CNC machine to classify the collected data from CNC and mounted sensors into different machine states [145]. Inspired by these previous research work, this paper further leverages the documented knowledge from interaction-based SOP and the instrumentation working principles of machines in the software design phase to expedite the contextual system development in CPS.

4.2.2 Machines and Their Components Monitoring

Technologies for monitoring multiple machines status have been reported using RFID [110, 63], Wireless Sensor Networks [90, 160], or interfacing with PLC [36]. On the other hand, the component characteristics of an individual machine in real time is information of interest for gaining its operation visibility, since in general a manufacturing machine has multiple components (*e.g.*, pump, heater, spindle). Drake et al. proposed a framework to characterize the energy consumption of machines and their components in real time by utilizing one power meter to monitor the total power of an individual machine and by analyzing its components' power based on the prior dataset collected from operating the components in a sequential

order [33]. Panten et al. correlated the machine condition data from PLC with aggregated power data to identify the energy consumption of machine components in an online manner [105]. Tan et al. correlated the production data with power consumption to monitor machine status in real time [135]. Cheng proposed an alternative by monitoring machine operation states through current analysis eliminating a need to interface with PLC [25]. Han et al. discussed using non-intrusive high-frequency audio and vibration signals to classify faults of a cutting machine [57]. In this paper, a knowledge transfer CPS is proposed to address both machine and components monitoring. The hardware uses a combined camera and power meter for the real time visual and energy information respectively. The software facilitates the correlation between the finite states defined by interaction-based SOP and the real time visual and energy information. As listed in Table 4.1 comparison with several previous studies of using PLC or energy states [55, 140, 30, 105, 25, 126, 125], this novel approach can be easily implemented without requiring interfacing with customized PLC, massive sensors, and labor-intensive dataset collections for model training. Furthermore, with the correlated SOP model and visual information from cameras, WMI contexts can be extracted effortlessly.

4.2.3 Energy Disaggregation In Machines and Their Components

With a great number of non-intrusive load monitoring (NILM) solutions for energy disaggregation being developed and evaluated on residential applications in recent years [62], researchers have begun to explore its potential in industrial sectors [59, 97]. There are typically three types of loads: single state (on/off), multi-state, and continuously varying [164]. Energy event detectors serve as major modules for the first two types to extract steady-state features, and the third type demands high sampling rates at kHz for capturing transient and high-order harmonics features [146, 84]. Several window-based event detectors are proposed by studying statistical features, e.g. Chi-squared test [69], generalized likelihood ratio detector [6], Teager–Kaiser energy operator [149], variance and absolute deviation[114]. Many

of the existing energy event detectors are evaluated on kHz signals or less oscillating signals for residential appliances, whereas in this study we develop a detector on low sampling rate signals superposed with fast-varying pulses for manufacturing equipment. Furthermore, we explore the use of human knowledge in instrumentation designs for machine component control such as temperature, spinning, heating, flow, etc., and their corresponding electrical signatures for energy disaggregation. This is done by correlating a main power reading from FSM-based SOP with electrical signatures of components to identify the power consumption of individual components. It is of interest to note that the main power reading is a result of context awareness of repetitive measured signals from a main power meter. While beyond the current scope of this paper, it is worth mentioning that the same methodology can be easily extended towards energy disaggregation of multiple machines for an entire manufacturing floor with a single power meter.

4.2.4 Operator 4.0

In the context of Industry 4.0, several frameworks of operator 4.0 have been proposed to empower workers' capability, monitor workers' behaviors, and identify operators' new roles. For example, Segura et al. introduced visual computing technologies to assist worker operations [166]. Zolotova et al. discussed how operators and cyber-physical production systems interact with new trending technologies [165]. In addition, Kaasinen et al. analyzed user expectations and worker concerns regarding the adoption of operator 4.0 technologies [71]. Cimini et al. conceptualized a human-in-the-loop framework to discuss humans' critical roles in interactions and enhanced decision making with manufacturing systems as a socio-technical system [26]. In this work, the contextual sensor system will enable the real time training of machine operation for workers, the operational fault detection, and the prevention of occupation injuries, since the WMI are constantly under surveillance in a non-intrusive manner.

4.3 Contextual Sensor System Design

As depicted in Fig. 4.1, the proposed contextual sensor system is based on an FSM model built from the SOP including workers and machines, which are correlated through state transition functions. The system hardware consists of a visual camera and a power meter to collect real-time data, and a contextual software to process the sensed contents to generate contextual information. The system implementation incorporates a knowledge transfer framework that leverages human knowledge and documented knowledge to initialize the contextual software design with these two simple sensors. A case study of a semiconductor fabrication machine is successfully demonstrated using the proposed contextual sensor system hardware and software architecture.

4.3.1 A Contextual SOP Model and Knowledge Transfer Framework

For a single manufacturing machine or workstation, the standard operation procedures (SOP) provided by equipment vendors define a sequence of operations a worker needs to accomplish, which can be modeled as a sequence of interactive events $\{e_0, e_1, ..., e_n\}$. The interactive events define the actions or information a worker needs to take and the expected result a machine will provide, which forms cause-effect pairs. The contextual information underneath an event can be modeled as $e = \{x, t, P\}$, where x is the location, t is the timestamp, and P represents the event properties including both sides (workers and machines). The worker and machine status can be modeled as a Finite State Machine (FSM) respectively to represent the consistent state transition. The SOP provides such state transition information as shown in Fig. 4.1(a). The machine (or its component) states q can be the operation states such as off, standby, on, and material loaded etc., and the worker states v can be operating actions. It is worthwhile mentioning that the operation states of a machine include multiple functional

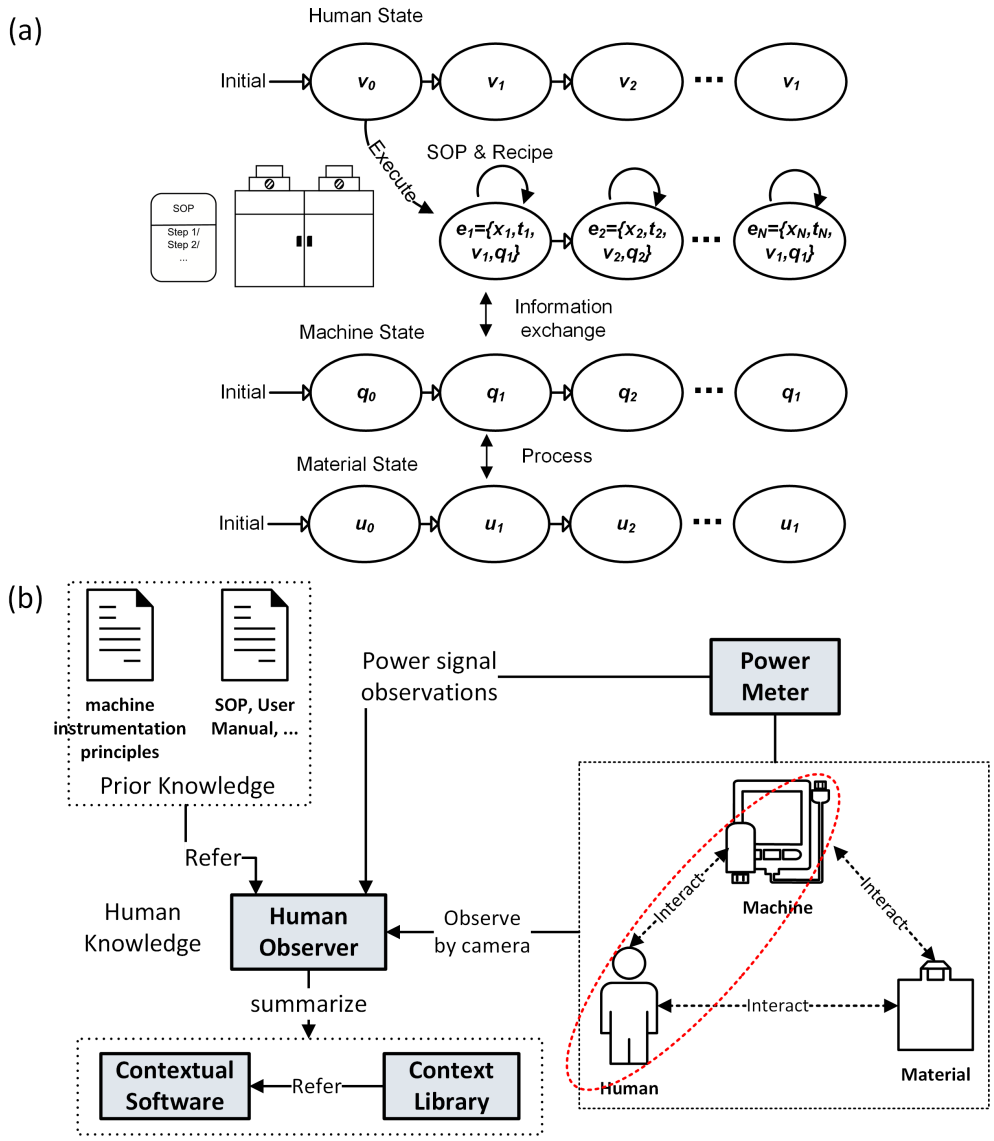

Figure 4.1: (a) An example of the FSM-based SOP model abstraction. The SOP defines an event-based operation sequence with worker state and machine state. Material state is changed by machine processing via a recipe developed by human. In (b), the proposed knowledge transfer framework in CPS. Note that this paper focuses on the worker machine interaction only.

instrumentation modules, *i.e.*, heating, pumping, spinning, etc., which are independently processed by various machine components and can operate in sequence or simultaneously. The machine states and worker states are correlated through the transition function δ defined

by SOP as

$$q_{i+1} = \delta(v_i, q_i), q \in Q, v \in V \tag{4.1}$$

where Q and V are the predefined machine state space and worker state space from SOP respectively. The SOP event context becomes $e = \{x, t, v, q\}$, in which the machine state change is a result of different worker states. For example, a manually controlled machine is turned on because a worker presses the switch a few moments ago. By using this correlated SOP model as the basis, machine events and worker events can be detected independently and correlated uniformly to uncover the WMI contexts. In this study, we focus on the machine energy state determination.

In addition to the related worker and machine state transitions, materials can also transit their states u after a worker controls a machine to process. The material state transitions can be additive or subtractive to a part (*e.g.*, wafer) to show shape changes, phase changes (*e.g.*, metal refining from solid to liquid), or chemical reactions with byproducts. The material state transition can also provide the contextual information similar to the WMI but is beyond the scope of this study.

The correlated SOP model serves as the basis of the knowledge transfer framework and the contextual sensor system. The correlated SOP model defines two entities to be measured, worker states and machine states. In order to capture signals from both sides, a visual camera (can be a security camera) and a power meter are selected as the hardware sensors for the contextual sensor systems. Visual Cameras are readily available sensors and contain meaningful contexts of workers and surroundings, which are selected to determine worker states and side channel information from the surrounding environments and machines. On the other side, the machine or component energy state change can be directly reflected on the energy consumption, which is measured by a main power meter in real time.

Based on the SOP model, a knowledge transfer framework is built to define a workflow to transfer the implicit engineering knowledge from workers and documented prior knowledge to the system design loop as illustrated in Fig. 4.1(b). Basically, a shop floor includes three major elements: people, machines, and materials, among which direct or indirect interactions occur to proceed the manufacturing processes in a way of state transitions. For example, machines interact with materials to process a recipe (*e.g.*, deposition, etching) for changing product states [147]. A Worker interacts with a machine through an interface to control process parameters, start running processes, and change the machine state. A human observer is introduced in this framework to serve as a knowledge accumulator by watching the always-happening interactions through cameras in accordance with the SOP, instrumentation principles, and the sensed power signals. In fact, the human observer can be senior process engineers and does not need to in-person watch the process since the engineer has already established their knowledge database during the long-term career. Initially, with prior knowledge, the human observer is to acknowledge the variation of power signals by analyzing the recent observable interaction sequence with corresponding power outputs to confirm the relevance and consistency among SOP, power signals, and realistic human-machine interactions. The observer can follow the SOP to recognize the worker state (from WMI) and thus understand the corresponding machine state and power signals. The corresponding segment of power signals can be attributed to a certain component or a group of them with respect to the SOP. Finally, the obtained and summarized knowledge from this observation can be leveraged and transferred to boost and append the context extraction capability to the software design process. After few iterations, a contextual sensor software can be developed to act as an artificial human observer to recognize component state transitions from aggregated power signals. Moreover, the knowledge of human observer can be abstracted and encoded into a context library where several known consequences of the interaction processes and events are stored, and which can be used as a look-up database to search for possible reasons when some typical sequences of events are detected. The proposed knowledge transfer framework based

on the correlated SOP avoids the submetering data collection to identify component power signals. It is also noted that the deviation of typical sequences of events could be used to identify anomalies of machine operation, which might be attributed to gradual performance degradation of functioning components or undetected intrusions in cyber-attacks.

In addition, with many component state transitions being detected, the WMI videos can be annotated in a label-free manner according to the FSM-defined state transition correlation to train a ML model to recognize the interactions, which will be addressed by another publication from the authors.

4.3.2 Contextual Sensor System Architecture

We applied the proposed knowledge transfer framework to develop and implement a contextual sensor system on a typical semiconductor fabrication equipment, PlasmaTherm, located in a cleanroom facility. PlasmaTherm is a PLC-controlled machine with dual chambers and functionalities: PECVD (plasma enhanced chemical vapor deposition) and RIE (reactive ion etching), by using the generated gas plasma. Several gases can be used to generate plasma for different purposes. The machine is equipped with multiple instrumentation functions: the creation of desired vacuum conditions for semiconductor processing, the generation of plasma from gases and RF sources, the control of semiconductor substrate temperature, the electronics for PLC and user interfaces. These instrumentation functions have corresponding components: mechanical vacuum pump (roughing pump), RF generator, heater with controller, and main body with PLC and PC etc., respectively. These components can be in various states at each process step for different functions. The two processing chambers are driven by the same set of components. RIE side does not require an elevated temperature setting, while PECVD requires a constant elevated temperature during deposition. The simplified and generalized SOP for the two functions is illustrated in Table 4.2 (STBY

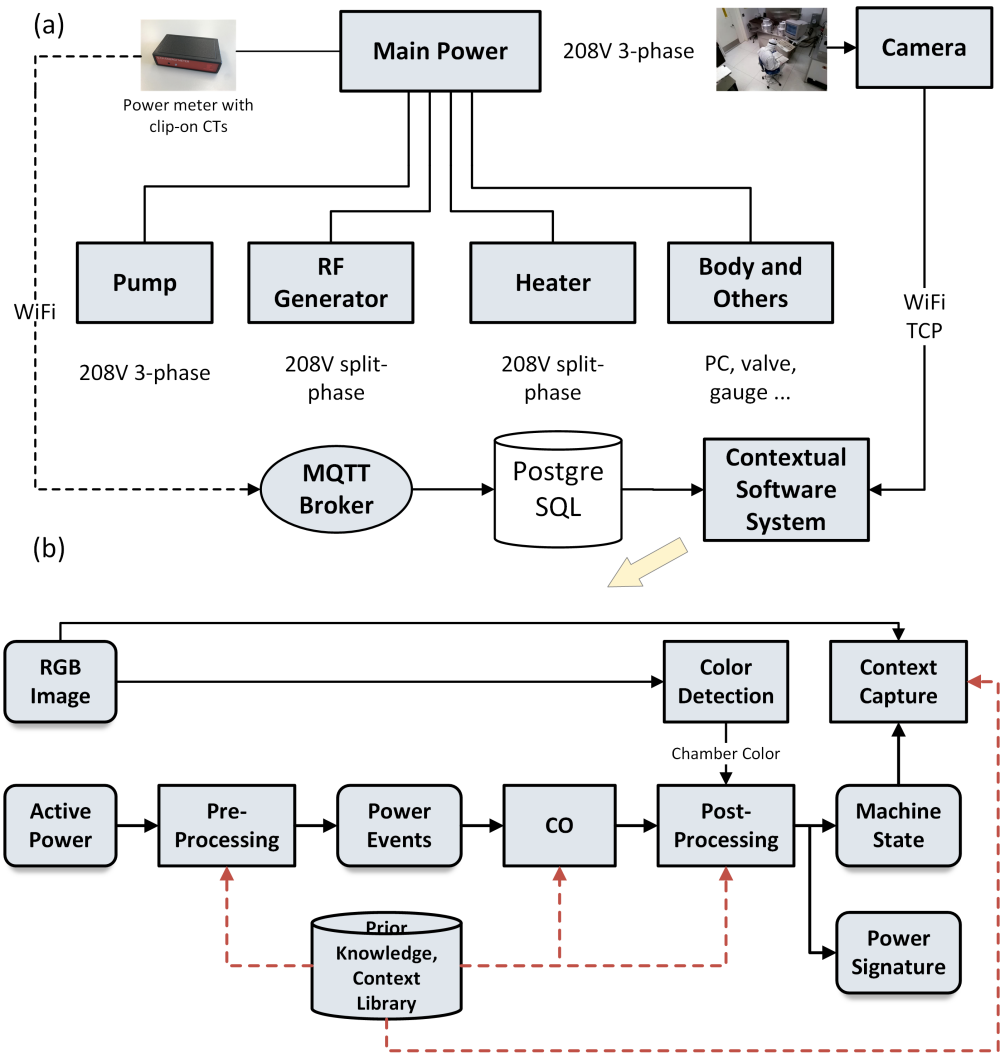

Figure 4.2: The hardware and software structure of the implemented contextual sensor system. (a) shows a semiconductor processing machine, the PlasmaTherm with 4 instrumentation modules (see the text) with their corresponding components connections with various power supplies. A visual camera is mounted from a near ceiling view to monitor the entire machine. (b) outlines the data processing pipe.

stands for standby states, and low-vac represents the chamber low-vacuum states). A worker is required to execute the SOP through the machine interface (a monitor with keyboard). For example, at step 5 a worker operates the keyboard to choose a product recipe and hits "RUN" button to start the process, which results in the RF state changed from standby to on when the inflow and removal rate of gases (by vacuum pump) reach a steady state.

Fig. 4.2(a) depicts the hardware and data acquisition and transmission settings. A visual

Step	Process	PECVD			RIE		
		Pump	RF	Heater	Pump	RF	Heater
1	Set temp.	on	STBY	on	-	-	-
2	Vent	on	STBY	on	on	STBY	off
3	Load	on	STBY	on	on	STBY	off
4	Pump down	low vac	STBY	on	low-vac	STBY	off
5	Run Process	on	on	on	on	on	off
6	Purge & Vent	on	STBY	on	on	STBY	off
7	Unload	on	STBY	on	on	STBY	off
8	Reset temp.	on	STBY	off	-	-	-
9	Pump down	low vac	STBY	off	low vac	STBY	off

Table 4.2: A generalized SOP of PlasmaTherm with dual functions

camera is mounted to capture the real-time image stream through WiFi connection and TCP protocol. The image size is set to be 640×480 with the frame rate of 10 fps. A clip-on power meter with current transformers (CTs) are installed on the circuit breaker to monitor the main power feed for the entire machine. The meter we used is easily installed by clipping on the CTs to the power lines with voltage sensing wires connected to the power lines. The meter is reconfigurable to monitor three-phase, split-phase, or single-phase load. Since other single-phase components, *e.g.*, PC, PLC, and valves, consume less power and maintain insignificant power change compared with main parts, they are omitted in this study. The power meter samples the active power signal at 1 Hz frequency and transmits JSON-format data through MQTT, the data is stored in a local PostgreSQL database. The developed contextual sensor software system queries the database every second to fetch the power data and accepts the real-time image stream to process.

The data processing pipeline is illustrated in Fig. 4.2(b). There are two streams for the image data and power data processing respectively. The power data processing stream analyzes the main power to extract different types of power events and disaggregate them to derive the individual component states with predicted individual signals. The details of this power signal processing will be addressed in Section 4.4. Since the visible light emission depends on the type of gases in use for plasma, a color detection module based on the chamber window

Name	Attr.	States					
		off				on	
Body	Power (W)	0				900	
	T_{res} (s)	-				1	
		off				on	
Heater	Power	0				1300	
	T_{res}	-				30	
		off		on		low vac	
Pump	Power	0		750		1100	
	Tres	-		-		2	
		STBY	30W	70W	125W	175W	300W
	Power	50	300	400	500	600	850
RF	$T_{res}(CF_4)$	-	90	90	90	90	90
	$T_{res}(O_2)$	-	65	65	65	65	65
	$T_{res}(SiH_4)$	-	215	215	215	215	215

Table 4.3: PlasmaTherm power states and corresponding response time

color intensity is developed to detect the plasma gas type. With the contents of power signatures and chamber color detected, a context capture module is developed to correlate the contents into contexts.

Since the SOP model defines the correlation between machine states and worker states as visible in WMI video snaps, one design aspect of the captured context is the response time T_{res} between WMI and machine state transitions. The response time is common for PLC-controlled manufacturing machines to conduct a self parameter inspection or adjustment before a process starts. Using PlasmaTherm as an example, when a user selects the processing recipe and hits the "RUN" button, the machine will first adjust the gas flow rate to reach steady states for a fixed period of time after which the RF is turned on and the manufacturing process begins. The corresponding response time is composed of a static segment and a transitional period depending on the gas flow. The response time for PlasmaTherm is listed in Table 4.3, where T_{res} is derived from its inactive state (heater: off, pump: on, RF: STBY) to operational (active) states. RF has three T_{res} distinctive on the gas type since the gas flow rate and the time to steady states are different. In reality, since the gas flow rate varies, T_{res} can be regarded as a normally distributed random variable depending

on multiple factors (*e.g.*, gas valve leakage, gas inventory, pressure). After several iterations in measurements, an averaged response time over measurements is selected as T_{res}. If the interaction starts at time 0, the machine state change will be recognized at time $T_I + T_{res}$, where T_I is the interaction duration. Therefore, the time period $(0, T_I)$ containing WMI contexts needs to be pinpointed.

The other design aspect of the captured contexts is to analyze the sequence of detected events with timestamps and compare them with the context library to determine possible consequences. For example, following the expert experience, a 30-min oxygen clean should be conducted to clean the inner chamber before any etching or deposition process begins. If a worker forgot to do it and failed to obtain the expected processed material, the contextual sensor system can provide a likely cause that the oxygen clean was not performed. Moreover, by comparing the duration or the magnitude of the low-vac state pump power signals, the system is able to estimate the efficiency of the pump or whether the pump or valves have unusual leakage. With the context library built upon expertise from humans and documented knowledge, the contextual information and actionable intelligence can be supported by the system. In this study, to illustrate the proposed framework, three predefined contexts are abstracted from facility staff's knowledge and SOP with reference to the event sequence: 1) A regular operation should follow a sequence of RF on (optional O_2 clean), pump low-vac, RF on (can be multiple times), pump low-vac, and RF on (optional O_2 clean), where over 60-minute continuous RF running is prohibited; 2) While it is rare that two consecutive pump low-vac states are detected, this sequence may indicate a pump malfunction during first low-vac state; 3) a small bump of the pump power signal during inactive pump on states can indicate an unusual gas leakage from the enclosed chamber, valves, or pipes.

4.4 Software Defined Sensor for Power Event Detection and Classification

In this section, the prior knowledge from SOP and the working principles of instrumentation engineering designs for functional modules are utilized to design the software defined sensor system, which is capable of detecting power events and reporting the individual components' energy consumption. As illustrated in the power signal processing in Fig. 4.2(b), it includes preprocessing, Combinatorial Optimization (CO), and post-processing. It is worthwhile mentioning that the knowledge of instrumentation principles and their corresponding components can highlight the anticipated power waveform during normal operation.

4.4.1 Working Principles of Functioning Instrumentation Modules and Their Components

Design of a Vacuum System. The rotary-vane vacuum pump, a type of mechanical pump, is typically used in semiconductor fabrication equipment as the roughing pump for creation of low vacuum. The pump is driven by a three-phase motor and its power consumption is related to the amount of gas in the enclosed chamber according to the working principle. When the machine chamber pressure is always low at idle states, *e.g.*, 10 mTorr, the power consumption of the motor is relatively constant and low. When the chamber is vented to atmosphere for sample loading and needs to be vacuumed again, the motor load increases abruptly, which will cause a power surge of the motor. With more gas being pumped out and lower chamber pressure, the motor load will gradually decrease, which reduces the power consumption to the constant level. From the prior knowledge about the working principle, we can derive an educated guess of the pump power signature during operation.

Design of RF Plasma Generator. RF plasma generators are pervasively applied in

semiconductor fabrication to generate reactive gas plasma for dry etch, PECVD, and inert gas for sputtering etc. In general, a RF plasma generator includes a RF power supply, a RF matching network and a reactor (torch) [80]. The generation of gas plasma depends on the gas type, gas flow, pressure, temperature, humidity, and RF power [98]. One of the key processing requirements of the generated plasma is to maintain a constant plasma power and density to stabilize the etching or deposition process. Therefore, the power supply of the plasma generator is designed to provide a stable power during the process and can be tuned to control the generated plasma property. PlasmTherm has a PLC to control the process with stable RF power using predetermined process recipes, allowing a user to select a recipe with specific plasma power and duration.

Constant Elevated Temperature Controller. In many industrial applications, a stable temperature control is important for product yield and thus tools are equipped with self-regulating heaters. With thermocouples to sense temperature for a feedback control, a heater is designed to be turned on and off when the temperature is low or high respectively for stabilizing a preset temperature. At the beginning of ramping up the temperature from 25 °C to a user selected temperature, such as 250 °C, the heater operates at a constant power mode until the temperature gets close to the set value. During elevated temperature stabilization, a feedback control mode kicks in to turn the heater on and off frequently. Compared with other instrumentation functions, the pulse-like waveform is unique for the heater and the harmonic features can be extracted to detect such a pulse signal with higher-order frequency components.

The knowledge from these three instrumentation function modules will be explored in the design of data processing in software defined sensors for identification of three machine components, *i.e.*, pump, RF generator, and heater.

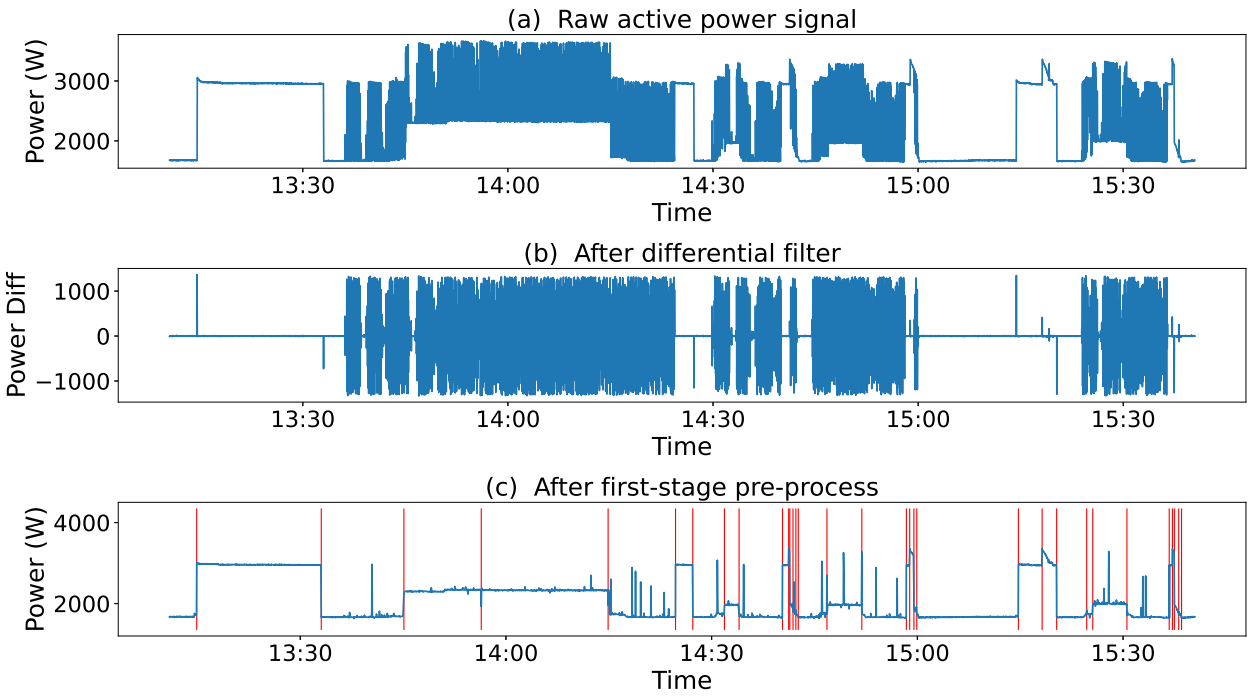

Figure 4.3: An example with the measured raw signal going through the first-stage preprocessing algorithm (based on the instrumentation functions) to show the performance. (a) an active power signal captured from the main power meter with heater, RF and pump at different states. (b) the signal after differential filter with signal variation being amplified. (c) the derived signal after first-stage pre-process to remove the pulses. The red lines in (c) indicates the detected power event from second-stage pre-process.

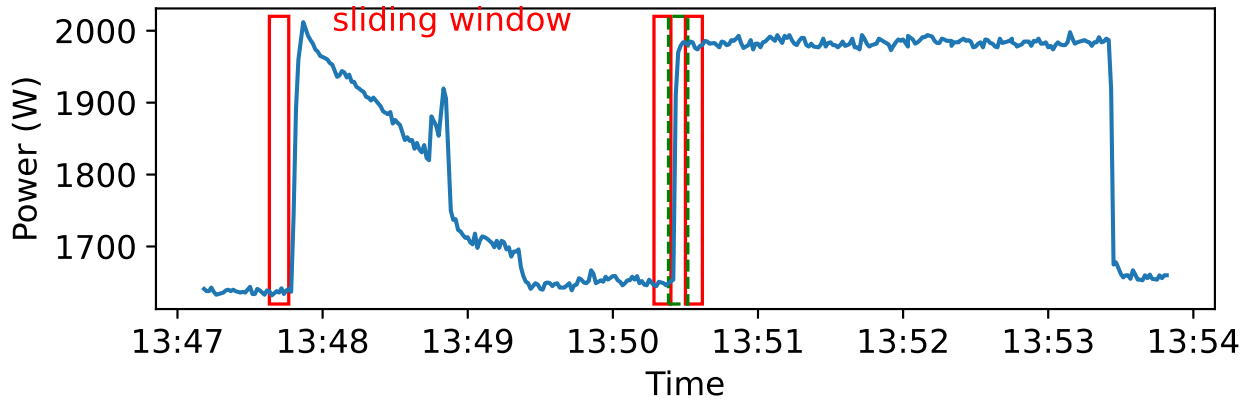

Figure 4.4: An illustration of a power signal with the SW-based second-stage preprocessing techniques to detect power events. In the middle, the two red boxes represent two windows right before and after the power ramp with the small variance, whereas the green dashed box represent the window capturing the edge with large signal variance. The two red windows also capture the steady state powers and the random noise or spikes can be avoided through comparison with steady power values.

4.4.2 Two-stage Data Preprocessing

In order to deal with the 3 components with similar or different power events, a hybrid two-stage algorithm is designed to pre-process the aggregated raw power signal and detect the power events of different types in real time. There are basically two types of power events, one for the pump, RF generator, and constant power heater with steady state features, the other for the heater in the pulsing mode. Fig. 4.3(a) shows a raw power signal captured from the main power meter during PECVD operation and consists of different combinations of the pump, RF, constant power heater, and pulsing heater at different states. There is a rather challenging case where the pump is in low-vac state and the heater turns on to the constant power mode and then transits to the pulsing mode or vice versa, increasing the difficulty to detect all types of power events. To overcome this challenging task, we segregate the detection of the two event types: pulsing states based on raw signals, and steady states based on filtered (removing the pulses) signals.

To detect the pulsing state from a raw signal, the raw signal is first partitioned into sliding windows (SW) with a width of 20 and a stride of 1. A differential filter is applied on the windowed signal to calculate the signal difference between adjacent timestamps, which can be represented as $P_d = P_{t+1} - P_t$. The frequency domain feature is calculated within the windowed P_d by Fast Fourier Transform (FFT). The 2nd and 4th order frequency components are extracted and a threshold (th_{fft}) is applied to determine whether the signal includes fast-varying pulses.

With the capability of detecting the pulsing mode, one can remove the pulsing component and extract the remaining waveform for the steady state power event analysis. The complete two-stage preprocessing method is illustrated in Algorithm Algorithm 1. When no pulses are detected by FFT, the raw signal value is kept in the filtered signal. When there is a pulse, a threshold (*bound* derived from the heater on-off power value in Table 4.3) is applied

Algorithm 1: A Hybrid Two-stage Preprocessing and Event Detection Method

Input: P_{raw}, and predefined thresholds
Output: event type, time, steady state power, P_f
$P_d = (P_{raw}[1:] - P_{raw}[:-1])$;
$freq_{mag} = FFT(P_d)$;
if $freq_{mag} > th_{fft}$ **then**
    Output pulsing heater event;
    if $abs(P_d[-1]) < bound$ **then**
        if $P_{raw}[-1] < th_{max}$ **then**
            P_f.append($P_{raw}[-1]$);
        else
            P_f.append($P_{raw}[-1] - HeaterOnPower$);
else
    P_f.append($P_{raw}[-1]$);
Second Stage;
Calculate signal Variance var on P_f;
if $var > th_{var}$ **then**
    $eventFlag = 1$;
else
    if $eventFlag == 1$ **then**
        s = mean($P_f[0:2]$);
        if $abs(s - s_{last} > th_{cb}$ **then**
            Output event type, time, steady state power;
    else
        s = mean(P_f);
        if $abs(s - s_{last}) > th_{cb}$ **then**
            Output event type, time, steady state power;
    $eventFlag = 0$;

on P_d to filter the fast-varying data points. The remaining signal is compared with the maximum possible value (th_{max} derived from Table 4.3) at non-pulse states to determine whether to keep the raw value or subtract the heater on-state power from the raw value. Then the filtered signal P_f is derived. Note that the P_f is also experienced re-sampling as the fast-varying pulses are removed instead of filtered. Fig. 4.3(b) and (c) show an example of signals after the first-stage preprocessing. We can observe in Fig. 4.3(c) most of pulses are removed and the steady-state waveform including RF, pump, and constant power heater

is preserved. There are still a few non-filtered spikes in P_f, which can be eliminated in the second stage signal processing with a noise-tolerant feature.

The second stage of the preprocessing method is designed to detect the edges and steady state values from P_f. We apply a sliding window with a width of 5 and a stride of 1 on P_f as illustrated in Fig. 4.4. The signal variance of each window is calculated and a variance threshold (th_{var}) is applied to filter event windows and non-event windows. This is based on an assumption that in industrial environment power events do not happen more frequently than the window width. Therefore, two non-event windows with steady-state power values should be right before and after a series of continuous event windows. When the difference between these two steady state values is greater than a threshold (th_{cb}) that can be derived from the minimum power difference during any possible state changes in Table 4.3, a power event can be determined, resulting in automatically eliminating the non-filtered spikes (noise) to enhance the robustness. In addition, a complementary checking is included to always verify the current steady-state values during non-event windows to avoid any missing events.

4.4.3 Energy Disaggregation

The basic idea of energy disaggregation is to solve an optimization problem by using the power signatures of each device as

$$P_{agg}(t) = \sum_{m=0}^{M} P_m(t) + e(t) \tag{4.2}$$

where $P_{agg}(t)$ represents the aggregated power signal, $P_m(t)$ is the individual component power signature, and $e(t)$ represents the realistic power deviation from the power signature. In this study, the components (main body, pump, RF generator and heater) of PlasmaTherm are regarded as the individual device for disaggregation. The main body is always on and consumes 900W power. The other component states are illustrated in Table 4.3. As process

recipes set different RF power and processing time, the distinguishable power states include different RF power levels.

CO is a simple and generic technique for solving such a combination problem in the energy disaggregation field [14]. The basic idea of CO is to combine the possible power signatures to find the closest combined signal compared with the real aggregated signal. One drawback of the CO is that it only considers the steady state power values rather than a sequence of power signatures, which can cause mis-classification when the power fluctuates beyond the allowable range or several combinations of the steady-state values are similar or even identical. In Plasmatherm example, the low-vac pump state has the same steady-state power as the 70W RF state, which cannot be resolved by CO. To distinguish this case, we leverage the prior knowledge about working principle differences. The power of pump low-vac state shows a time-varying decrease while the power of RF states is stabilized. We leverage this feature in the post-processing module to distinguish the low-vac state and the 70W RF state and to disaggregate the pump signal with this specific ramp-down waveform. Furthermore, the deviation $e(t)$ is distributed to individual components by considering the instrumentation working principles and operation sequence. The SOP provides the sequence of components being turned active and the prohibited combinations of active components. For example, active pump and active RF are not allowed to occur simultaneously, which is used to distribute $e(t)$ when a component is active. By doing so, the individual component power signal is recovered.

4.5 Experiment Results

The proposed contextual sensor system is evaluated by using PlasmaTherm and demonstrates its capability of the power signal pre-process and machine event classification with the WMI context extraction. In addition, we tested the disaggregation method on another

machine (E-Beam) to further validate its performance.

4.5.1 Machine Event Detection and Disaggregation.

We tested the proposed method on three different cases: PlasmaTherm for PECVD, PlasmaTherm for RIE, and Electron Beam Evaporation (E-Beam) tool, to show the effectiveness of the machine event detection and disaggregation.

We deployed the contextual sensor system on PlasmaTherm to extract events without human interventions. It is noted that the extracted events of PlasmaTherm represent the power events of steady state transitions, including pump, RF, and constant power heater. The pulsing mode power events will be pointed out separately since the pulsing events do not involve WMI but are attributed to the automatic temperature control. Fig. 4.3(c) plots the filtered signal during a PECVD process with SiH_4 gas and several O_2 clean involved. There are 28 power events during this process and the software defined sensor algorithm detects 34 power events including all the 28 ground truth power events with additional 6 events. The extra detected events do not affect the disaggregation result as they are not classified as machine state transitions in disaggregation.

To evaluate the energy disaggregation performance, we collect the actual individual power signals for each component as the ground truth data.Fig. 4.5 depicts a typical segment of the machine event detection and component energy disaggregation results in PECVD case. The specific waveform of the pump is successfully disaggregated. For the RF signals, there are relatively small deviations between the ground truth and disaggregated data since in practice the RF generator needs to adjust its power slightly depending on operational conditions to maintain the generated plasma power constant. Fig. 4.5(e) displays the disaggregated signal for the heater observing that the constant power heater signals are recovered. For the pulsing mode heater, the proposed approach can identify the start and end of the pulses with the

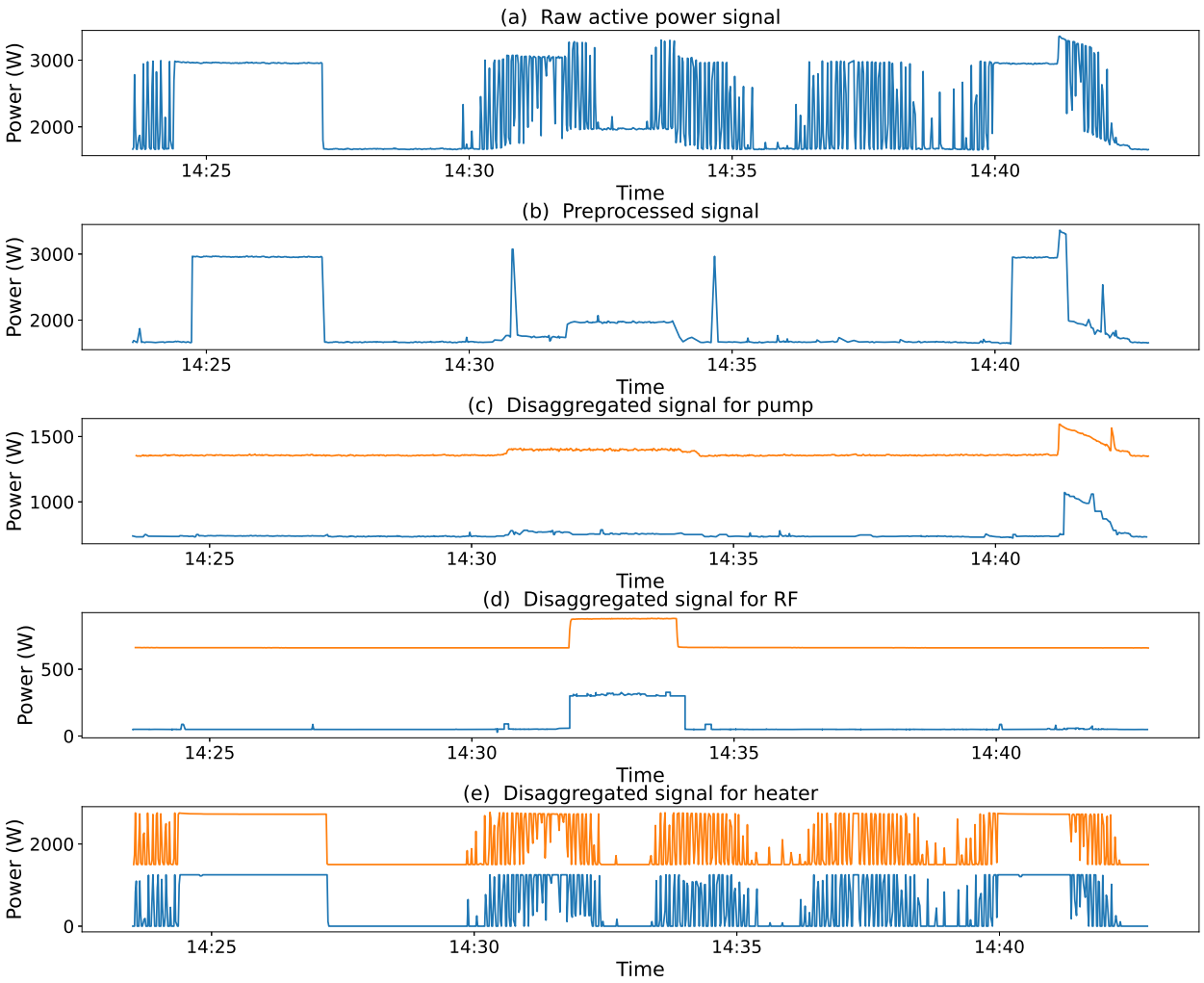

Figure 4.5: An example of the measured raw power signal during a PECVD process and its disaggregated component signals. (a) the captured raw signal with pump, RF and heater being active. (b) the signal after removing pulses by the first stage of preprocessing. In (c), (d), and (e), the disaggregated component signal (in blue) and the ground truth signal (in orange) are plotted for pump, RF, and heater respectively. Orange lines are lifted for better views.

capability of roughly extracting the pulsing heater signal.

Fig. 4.6 illustrates an example of a RIE process performed in PlasmaTherm, where RIE does not require elevated temperature setting but in fact heater is active for a few seconds. The reason is that during non-PECVD processes including RIE and machine standby, the heater temperature is set to be 23 °C close to the cleanroom temperature. When the thermocouple detects temperature deviations (below 23 °C), it will trigger the heater to be on, resulting

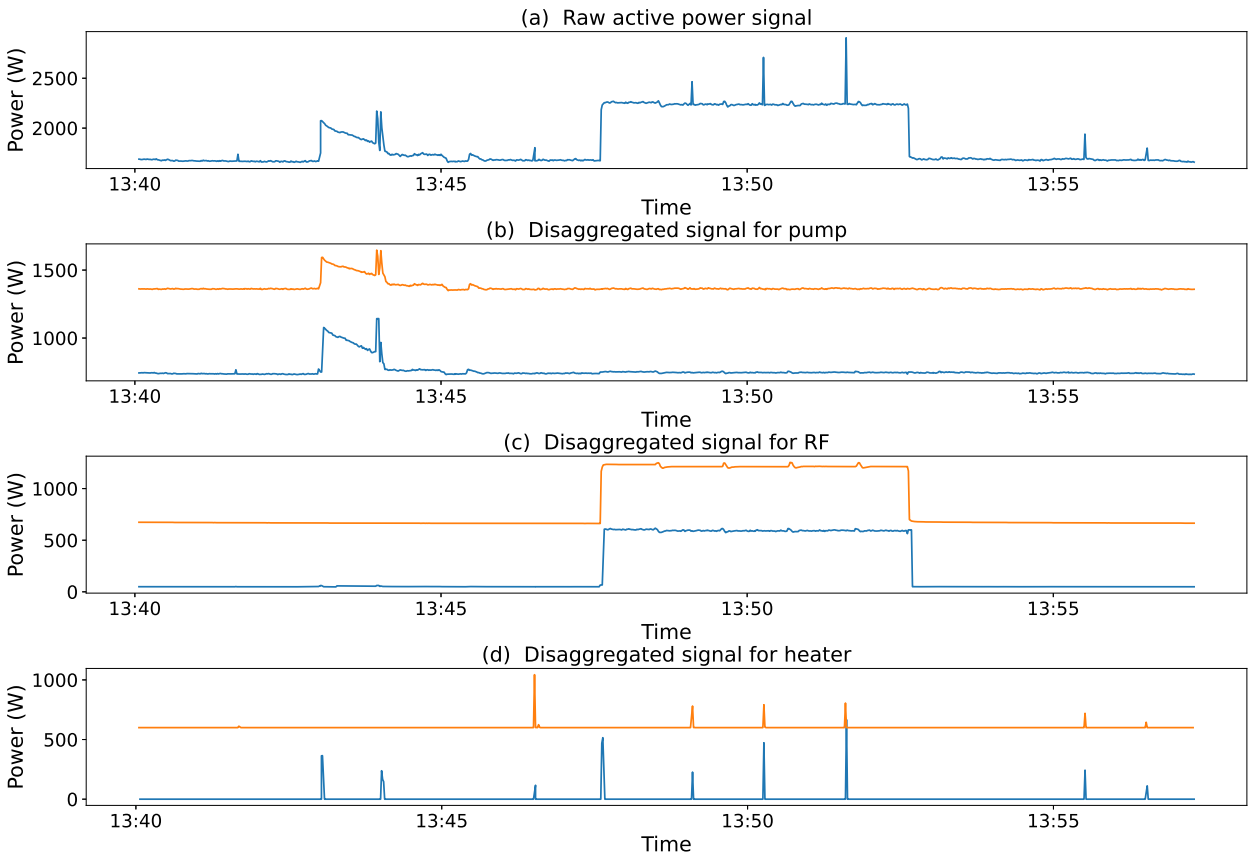

Figure 4.6: An example of a RIE process. (a) the captured raw signal with pump, RF generator, and heater. The red lines show the detected power events. In (b), (c) and (d), the disaggregated component signal (in blue) and the ground truth signal (in orange) are plotted for pump, RF and heater respectively. Orange lines are lifted for better views.

in several spikes in the heater power and main power. Other spikes belong to noise from the CTs. Without the pulsing heater involved, power events in RIE are easier to recognize than those in PECVD. Fig. 4.6(b) and (c) show the successful disaggregation of power signals for the pump and RF during RIE respectively.

From Feb. 19, 2021 to Mar. 24, 2021, there are 17-time PlasmaTherm usages with 15 for RIE and 2 for PECVD. In total 103688 data points (around 29 hours) of raw signals during active machine usages are collected from a power meter. Table 4.4 lists the usage information during this period with time, process, the number of events (in the column, the values stand for pump, RF, and constant power heater in order), the number of detected events, and the number of data in active modes for each component. To provide a quantitative evaluation,

the mean absolute error (MAE), root mean square error (RMSE), and mean percentage error (MPE) are calculated for comparing predicted signals and ground truth signals. Table 4.4 also shows the MAE (Watt), RMSE (Watt), and MPE (%) of each usage with the average values of all the data points. Since the heater signals have 0 values which cannot be used to calculate MPE, MPE is omitted for the heater. In total there are 47 pump events, 67 RF events, and 11 constant power heater events, and they are all successfully detected and classified by the proposed method with related event contexts being extracted. During each usage, the active RF states account for most of the usage time and power consumption as workers prefer running long-time oxygen clean before and after the process. For the PECVD processes, even though there are in total 11 constant power heater events, in practice a user only sets the temperature once at the beginning and the rest of the constant power heater events are due to the temperature change when a user opens chamber to load and unload a semiconductor wafer and thus decreasing chamber temperature sharply.

An interesting observation can be derived from Table 4.4. An average time for pumping down during RIE can be derived which is 66.6 second. However, the mean pumping down time of usage ID 11 is 90 second, which is significantly longer, indicating a possibility of lower efficiency in the pump or a virtual leak in the machine. It is noted that the time information extracted from power events with our proposed contextual sensor system, *e.g.*, time for pumping down, can be used as a reference for the system to identify unusual pump behaviors, which is of great context value in operation.

To evaluate the usability of the proposed machine event detection and disaggregation method, we further test the contextual sensor system on an E-beam deposition tool with Pump, Electron-gun (E-gun) and Controller as components for metal thin film deposition. This E-Beam is chosen not to be equipped with a PLC, *i.e.*, manually controlled. The E-gun serves as the major processing component to provide high-voltage electron beams to melt metal and its current is adjusted manually by a worker using a built-in current meter. A

ID	Time	Process	#data	#evt	#det. evt	Pump				RF				Heater			
						#act.	MAE	RMSE	MPE	#act.	MAE	RMSE	MPE	#act.	MAE	RMSE	
1	02/19 10:12-11:53	RIE	5478	2, 5, 0	2, 5, 0	118	28.6	31.7	3.6	1809	11.7	40.9	13.2	41	4.5	50.3	
2	02/23 13:40-16:05	RIE	7448	0, 8, 0	0, 8, 0	0	24.5	25.4	3.1	3787	16.3	40.0	13.6	51	5.9	52.5	
3	02/24 11:35-12:25	RIE	2441	2, 2, 0	2, 2, 0	117	32.3	35.8	4.0	1694	10.6	34.3	7.6	14	6.9	52.4	
4	02/25 09:17-10:00	RIE	2318	2, 2, 0	2, 2, 0	110	34.3	38.1	4.3	1358	15.5	35.2	10.0	18	6.8	54.4	
5	03/01 10:00-14:00	PECVD	12010	6, 5, 5	6, 5, 5	542	33.5	40.6	4.2	5520	37.1	58.1	18.9	5895	367.4	596.4	
6	03/10 14:14-16:14	RIE	6384	4, 4, 0	4, 4, 0	227	21.9	27.1	2.8	2411	21.2	45.1	14.9	25	4.8	47.1	
7	03/11 11:18-14:15	RIE	9197	4, 5, 0	4, 5, 0	222	23.5	27.4	3.0	3758	16.2	32.7	13.2	82	2.5	31.3	
8	03/11 15:06-15:21	RIE	818	2, 1, 0	2, 1, 0	115	30.3	39.8	3.7	55	14.0	48.2	17.8	5	7.4	54.9	
9	03/11 15:57-17:21	RIE	4620	2, 4, 0	2, 4, 0	118	22.0	25.3	2.8	1956	17.9	46.8	13.7	31	4.3	40.6	
10	03/15 09:43-11:07	RIE	4504	3, 4, 0	3, 4, 0	191	23.8	28.4	3.0	1738	17.8	36.5	14.7	41	6.5	51.3	
11	03/15 11:08-12:32	RIE	4618	2, 3, 0	2, 3, 0	180	21.6	25.7	2.8	3026	17.7	39.3	10.5	37	5.8	48.1	
12	03/16 11:36-13:06	RIE	4616	2, 3, 0	2, 3, 0	131	22.3	25.8	2.9	3136	15.6	34.8	8.8	41	2.7	33.6	
13	03/17 13:10-15:40	PECVD	8184	4, 4, 6	4, 4, 6	275	21.3	25.7	2.7	2321	24.9	50.1	18.7	3753	165.5	367.0	
14	03/19 12:48-17:18	RIE	14439	3, 5, 0	3, 5, 0	257	23.7	27.3	3.1	5015	37.6	40.1	16.5	95	5.3	54.0	
15	03/22 12:12-13:50	RIE	5454	3, 6, 0	3, 6, 0	232	24.4	30.0	3.1	3000	20.3	44.7	13.9	49	4.9	45.2	
16	03/23 13:05-15:37	RIE	8259	4, 4, 0	4, 4, 0	307	22.1	26.4	2.8	3662	16.7	30.9	13.1	49	4.9	45.6	
17	03/24 13:21-14:14	RIE	2897	2, 2, 0	2, 2, 0	140	24.6	29.3	3.1	1629	14.4	39.5	11.5	21	6.3	51.8	
18	Mean	-	-	-	-	-	25.6	30.0	3.2	-	18.3	41.0	13.4	-	36.0	98.6	

Table 4.4: Usage information of PlasmaTherm with detected component events results

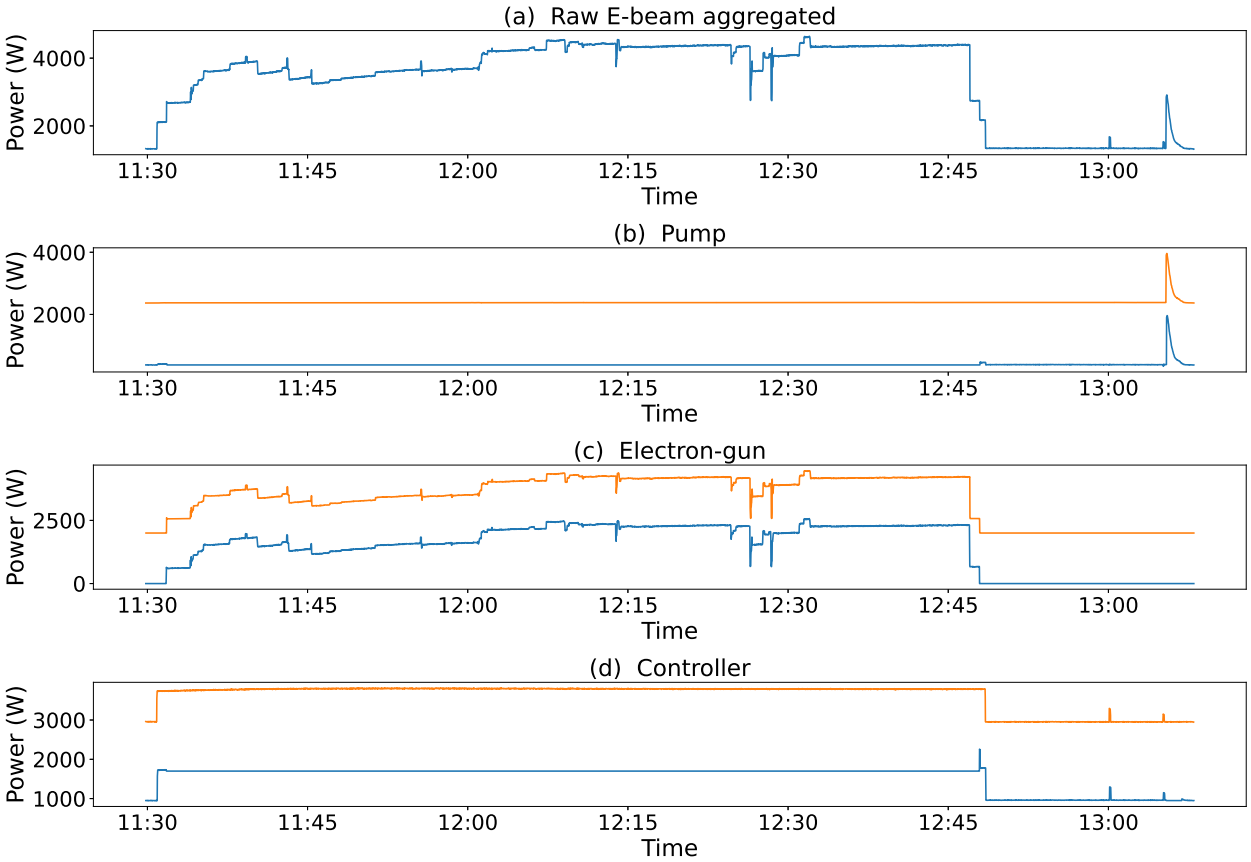

Figure 4.7: An example of a manually operated E-beam metal deposition tool is shown. (a) the measured raw E-beam signal. (b) to (d) the disaggregated (in blue) and ground truth (in orange) power signal for pump, E-gun, and controller respectively. The orange lines are intentionally lifted by 2000 W to keep the curves apart for easier views.

simplified SOP of E-Beam is venting, hoist up, hoist down, pumping down, turning on controller, turning on E-gun, turning off E-gun and controller, venting, hoist up, hoist down, and pumping down again. The pump controls the vacuum steps, and the controller controls the hoists. Compared to PlasmaTherm with PLC, the processing time for E-Beam is manually controlled by a worker meaning that more WMIs are needed to turn off any active components as opposed to a PLC-controlled machine turning off active components automatically. Similarly, from the knowledge of human observer and SOP regarding the WMI sequence and measured power signals, the contextual sensor system is initialized on E-Beam. We used the same algorithm in Section 4.4 but adjusted the threshold parameters to better accommodate individual components. To disaggregate E-gun power signals in the post-processing module, we applied the SOP knowledge that the E-gun will be turned on after controller is turned on. The disaggregated power signals from the measured raw signal are plotted in Fig. 4.7. In Fig. 4.7(d), the two spikes around 13:00 correspond to the hoist-up and hoist-down steps for loading and unloading wafers, which are successfully detected. This experiment further validates the effectiveness of the proposed method.

4.5.2 WMI Context Capture

Since PlasmaTherm is a PLC-controlled machine, the fabrication process can be turned off automatically depending on the process time set by users. Only the positive leading edges, which indicate a component transits from inactive to active modes, can trigger the WMI context capture. There is a distinctive worker gesture difference between changing pump state and RF state as illustrated in Fig. 4.8 that the worker tends to put their hand on the chamber handle to push down when performing the chamber pumping down. This is because the chamber cannot be completely sealed by its own gravity and hence requires extra force to push down. This distinctive WMI context is useful as well. For example, when a worker finishes the process and conducts the pumping down again to keep the chamber under

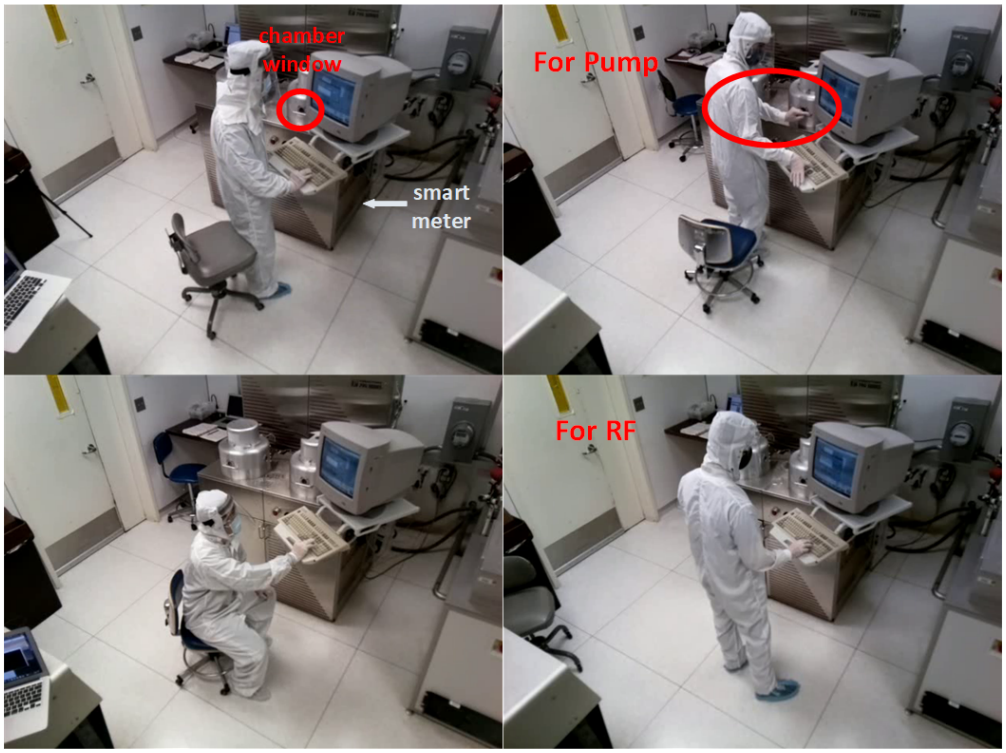

Figure 4.8: Example images of interaction with PlasmaTherm by different users captured through WMI context capture process. The right column shows different interaction gestures to initiate pump or RF generator. Upright is referred as pump action and bottom right is RF action. The upleft indicates the locations of the smart meter installation and the chamber windows used for plasma color detection.

vacuum for protecting its integrity but he/she forgets to push down the chamber handle to tighten the gap, the gas in the chamber cannot be vacuumed to the set pressure. With the WMI context being captured and recognized, this information can be provided to the user to check the machine chamber status and to avoid this incorrect operation.

4.5.3 Event Sequence Context Capture

Fig. 4.9 illustrates an example of the first type context the system can capture, which is a typical machine usage following SOP. During this usage, a user first conducts a 30-min oxygen clean using 300W plasma power, then opens the chamber to load a wafer and conducts pumping down. A 70W oxygen photoresist ashing process is carried out for 1 minute in the

RIE chamber, after which the user opens the chamber to take out the wafer and pumps down again. At last, another 30-min oxygen clean at 300W is applied to clean the chamber for the next user. During this operation, the component state with realistic power, the recipe information (including gas type, running time and plasma power), and the sequence of the operation are extracted and stored in a database. Accordingly, the WMI video clips during each interaction are captured and saved in the local file system according to the response time of each component and gas type.

Fig. 4.10 shows a captured example of a combination of the second and third type context related to pump issues. After a typical RF process, a user conducts pumping down as usual after which an irregular bump is detected by the contextual sensor system. Then, a second pumping down is conducted again by the user. The corresponding contexts during this period are captured. After the first pumping down, from the monitor the user noticed abnormal pressure value and informed the facility staff. From the disaggregated pump power signal, the bump corresponds to the abnormal pressure noticed by the user, which indicates that the gas inside the chamber is not vacuumed to the expected pressure and the air-tightness of the vacuum system is likely faulty. After a second-time pumping down, the pressure becomes normal. The extracted context is saved in the database and can be used as a reference when the same event sequence is met. One of the significances of this captured context is to be potentially used to conduct predictive maintenance and anomaly detection in the future.

4.6 Discussion

Comparison: To further validate the efficacy of the proposed machine event detection methods, we compared our method with some typical previous work with only the machine event detector replaced. We further tested on more data: 80 pump events and 105 RF events for PlasmaTherm RIE, 68 pump events, 80 E-gun events and 220 controller events

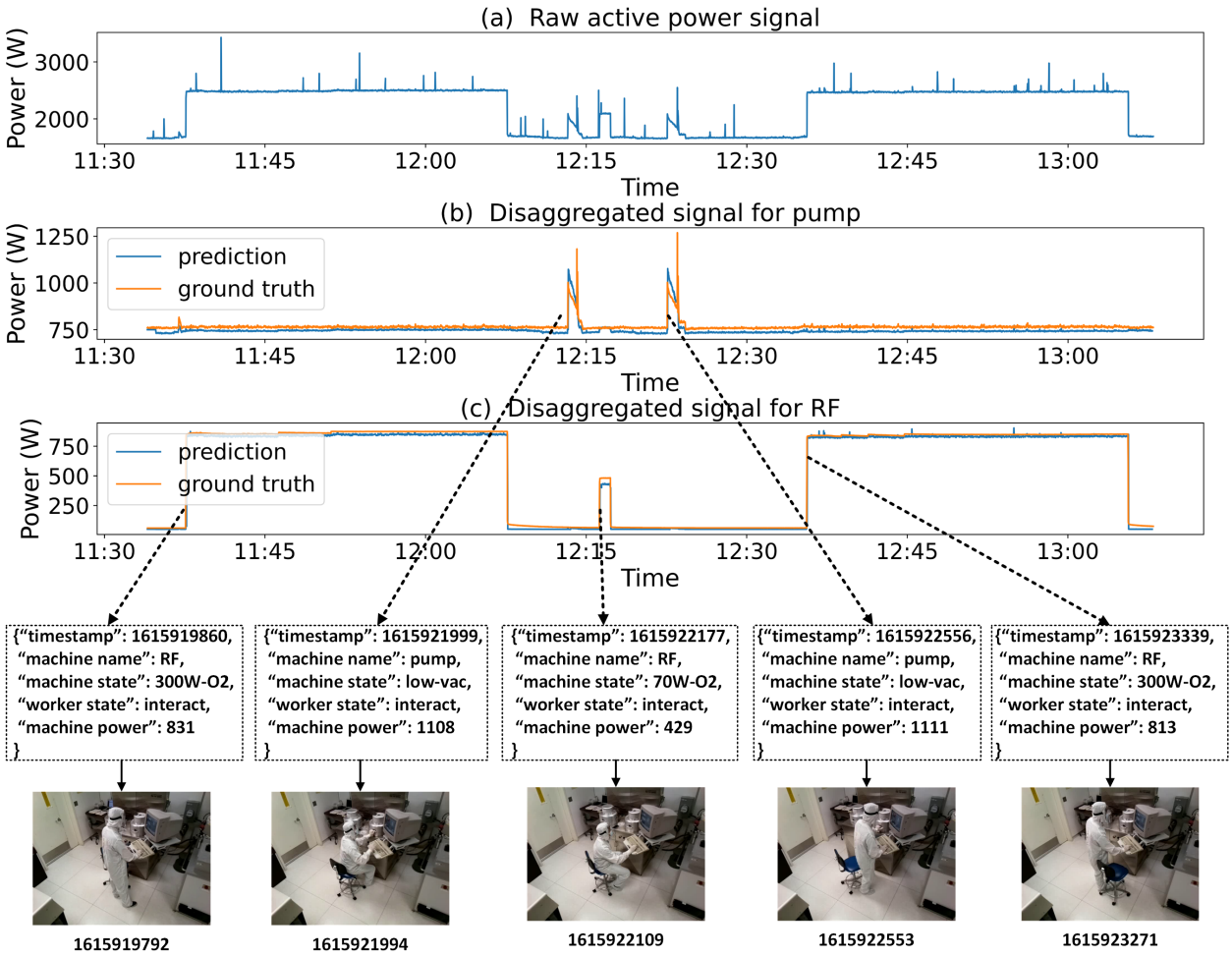

Figure 4.9: A first type of captured context during a RIE process is illustrated. The measured main power signal with disaggregated signals and ground truth signals are plotted. The heater signal is absent as RIE does not need heater. 5 positive edges correspond to 5 events with components from inactive mode to active modes. The extracted event contexts with UNIX timestamp, machine (component) name, state and actual power, and worker state are formulated in a JSON-format. The 5 corresponding WMI contexts are shown with the captured timestamp.

for E-Beam, and the two PECVD usages. We considered the precision (P) and recall (R) as the metric for the machine event classification. As shown in Table 4.5, Section 4.6, and Table 4.7, our approach achieved better performance on all the three test cases. Particularly, the proposed method can handle the heater pulses while other methods fail to detect the pulsing mode heater as well as other events with heater pulses in a low sampling frequency. This is achieved by the segregation of the heater pulses and steady state signals. Statistical

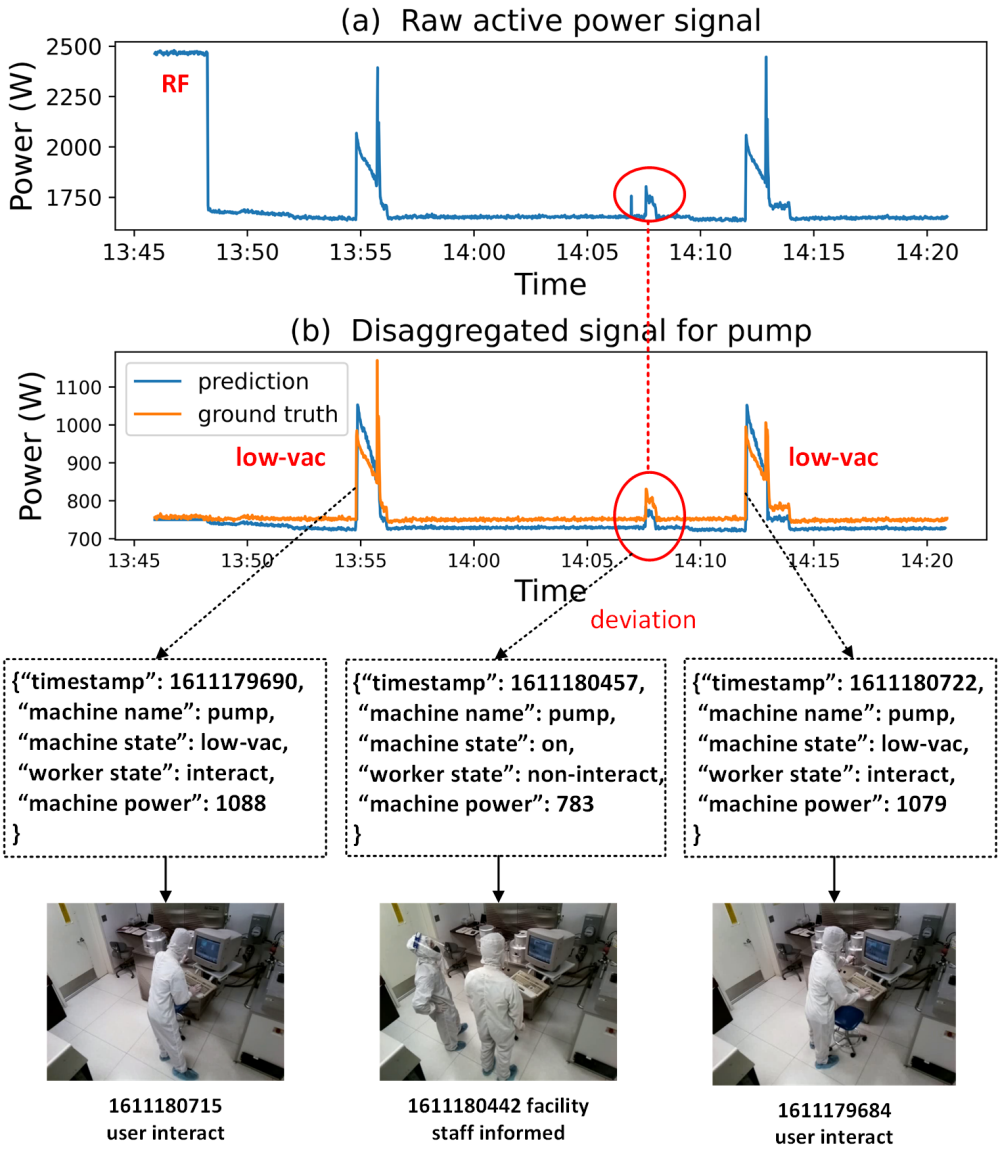

Figure 4.10: A combination of the second and third type of contexts is captured and illustrated. Between two pumping down (low-vac state), a small bump with actual power deviates from the average of pump on state. The corresponding WMI contexts are shown. During the anomaly occurrence, the facility staff is informed and checks the machine status.

methods are widely used for abrupt steady state changes but are not effective with the heater pulses in this study as the statistical features of the signals are not stable. To extract the contextual information, it is essential to extract the signal envelope and keep the signal envelope undistorted while detecting the heater pulsing states at the same time. Because the WMI context requires reliable detection of component state transition time and the context

Method	Pump		RF	
	P	R	P	R
GLR[6]	0.896	0.863	0.864	0.848
Chi square[69]	0.951	0.963	0.970	0.914
Rehman et al.[115]	0.950	0.950	0.981	0.971
Ours	**1**	**1**	**0.99**	**0.99**

Table 4.5: PlasmaTherm RIE Event Classification Comparison

Method	Pump		RF		Heater		Pulsing Heater
	P	R	P	R	P	R	
GLR[6]	0.75	0.6	failed		0.526	0.909	not able
Chi square[69]	0.667	0.6	failed		failed		not able
Rehman et al.[115]	0.7	0.7	0.12	0.333	0.524	1	not able
Ours	**1**	**1**	**1**	**1**	**1**	**1**	**can process**

Table 4.6: PlasmaTherm PECVD Event Classification Comparison

of abnormal machine states (as the example in Fig. 4.10) requires disaggregated component-level power signals. We further tested the wavelet thresholding to remove the heater pulses as shown in Fig. 4.11. Compared with our method in Fig. 4.5(b), the wavelet thresholding as well as other regular low-pass filters can remove the pulses in some regions but highly distort the underlying signal of steady state machine components. The reason is that the pulses are not the true noise but the result of feedback control of temperature. Sometimes the heater stays active longer than several seconds as we can observe in Fig. 4.5, which causes that in some regions the heater pulses can have similar frequency as the base signal. In fact, the pulses have different frequency distributions with the true sensor noise (*e.g.*, white noise), are not random, and are correlated through the feedback control. In our two-stage preprocessing method, we use the frequency analysis to identify the start and end of pulses and apply the prior knowledge of machine components to provide thresholds for removing the pulses from the base signal, instead of filtering the pulses in frequency domain.

Discussion: The FSM model generated from SOP defines the state transitions of machines and workers, and the causality between worker and machine states. This model not only

Method	Pump		E-gun		Controller	
	P	R	P	R	P	R
GLR[6]	0.853	0.941	0.923	0.900	0.867	0.950
Chi Square[69]	0.958	1	0.963	0.963	0.882	0.955
Rehman et al.[115]	0.932	1	0.904	0.938	0.932	0.932
Ours	**0.986**	**1**	**1**	**0.975**	**1**	**0.964**

Table 4.7: E-Beam Event Classification Comparison

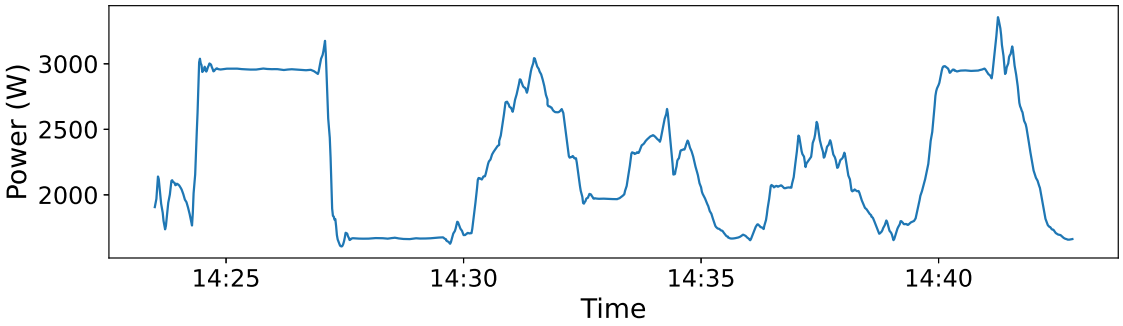

Figure 4.11: An example of the processed raw signal by wavelet thresholding.

helps the determination of machine component states and worker states independently, but also opens a new way of leveraging the causality to capture the contexts and further to achieve automated data labeling for ML. This paper demonstrates the effective state detection and energy disaggregation of machine components and the context capture capability by using the FSM-based SOP model. The prior knowledge of SOP and instrumentation principles alleviates the requirement of data collection for supervised energy disaggregation and the requirement of high sampling rate of power meters. The automated data labeling can be achieved by extending this study leveraging the causality between worker actions and machine responses defined in the FSM-based SOP model. For example, recognizing worker gestures for interacting with machines is important for operation integrity. An ML model for action recognition can be trained with the collected video snaps of worker interaction moments effortlessly without any manual data collection and annotation. More importantly, each machine can have a totally different interface requiring different gestures for worker interactions. This automated data labeling method enabled by the proposed FSM-based SOP model can enhance the adaptability of ML models to dynamically adjust to different

machine interfaces during deployment. The underlying concept of the causality induced automated labeling can easily be extended to various interactive activities between two objects in manufacturing. It brings a new angle of understanding manufacturing interactions. Given an assembly line as an example, materials are processed by workers or machines stage by stage with intermediate quality inspection to sense material properties. If the response time between worker/machine-material interactions and material state transitions can be derived, the proposed concept can be applied to use the material state change to capture the worker/machine-material interaction contexts and environmental contexts to assist the product quality inspection and potential automated labeling.

Furthermore, the sequence of machine component events with anomalies are extracted with the corresponding WMI contexts. The context extracted by the proposed novel method is important in terms of several practical applications. For example, the extracted event contexts can be used to identify the integrity of worker and machine operation compared with SOP at the component level. Any deviations can lead to immediate malfunctions or unnoticed tool wear and tear accumulated to cause a serious machine breakdown in the future. The captured anomaly contexts due to the effective disaggregation of component signals can assist the development of prognosis applications. On the other hand, the disaggregated power signals with the color information from gas plasma emission can be exploited for identifying the gas type, processing duration and plasma power level, implicating a stable processing condition for manufacturing quality control.

Application Restrictions: The proposed context capture is based on the SOP-defined worker and machine state domain. There are two constraints. The first constraint is that if some unknown state occurs beyond the SOP, the system could fail in detecting the state. A perspective is to leverage the worker intelligence and predefined worker states indicating abnormal machine states and followed by unsupervised clustering methods to detect the data similarity to form a new class data for the machine and worker state detection. The

second constraint is that while this study uses power signals to successfully detect machine responses of energy states, there are some cases that machines do not respond in a way of energy consumption, such as the state of material loading. Other responsive sensors for these undetected states, such as acoustic and IR sensors, can be selected to detect the machine operation within the same cause-and-effect concept. Similarly, the corresponding worker interactions can be captured by using these additional channels of sensors. While these limitations may impede the application of the contextual sensor system, the proposed remedies shed a light on new research directions for future improvement.

Chapter 5

Interactive and Adaptive Learning Cyber Physical Human System

5.1 Introduction

The development of advanced machine learning (ML) algorithms and hardware equips smart manufacturing (SM) systems with ML-based cognition models and ML-based Cyber Physical Systems (CPS) to augment insight and contextual awareness. However, typical supervised learning for specific application scenarios demanding manually annotated data renders ML models unable to adapt to dynamic environments and unforeseen circumstances without additional hand-labeled datasets [139]. Despite laborious work, the collected data for a specific manufacturing task is difficult to be reused for a new task due to technical issues (*e.g.*, less common data features [35]) and users' concerns (*e.g.*, privacy and IP [111]). Several unsupervised or semi-supervised learning algorithms have been created to mitigate the barrier of intensive data annotation by statistical feature and representation learning but still require labels for downstream classification tasks [153, 162]. They generally perform worse than

supervised methods, and can fail when data distributions are nonstationary in dynamic environments [109]. Therefore, a methodology for establishing a ML system with adaptive learning capabilities that broadly extend to many SM applications is of great interest and requires more investigation.

Adaptive learning systems aimed at learning from streaming data in an ever-changing environment are gaining research attention [163]. Ideally, an adaptive learning system with ML detectors should be automated [74], where training data can be collected and annotated automatically without human intervention. This will be particularly valuable in manufacturing environments where adaptive dynamic control offers significant advantages that would not be available if the methods required real-time human intervention from IT professionals. The scope of this study is focused on this important, largely unresolved challenge of SM.

During production, interactions among humans, machines, and materials are pervasive. Much like the Industry 4.0 concept has revolutionized manufacturing via integration of information technology and operation technology, a new concept, Operator 4.0, is emerging to address humans' critical roles in terms of operational efficiency, adaptive feedback, and improved productivity. While novel technologies have been proposed for reliable connections with workers (*e.g.*, portable devices), workers are naturally connected with manufacturing systems through their active and reactive interactions with machines and materials. The interactions between workers and machines contain meaningful contextual intelligence in operation integrity, worker intention prediction, and anomaly detection of abnormal machine conditions. For example, active worker interactions performing improperly can cause machine malfunctions, and reactive worker interactions towards abnormal machine states engage workers' intelligence in perceiving and managing anomalies. These practical reasons necessitate a reliable way to detect interactions. While vision-based supervised ML models have demonstrated effective human action recognition, the practical constraints in manufacturing (*e.g.*, wide variety of machine interfaces, nonstationary worker behavior, and worker

mobility) complicate the model generalization to cover broader and unforeseen cases. To overcome these deployment barriers, it is essential to develop ML systems capable of evolving and adapting without human annotation to recognize ever-changing interaction gestures. To achieve this, more research work is needed to address the challenges in: 1) adapting ML models to unpredictable human nature and variable machine interfaces; 2) automating the model adaptation process without human intervention; and 3) developing a generic solution for various manufacturing environments.

To address these challenges, we propose in this paper an Interactive Cyber Physical Human System (ICPHS) driven by the correlation underlying interactive manufacturing processes involving workers using machines to design an adaptive human-machine interaction (HMI) recognition model. Manufacturing interactions occur on two or more objects restricted by compiled instructions developed by manufacturing engineers (*e.g.*, standard operating procedures SOP), and have a mutual effect upon one another. During an interaction the action of one object (*e.g.*, worker) can cause a reciprocal response of the other object (*e.g.*, machine or material), which describes the common causal relationships among interactive objects. By leveraging this causality among interactions, the system can collect and self-label one object's data by using the other object's status for retraining and improving the ML model. To demonstrate a real-world utility of the ICPHS, a case study in two machines, one being fully automated with a programmable logic controller (PLC) and the other being purely manually operated, is conducted in a multi-user semiconductor manufacturing facility. We applied energy disaggregation techniques on power signals to detect machine state changes in real time to self-label worker actions. The worker actions are detected by pose estimation and a Graph Convolutional Network (GCN) from video data. The GCN is retrained adaptively by the self-labeled dataset to achieve automated adaptation. The experimental results successfully show the proposed ICPHS capability to adaptively improve accuracy and significantly reduce data collection and labeling efforts.

In brief, our novel contributions include: 1) We propose to use the causality of HMI to achieve a self-labeling method for ML adaptation by conceptualizing and designing a HMI correlation model with temporal and causal relationships to capture an interaction window; 2) Generalization to a variety of human roles and prior domain knowledge embedded in machines or acquired from humans in ICPHS design; 3) Increased accuracy of automated adaptive learning by leveraging the advantage of supervised learning while significantly reducing human labeling efforts; and 4) Demonstration of excellent potential to achieve class incremental learning with more interaction types being recognized through the retraining by self-labeled data.

5.2 Related Work

To design the adaptive ICPHS, we review the relevant progress in adaptive and self-supervised learning as well as in Cyber Physical Human Systems as a benchmark.

5.2.1 Adaptive and Self-Supervised Learning Applications

Adaptive learning solutions mostly focus on the concept drift (CD) problem that the relation between input data and output labels changes over time [41]. Several studies have been proposed, such as designing CD detectors to analyze data drift [66], introducing experts and adaptive mechanisms to react with experts [13], and ensemble learning to deal with novel class arrival [67]. Most current research in adaptive system focuses on novel algorithms to accept new data for back-propagation demanding that labels are already available or CD can be analyzed. However, one of critical challenges is the unpredictable CD with multiple variants. We devise a novel method that the system can automatically label data, clean data, and use them to retrain the model.

A classical example of self-supervised learning was proposed in [27] that hearing mooing and seeing cows tend to occur together. Recently, several studies applied self-supervised techniques in representation learning by using different attributes of intrinsic data features [106, 45] or multimodal information from environments as the self label [103]. Moreover, several studies investigated object visual feedback for robots learning tool affordance through robot-object interaction trials [132, 96]. Inspired by these ideas, we further focus on the causal correlation between multiple objects involved in manufacturing interactive activities where the data attribute of one object can serve as the supervision for the other object and vice versa.

5.2.2 Cyber Physical Human Systems

Human factors have recently become a crucial element in CPS design for effective inclusion and leveraging human intelligence to augment the decision making as indicated by many conceptual CPHS designs in various fields [68, 152, 46]. Particularly, Madni et al. proposed a conceptual adaptive CPHS where humans and CPS can mutually adapt and learn from each other to enhance cognition [94]. While pointing out the role of CPS and human in the adaptation process where humans serve as supervisors to assess CPS behaviors [94], they emphasize the need to achieve automated adaptation without human supervisors. In this paper, we design and implement such virtual supervisors for ML-based CPHS to adapt by observing ongoing HMIs from different operation aspects.

5.3 ICPHS Methodology

The proposed ICPHS focuses on scenarios of interactive manufacturing work, where multiple people and machines are involved. In Fig. 5.1, three phases for ICPHS design framework

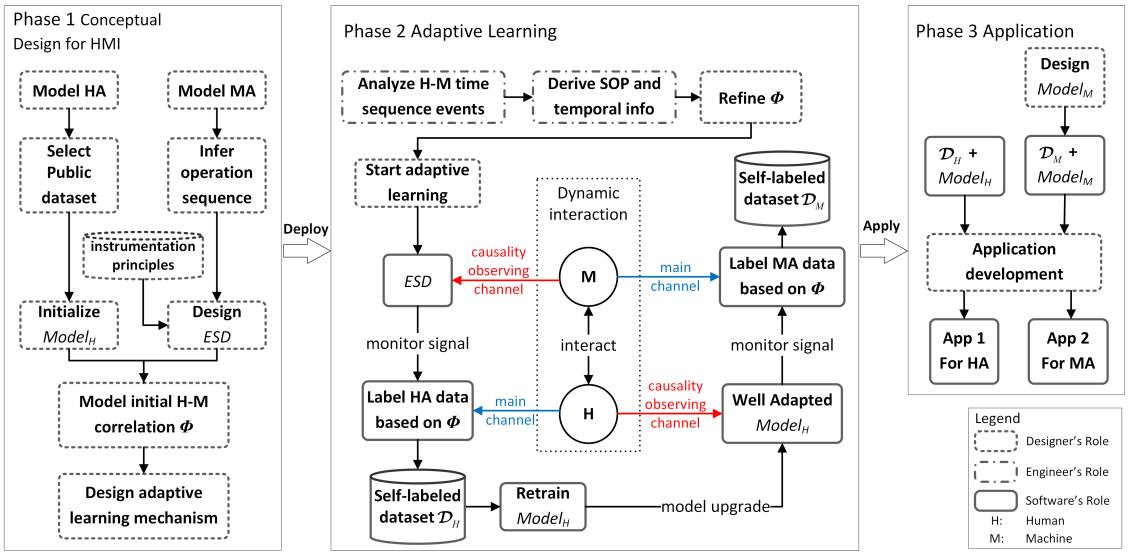

Figure 5.1: The ICPHS framework illustrating a conceptual ML design and learning workflow for manufacturing interaction scenarios.

are illustrated, where phase 1 is a static conceptual design to develop HMI models, phase 2 is a deployment phase to acquire dynamic information from HMI to self-adapt ML models with minimum human intervention, and phase 3 is application development to align self-adapted models to opportunities for manufacturing predictive intelligence. Several types of human roles are involved in the ICPHS to work collaboratively for a comprehensive evolving ICPHS, namely system designers who are data and computer scientists from data service providers, and onsite workers including operators, engineers and technicians from manufacturers. System designers undertake the tasks to design and implement the data service with adaptive learning software. Workers interact with machines to conduct equipment operation for production and provide necessary floor information for system designers.

Phase 1 is to accomplish a static goal of data-driven management of any generic manufacturing equipment by using known information from machines and human experience before deploying sensors to acquire signals from machines and humans. This ML design principle allows extended and scalable applications to various manufacturing fields. In a factory, workers generate human activities (HA) when interacting with machines to operate, inspect, or repair in the form of physical movements (*e.g.*, hand, foot, body) and functional objec-

tives (*e.g.*, running recipe, developing SOP, troubleshooting). Based on such knowledge and machine manuals, possible HA gestures and orders are estimated by system designers and modeled in the form of logic state transition. An unformed HA state detector using a vision-based ML model $Model_H$ with data from visual cameras is then built from selecting public dataset resembling estimated HA. Similarly, machines are composed of multiple functional components (*e.g.*, heating, vacuum pumping, spindle) and generate machine activities (MA) induced by human interactions in the form of logic operation sequences and functional tasks. With machine manuals and basic engineering knowledge, MA are modeled as logic state transitions and the component operation sequence can be inferred. An unsupervised MA state detector such as energy state detector (ESD) with data from power meters, which can be regarded as a naive classifier for energy events, is built leveraging instrumentation principles of machine components [116]. With HA and MA models as well as inferred operation sequence, an initial H-M correlation model Φ based on the temporal relationship and causality between HA and MA is built as the foundation of adaptive learning mechanism. For example, a machine component is turned active because of a worker's operation towards a control panel a few moments earlier. Such correlation can be used to design the data self-labeling mechanism to achieve adaptive learning. When phase 1 completes, sensors and initial ML software are deployed at the manufacturing floor to proceed to phase 2.

In phase 2, real time video and power data are accessible through a GUI to be analyzed. The HMI time sequence events are analyzed by field engineers to derive SOP and temporal information for each machine. The analyzed information is transmitted to remote system designers to refine Φ and start the automated adaptive learning. Note that for mature manufacturing processes, prior developed SOP can be readily accessible at phase 1 to ease the design of Φ and field engineer involvements. The first 3 steps at phase 2 can be iterated several times to refine and evaluate Φ. Now, the refined adaptive learning software is ready for tracking HA and MA and collecting real time data without the need of human intervention. During dynamic HMI, information retrieval from two sides can be defined as main channel

and causality observing channel, and their roles can exchange depending on tasks. Main channel is defined as the channel to optimize its performance through self-labeling. The causality observing channel is defined as the channel that validates events happening on the other side involved in HMI to generate monitor signals for annotating main channel data. Initially, the machine side is regarded as a causality observing channel to feed power signals of individual machine to its ESD to detect current component states (*e.g.*, on/off) of a single machine. Based on Φ, the system knows mappings between machine state transitions and HA with temporal sequences and responses, which are leveraged to label the corresponding HA segment with the machine state transition information. After some duration of data self-labeling and collection, a self-labeled HA dataset $\mathcal{D}_H$ is generated to retrain $Model_H$ for better HA recognition accuracy. In addition, the self-labeling based on machine state transition can capture HA differences regarding various machine/component operation, enabling $Model_H$ upgrade to recognize more HA classes with more fine-grained $\mathcal{D}_H$. After several rounds of adaptive retraining, a well-adapted $Model_H$ is established. Note that this downward branch for optimizing $Model_H$ can be accomplished several times independently for multiple machines to derive $Model_H$ and $\mathcal{D}_H$ for each machine. Similarly, the well-adapted $Model_H$ can serve as the estimator of the causality observing channel to annotate the main channel MA because $Model_H$ is able to recognize individual HA for operating a specific machine component. As a result, a self-labeled MA dataset $\mathcal{D}_M$ is established.

In phase 3, applications aligned to manufacturers' interests can be designed with self-labeled datasets and ML models on behalf of humans and machines. Novel ML models, such as $Model_M$ or advanced $Model_H$, can be designed or redesigned based on applications.

To achieve the adaptive learning, the correlation between humans and machines needs to be identified and modeled as Φ. Different from human daily activities, manufacturing worker activities generally follow assigned task schedules and machine operation manuals or procedures (*i.e.*, SOP), which constrain the degrees of freedom of HA and MA and thus ease

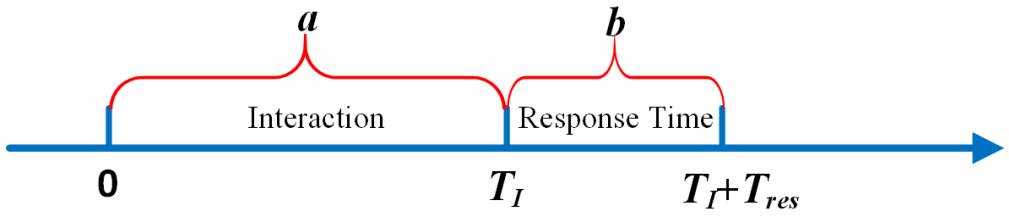

Figure 5.2: An illustration of HA and MA causal temporal relation for traceback.

model design. For operating a machine, the SOP defines a sequence of HA to follow, resulting in a sequence of MA. The SOP is modeled as a series of events, where each event contains information about location x, time t, worker state v, and machine state q. Finite State Machines (FSM) are used to model HA and MA as states and their collaborative state transition function δ as

$$q_{i+1} = \delta^m(q_i, v_{i+1}, sop_{i+1}), q \in Q, v \in V \tag{5.1}$$

$$v_{i+1} = \delta^h(v_i, q_i, sop_{i+1}) \tag{5.2}$$

where the machine and worker state space Q and V are predetermined from SOP, and index i represents steps. Superscript h and m indicate human (worker) and machine respectively. Q consists of two parts, Q^{reg} and Q^{ir}, respectively representing normal operational states and irregular states such as malfunctions. V involves regular worker states and predefined worker states indicating observed irregular machine states. Eq. (5.1) and Eq. (5.2) define the machine (or worker) state transition with feedback from worker (or machine) state and predefined SOP respectively. Eq. (5.1) and Eq. (5.2) implicitly define v_i is the cause occurred earlier at t_i^h and q_i is the effect occurred later at t_i^m as a process initiated by a worker.

To capture and perceive information from HA and MA, the information leakage during human or machine state transition is critical to identify state differences and characterize

current states. HA observations o^h during worker state transitions can be modeled as

$$o^h_{i+1} = F(v_i \rightarrow v_{i+1}, v_{i+1}) + N^h \tag{5.3}$$

where F is the HA observation function depending on sensor type, and N^h represents a random variable denoting the noise independent of F. Given o^h, ML models can be designed to infer the worker state as written by

$$\hat{v}_{i+1} = \hat{F}((o^h_{i+1-s^h}, ..., o^h_{i+1}), \alpha) \tag{5.4}$$

where $\hat{F}$ is a trained ML estimator for HA, α is the learnable parameter, s is window size for input data, and $\hat{v}$ is the inferred worker state. Likewise, MA observation o^m during machine state transition and statistical or ML enabled MA estimator ($\hat{G}$) can be written as

$$o^m_{i+1} = G(q_i \rightarrow q_{i+1}, q_{i+1}) + N^m \tag{5.5}$$

$$\hat{q}_{i+1} = \hat{G}((o^m_{i+1-s^m}, ..., o^m_{i+1}), \beta) \tag{5.6}$$

where G is the MA observation function, $\hat{q}$ is the estimated machine state, and β is the learned parameter of $\hat{G}$. Eq. (5.6)) defines a general formula for inferring $(i+1)th$ machine state from MA observations that can be multi-sensor fusion.

Either HA or MA can be a main or causality observing channel. For example, a worker is main channel and a machine is causality observing channel. With the SOP being executed, q varies through information exchange with the worker. With $\hat{q}_{i+1}$ estimated from $\hat{G}$ and knowing the last machine state $\hat{q}_i$, $\hat{v}_{i+1}$ can be determined from Eq. (5.1). Consequently, the state transition correlation can empower the self-labeling of main channel data. An essential design parameter is the temporal relationship between human and machine state transitions as they do not always occur concurrently. Fig. 5.2 shows an example that if the interaction starts at time 0, the machine state change will be recognized at time $T_I + T_{res}$,

where T_I is interaction duration and T_{res} is response time. The temporal relation is to start from $T_I + T_{res}$ to pinpoint the interaction period $(0, T_I)$. To adequately cover more informative data segment due to time variations, two values a and b, representing respectively one-step and two-step backtracing step sizes, need to be predetermined. Since T_I and T_{res} normally follow Gaussian Distributions, selecting a and b as the mean can derive a minimum expectation of duration mismatch.

5.4 Case Study in Semiconductor Fabrication

We describe a case study for the ICPHS validation in a multi-user semiconductor fabrication facility. Two types of machines, *i.e.*, automated with PLC and manually controlled without PLC, are selected. The former's process can automatically transition to the next step without HMI according to preset recipes and SOP, while the latter requires additional HMIs for machine functional component state changes. A PlasmaTherm tool with PLC and a manually controlled E-Beam tool are selected. A generalized SOP for them is illustrated in Table 5.1. Each machine is equipped with multiple functional components and requires one worker to operate through interfaces.

Fig. 5.3(a) describes the complete real time data processing pipeline. A webcam and a three-phase power meter are deployed for each machine to collect real time videos of machine surroundings as main channel and power signals of the entire machine as causality observing channel. The video stream is partitioned into segments and fed into a two-step cascaded ML models as HA estimator to recognize actions that are then associated with a given machine based on the spatial consistency. Meanwhile, the power signal is processed to identify machine component states. A correlation and confirmation module is included to compare the two information streams and complete the self-labeling. Important modules are explained as follows.

Step	PlasmaTherm			E-Beam			
	Pump	RF	Heater	Pump	CNTLR	Hoist	E-gun
Verify OK	on	stby	off	on	stby	off	off
Temp. set	on	stby	on	on	stby	off	off
Vent	on	stby	on	on	stby	off	off
Load	on	stby	on	on	stby	on	off
Pump down	low-vac	stby	on	low-vac	stby	off	off
Run process	on	on	on	on	on	off	on
Purge/Vent	on	stby	on	on	stby	off	off
Unload	on	stby	off	on	stby	on	off
Pump down	low-vac	stby	off	low-vac	stby	off	off

Table 5.1: A generalized SOP of PlasmaTherm and E-Beam in semiconductor fab with component state transitions

5.4.1 HA: Worker Action Recognition

To preserve worker privacy in working environments, we apply OpenPose [21] as the first step to extract skeletons of Body25 type with 15 joints excluding the head and foot joints. To extract features and learn representations from graph-structured skeletons, GCN is explored extensively, including multi-scale GCN (MSGCN) for capturing multi-scale structural features from non-local neighbors [88]. We modify the GCN with multi-scale connections and the layer-wise structure is shown in Fig. 5.3(b).

A human skeleton, composed of joints and bones, is denoted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of N nodes (joints), and $\mathcal{E}$ represents the edges (bones). A graph can be represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$ and a feature tensor $X \in \mathbb{R}^{T \times N \times C}$, where T is frame number and C is the channel number. The multi-scale connections can capture long-range node features and are achieved by the k-th polynomial of A. The graph convolution at layer $l+1$ is achieved by $X_t^{l+1} = \phi(\sum_{k=0}^{K} A^k X_t^l W_k^l)$, where $\phi(\cdot)$ is the activation function and W is the learnable weight matrix which is a 1×1 convolution layer. This can be intuitively understood as a spatially weighted aggregation of the neighbor node features. The temporal

graph convolution (TCN) is achieved similarly along the temporal dimension.

Three MSGCN-TCN-TCN sub-modules are stacked to formulate a complete model with pooling layers and fully connected (FC) layer at end. The number of channels of sub-module is set to be 96, 192, 384. The number of training epochs is 65, and the learning rate is 0.05 initially with step degradation at epoch 45 and 55. Adam optimizer is used in the training with weight decay of 0.0005 and batch size of 32.

5.4.2 MA: Energy Disaggregation

In [116], we illustrated a method to detect and classify power events and conduct unsupervised energy disaggregation as ESD. The basic idea of energy disaggregation is to solve an optimization problem by using power signatures of individual machine functioning components to search for possible combinations and find the closest combined signal compared with the actual aggregated signal. In this study the goal is to disaggregate the state transition of each component from main power signal. With the developed energy disaggregation method, the component states can be classified in real time to assist the self-labeling of worker actions.

5.4.3 Adaptive Learning Mechanism

The GCN learning process exemplifies the first two phases of ICPHS. During phase 1 before deployment, selected public dataset is preprocessed to pretrain the model. Next, the pretrained GCN is deployed as part of the main channel estimator to start the adaptive learning journey. When new machine state is detected, the response time for the specific machine state transition is looked up to traceback videos saved in the buffer and to automatically label the video over that duration as interaction samples with state transition information. In

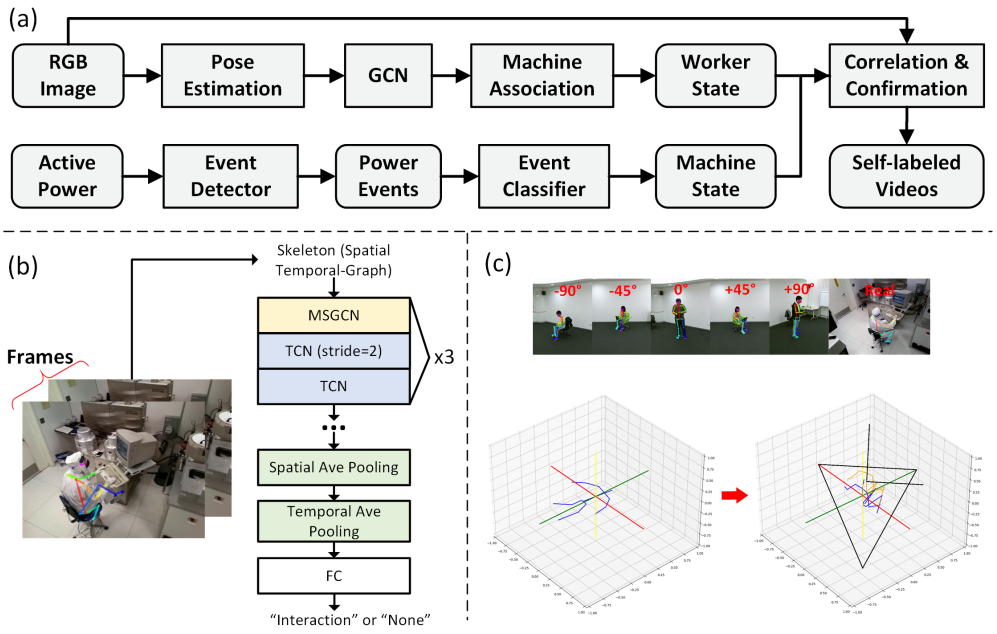

Figure 5.3: (a) Data processing pipeline of the case study. (b) The layer-wise diagram of the implemented GCN with a ReLu and BatchNorm layer after each MSGCN and TCN. (c) The top shows examples of 5 viewpoints from NTU dataset and 1 realistic example of PlasmaTherm case. The bottom illustrates the rotation and projection. Red, green, yellow lines represent x, y, z axis. The left one is the raw 3D skeleton. At the right, the black triangle is the viewing plane determined by three vertices. The two orthogonal black lines represent x and y coordinates on the viewing plane. Blue skeleton is the one after rotation, and the orange skeleton is the one after projection.

addition, a rule-based post-processing filter is designed to eliminate improper video samples. The rules for improper videos are: 1) when including two or more people in the scene due to the difficulty of selecting the main operator with minimum errors; 2) where the worker position is out of pre-defined regions of interest (ROI) for interaction regions or no worker is in the scene. The post-processing filter can further reduce the label noise enhancing the learning performance. When there is a human activity located outside of ROI and the deployed GCN recognizes it as non-interaction, this period of human activity is automatically labeled as negative samples. Until self-labeled interaction samples accumulate to a certain amount, the GCN model continues being retrained to improve detection accuracy with no manual labeling. The retrained model is redeployed and retrained iteratively with additional self-labeled samples until it converges or achieves good enough accuracy.

5.4.4 Public Dataset Preprocessing

NTU-RGB+D 120 is a large-scale dataset including 120 human daily actions with 114,480 3D skeleton samples captured from five viewpoints [87]. Fig. 5.3(c) illustrates the viewpoint discrepancy between NTU dataset and PlasmaTherm case. Current GCN models do not generalize so well in this case. Hence a skeleton rotation and projection preprocessing to convert 3D skeleton features to the 2D target domain is essential to augment the similarity between pretraining and target dataset. Shi et al. provided a coordinate translation that parallels the x axis to the vector from "right shoulder" to "left shoulder" and the z axis to the vector from "spine base" to "spine" [123], which can unify the five viewpoints to be identical facing towards +y axis as shown in Fig. 5.3(c). Next, the rotated skeletons should be projected to a 2D viewing plane. Referring to Euclidean geometry, a plane in 3D space can be determined by three points, and to do projection an origin $r_o = (ox, oy, oz)$ and two coordinate axes defined by normalized vectors $\boldsymbol{e}_1 = (ex_1, ey_1, ez_1)$ and $\boldsymbol{e}_2 = (ex_2, ey_2, ez_2)$, should be selected. Given a 3D point $s = (sx, sy, sz)$, its projection on the 2D plane is $p = (p_1, p_2)$ where $p_1 = \boldsymbol{e}_1 \cdot (s - r_o)$ and $p_2 = \boldsymbol{e}_2 \cdot (s - r_o)$. A multiplication scaling factor is used for the projected skeletons matching to the practical data.

5.5 Experiment Results

We evaluate the proposed system on PlasmaTherm and E-Beam to demonstrate adaptive learning capability in ICPHS. To deliver more convincing results, every evaluation result below is averaged over 20 trainings with different seeds.

5.5.1 PlasmaTherm with PLC

The machine interface of PlasmaTherm is a keyboard with monitor. Machine operations are done through interacting with the keyboard. We randomly select 766 samples from NTU dataset class 30 (typing keyboard) as positive (interaction) samples and 766 samples from class 08, 12, 29, 33, 68, 96 as negative samples, and name it Pre-NTU dataset. Pre-NTU dataset is then preprocessed by the rotation and projection method. To provide validation results and achieve a fair evaluation, we build a test benchmark dataset for PlasmaTherm with 1346 video clips (693 positive and 653 negative samples) collected from the realistic viewpoint. The test set includes various interaction samples (keyboard operations, and special pump operations to be addressed below) and non-interaction samples (walk around, sit, stand, inspect, take notes, log in another computer, check cell phone) performed by 2 people. Each video in test set is 3 second and resampled to 10 fps. OpenPose is applied locally on the test set to extract 2D skeletons. The evaluations with performance metric of precision, recall, F1 score, and accuracy are done on the benchmark test set. During deployment, the video streaming rate is set as 10 fps. The response time for different machine state transition and the interaction duration are derived from the average of several test measurements.

We first compare the initialization performance pretrained with Pre-NTU before and after the rotation and projection as illustrated in Table 5.2. Suffix p indicates preprocessing applied and the first column (*i.e.*, 766, 100, ...) represents the data amount per class. The viewing plane for projection is selected based on the actual camera placement where the intersects with -z axis, +y axis, and -x axis are 45°, 25°, and 65° respectively. The pretraining data without preprocessing are extracted from color-XY attribute in NTU dataset. Before and after rotation and projection, the accuracy improves by 17.5%, which demonstrates the effectiveness of the preprocessing technique. In addition, we vary the number of Pre-NTU data to learn how it affects the initial detection accuracy since some applications can have initial accuracy needs, and different number of pretraining data can be chosen accordingly.

As noted in Table 5.2, more public pretraining data do not always lead to better results since the model tends to overfit on the pretraining data distribution resulting in a worse model generalization.

From Apr. 8, 2021 to May 21, 2021, the system was deployed on PlasmaTherm to automatically collect and label positive and negative samples. In total 139 self-labeled positive samples are collected. Among them, 58 samples are collected because of pump operations, 5 are related to heater operations, and the rest 76 are due to RF operations. For evaluation purpose, we manually inspected the self-labeled samples and found that 23 out of the 76 RF samples were labeled at a wrong timing due to variations of the on-set RF operation response time. As explained in Section 5.3, the response time variation can cause the self-labeling mechanism failure in capturing data at a deviated timing. With the proposed automated post-processing filter, 22 mis-labeled samples are filtered out (117 positive samples left) since no person in the scene or people out of ROI. The label error rate for positive samples is reduced to 0.85%. Correspondingly, the same number of negative samples (no mis-label) are randomly selected from self-labeled negative collection automatically.

Dataset	Precision	Recall	F1 score	Acc
766	0.648	0.887	0.738	67.1%
766p	0.829	0.910	0.860	84.6%
100p	0.843	0.912	0.869	85.7%
200p	0.865	0.926	0.889	88.1%
300p	0.852	0.952	0.897	88.7%
400p	0.840	0.965	0.895	88.1%
500p	0.834	0.900	0.858	84.9%

Table 5.2: Pre-NTU pretraining results in PlasmaTherm case

To demonstrate the adaptive learning capability, the self-labeled samples are grouped from Apr. 8 to Apr. 22 (39 samples per class), Apr. 8 to May 6 (78 per class), and Apr. 8 to May 21 (117 per class) in order to evaluate the adaptability evolution with more data feeding in and model retrained. We retrained the model based on the one pretrained on 100 samples as it had achieved a relatively good initial accuracy with less possible overfitting. Table 5.3 lists

Method	Dataset	Precision	Recall	F1 score	Acc
Ours	100p (initial)	0.843	0.912	0.869	85.7%
Ours	100p+39 (04/22)	0.946	0.980	0.962	96.0%
Ours	100p+78 (05/06)	0.956	0.980	0.968	96.6%
Ours	100p+117 (05/21)	**0.981**	**0.985**	**0.983**	**98.2%**
Ours	100p+139 (unfiltered)	0.959	0.967	0.962	96.1%
Ours	766p+117	0.886	0.971	0.925	91.8%
K-means	100p+117 (05/21)	0.592	1	0.744	64.5%
P&C [131]	100p+117 (05/21)	0.858	0.942	0.899	89.0%
CrosSCLR [82]	100p+117 (05/21)	0.879	0.964	0.919	91.3%

Table 5.3: Adaptive Learning Results for PlasmaTherm case

the evaluation results after adaptive retraining. With more self-labeled data, the accuracy improves gradually. With full 117 self-labeled data, the detection accuracy is 12.5% higher than the initial, and it is also 9.5% higher than the highest pretraining accuracy. This demonstrates that the proposed adaptive learning mechanism can improve its performance and is better than using more public dataset to pretrain. A retraining experiment is also run on the full 766 pretraining dataset with the 117 self-labeled data. It shows a relatively worse performance with a 6.4% accuracy degradation compared to the 100 pretraining case, but shows an accuracy improvement of 7.2% over the initial. This is attributed to the ratio between the amount of pretraining data and that of self-labeled ones. Even though with the rotation and projection preprocessing, the data similarity between public and practical data is improved and consequently the accuracy improves, the public dataset is still less effective than the self-labeled data. More pretraining data with different data distribution tend to slow down the convergence of adaptive learning. In addition, an experiment to retrain the model with unfiltered self-labeled samples is conducted. With noisy labels, the adaptive learning can work to improve accuracy by 10.4%, which is 2.1% worse than the retraining case with cleaned data.

Furthermore, we compare our method with recent unsupervised methods as shown in Table 5.3 since our solution does not require manual labeling. While [131] and [82] are unsupervised, they require a simple supervised classifier on top of their unsupervised representation

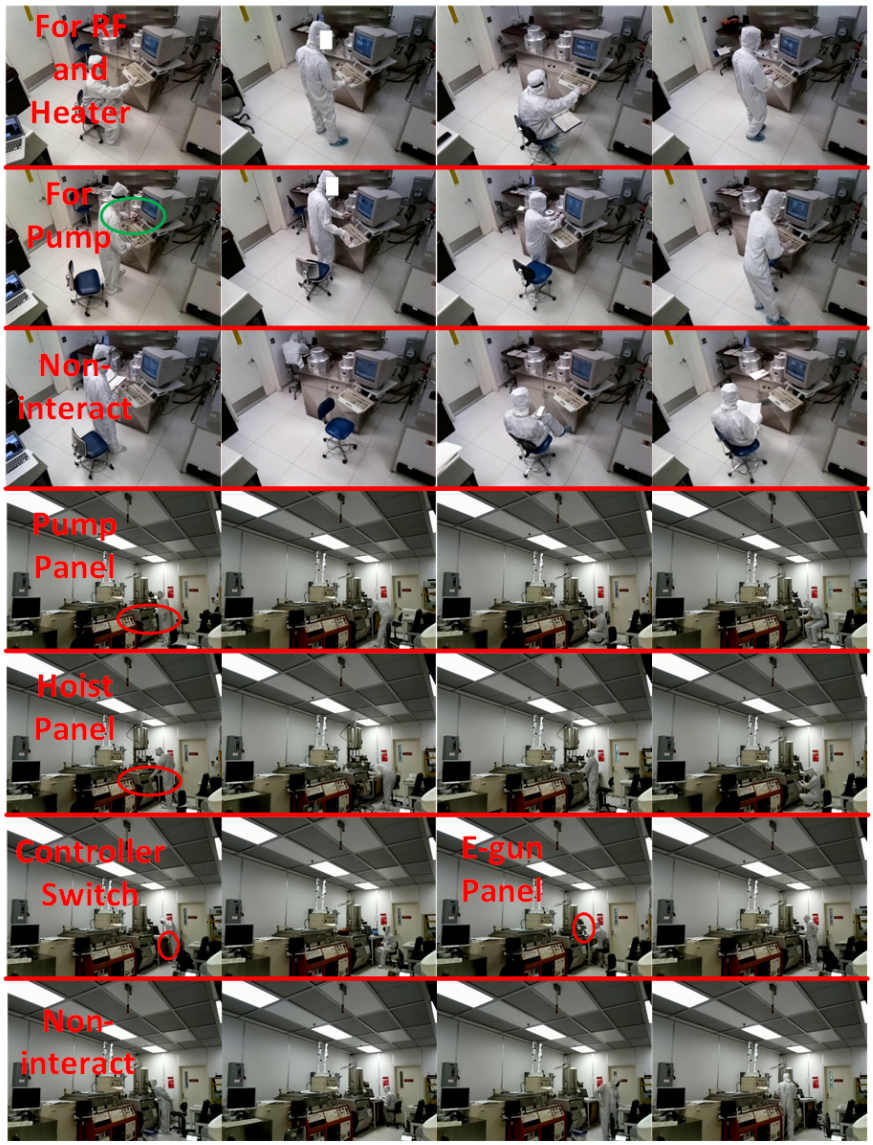

Figure 5.4: Example images of actions towards PlasmaTherm (row 1-3) and E-Beam (row 4-7) by different users captured and labeled through the self-labeling mechanism. Each row's action type is marked in red text where row 6 is split for two action types. Red circles indicate the E-Beam panel/switch locations. Green circle highlights the pump push-down action.

learning to classify actions. Our method also outperforms them because of the self-labeling supervision.

Beyond the above results, we found that even though all the typing keyboard actions are categorized to be the same interaction class, there is minor difference between the action to

Overall Acc	Pump Precision	Keyboard Precision	Non-interaction Precision
91.4%	0.863	0.960	0.959

Table 5.4: Three-class model performance metric

start pump and RF/heater respectively on PlasmaTherm as illustrated in Fig. 5.4. Users tend to put their left hand on the chamber handle when triggering the pump action because the chamber needs extra force to be closed tightly, which is also required by SOP. With the machine state transition determined, this feature can be differentiated and labeled, revealing the potential to reach more fine-grained action recognition. An experiment is conducted to demonstrate this potential. Among the cleaned 117 self-labeled positive samples, 58 samples belong to pump operation and 59 samples (with 1 wrong label) belong to RF/heater operation. 58 non-interaction samples are randomly selected to build a retraining dataset with 3 classes. The 3-class model is trained based on a model trained on 100p+117 data with the FC layer replaced. The test set used above includes 199 pump operation samples, and we further collected 189 pump-related samples (stand/sit to push down, use one or two hands to push down) to build a more comprehensive test set (388 pump-related interactions, 494 keyboard interactions, and 653 non-interactions) to evaluate the 3-class model. Table 5.4 lists the overall accuracy with class-level precision, illustrating that the proposed adaptive learning mechanism can achieve class incremental learning with self-labeled samples to classify interaction types related to different machine components without extra labeling.

Furthermore, the worker action recognition is built with a cascaded ML model. The first step outputs, 2D skeletons, are noisy due to missing joints, wrong joint locations and jitters. Noisy data is common in practical skeleton-based action recognition but most research in this field focuses on training with intact skeletons. The first-step ML model can be replaced by other 2D or 3D pose estimation applications, such as Microsoft Azure Kinect, but the noisy data issue can persist. The proposed adaptive learning method uses noisy skeletons after score-based naive filtering to adaptively retrain the GCN. Since in practical industrial applications,

camera locations and surroundings do not change significantly, the pose estimation errors tend to be consistent. For example, in this study parts of the left arm are missing in some frames because of occlusion. By injecting noisy data for training, GCN can adapt to the noisy data and generate right results. It is significant because our approach naturally embeds the noisy data training into the adaptive process targeting this practical issue.

5.5.2 E-Beam Without PLC

After the successful demonstration on a PLC controlled machine, next we show the results of a manually operated E-Beam machine. The machine interfaces of the four E-Beam functional components involve multiple control panels and switches located at different positions as shown in Fig. 5.4. A manual machine typically requires worker interactions using various interfaces at different locations, complicating the interaction recognition. Moreover, workers tend to interact with the interfaces with other still gestures, *i.e.*, stand, sit, squat or bend, and interactions driven by hands need to be recognized with these various still gestures. From NTU dataset, 500 samples are randomly selected from class 69, 70 as positive samples, and 500 samples from class 8, 9, 35, 92, 96 are used as negative. To adapt to the actual viewing angle, the projection plane is chosen as 10° and 80° for intersects with +y axis and +x axis, respectively, and parallel to z axis. We collect 260 interaction samples (with all the interfaces in different still gestures) and 260 non-interaction samples (sit, stand, manipulate materials, clean chambers, watch chamber through windows, and operate other adjacent machines) performed by 1 worker as test set for evaluation. The video streaming rate is 6 fps. Each sample in the test set is 5 second and resampled to 6 fps. The response time for E-Beam state transition and the interaction duration are derived similarly from mean values.

The adaptive learning system is deployed from Apr. 8, 2021 to May 21, 2021 to collect and self-label data. There are 211 positive samples (45 related to pump, 48 related to E-gun,

Dataset	Precision	Recall	F1 score	Acc
500	0.560	0.866	0.656	57.5%
500p	0.800	0.699	0.736	75.2%
500p+141 (05/21)	0.893	0.802	0.843	85.1%

Table 5.5: Adaptive Learning Results for E-Beam

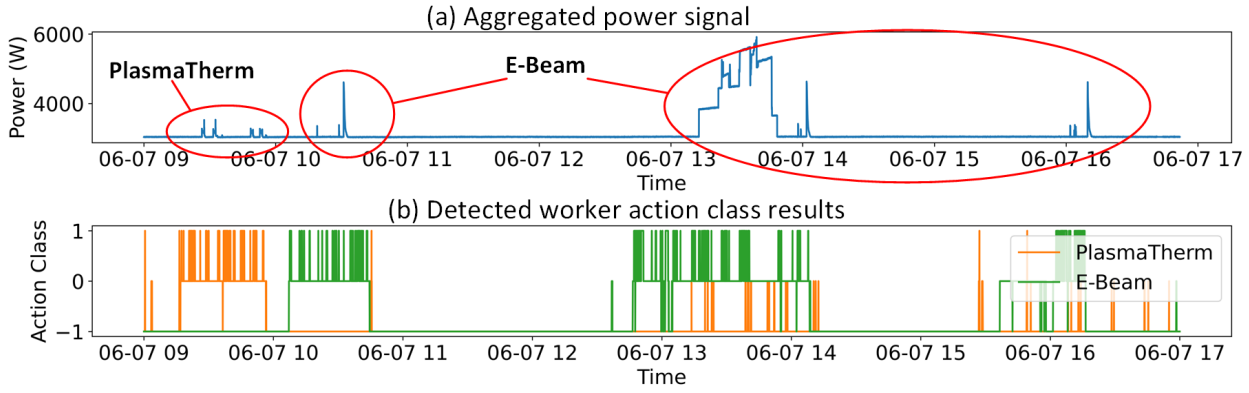

Figure 5.5: In (a), an aggregated power signal of PlasmaTherm and E-Beam including 7 components is given. Power signals for active machines are labeled in red circles. In (b), the action classification results are given for the two machines. 1 means interaction, 0 means non-interaction, and -1 indicates no worker in the scene.

75 related to hoist, and 43 related to controller) collected and labeled. Among them, 16 samples are mis-labeled due to the response time variation. In addition, a lot of E-Beam usages (67 samples including 2 mis-labeled) involve tool training of a user by staff in the scene. By applying the post-processing filter, 70 samples are automatically eliminated from the self-labeled data, where 11 improper samples cannot be removed due to people overlap (7 samples) or ROI selection (4 samples) in our monocular camera setup. After filtering, there are 141 self-labeled positive samples left and correspondingly 141 self-labeled negative samples (no mis-label) are randomly selected. The label noise level for self-labeled positive samples is 7.8%.

Table 5.5 lists the evaluation results on E-Beam. Comparing the performance before and after the rotation and projection, the accuracy significantly improves. With self-labeled samples to retrain the model, the model accuracy improves by 9.9%. This result further validates the feasibility of the proposed adaptive learning framework. It is worthwhile noting that

the accuracy improvement in PlasmaTherm case is better than in E-Beam. Several reasons could explain the difference. Firstly, the noise-level after post-processing filter in E-Beam case is much higher than that in PlasmaTherm, and most of the noisy data come from two-people overlap scenario. One way to improve is to apply camera triangulation to identify overlap issues. Moreover, it is noted before post-processing filter that the mis-labeled ratio of E-Beam case is less than that of PlasmaTherm case. This is because the response time variation of manual machines comes from hardware circuitry response while that of PLC machines is determined by both hardware and software. PLC software is designed to define a stabilization period for process parameter check, introducing more variation on response time than hardware circuit. It is of interest to note that the variation in response time can form a new class of learning for determining PLC machines' functional components operation conditions over time as a diagnostic tool. Secondly, E-Beam interaction interfaces are more diverse and complex than PlasmaTherm, making it a more challenging task. This can be attributed to the intrinsic differences between PLC machines and non-PLC machines. The third reason is that the pretraining accuracy baseline of E-Beam is not as high as that of PlasmaTherm, suggesting a requirement for a manual operated machine to collect more self-labeled data to reach comparable accuracy.

As we have successfully demonstrated the downward branch of labeling worker actions using machine states, worker actions can be used as the context of machine state transition in a multi-machine environment to assist energy disaggregation, which is the upward branch in ICPHS phase 2. Fig. 5.5 illustrates a realistic example based on the experimental results. We can clearly observe dense worker interactions for specific machine during its state transition, which demonstrates the potential to achieve reverse self-labeling.

5.6 Discussion

Compared with adaptive ML studies relying on concept drift detection, this study leverages the unchanged causality in different domains. As argued by Schölkopf [121], the inner causality can stay unchanged, while data distributions can vary among different domains for a ML task. Specifically, there are several performance benefits in the proposed system compared with conventional supervised ML of action recognition and methods of concept drift detectors. They are: 1) the adaptability ensures successful interaction recognition for unforeseen cases such as new workers and new machine interfaces; 2) while some real-world drifts are hard to predict, the unchanged causality provides a more reliable way to automatically annotate data; 3) the model adaptation and class increment are achieved simultaneously using the same system.

As the ICPHS being successfully demonstrated, several potential applications for workers and machines are proposed.

Application 1: A real time ML model for worker action recognition can be explored to prevent human operation mistakes, such as missing or incorrect operations. Moreover, the system can be utilized as a virtual supervisor during new worker hands-on training to reinforce new worker's learning.

Application 2: A real time ML model for machine activity recognition can investigate machine components' power variation over time, which can be used as a machine prognosis tool for identifying issues such as pumping speed decrease due to decaying pump efficiency. Furthermore, HMI detection offers opportunities to improve machine energy efficiency by introducing sleep mode operation during machine idling.

Moreover, the adaptive learning method enables model training at the edge by relaxing requirements of data storage. The static training by large datasets can be replaced by a

gradually self-labeled dataset of phase 2 at edge platforms. Additionally, this study devises a novel method to self-label data using causal relationships, which is not constrained by specific ML models. For instance, extra cameras can be installed to focus on specific regions of control panels to recognize the operated buttons and hand/finger movements by other ML models. The proposed method can work in this case to dynamically self-label data and adapt the ML models.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation targets at a critical problem that slows down the spreading of AI in many traditional fields. The problem is how to reduce the cost of applying AI in terms of manual data collection and annotation and how to let deployed AI models adapt to the local data distribution shifts autonomously. This problem significantly raises the barrier of AI adoption and constrains AI's popularity in many traditional industries. Different from conventional algorithmic methods to improve ML adaptability, this dissertation transforms its angle to a system level and first proposes to use the interactive causality and its learnable causal time lag as a means to automatically associate real-time data streams for adapting ML models to local data distributions. This dissertation presents and studies the new idea in a comprehensive way from theory, methodology, to real-world application. Dynamical system theory is utilized to prove that the proposed interactive causality based self-labeling method is more robust on data shifts compared to traditional semi-supervised learning based on pseudo labels generated by trained models. Theories regarding complex causal structure

for self-labeling are proposed and certain simulations are provided to demonstrate the effectiveness. From methodology perspective, a comprehensive framework utilizing the concept of ontology and knowledge graph is presented to complete the questions of where causality can be found and modeled, how to select interaction and observing channels, and how to proceed with self-labeling, and how to gain new knowledge and revise knowledge graphs. It emulates how humans build up new knowledge for a domain. A real-world study in semiconductor manufacturing has shown superiority of the proposed self-labeling to recognize worker machine interaction adaptively. Overall, we believe the proposed idea opens a new chapter in the area of adaptive AI and can be referenced by other researchers to explore deeply in this direction.

6.2 Future Work

The future directions of this interactive causality based adaptive learning can be summarized into three aspects: theory, methodology, and application. From the theoretical perspective, this dissertation only provides a strict proof in 1-d case with defined systems of differential equations. A future direction is to extend the proof to n-d with more complex and general interaction conditions. For n-d dynamical systems, each DS is internally coupled among its dimensions. Two interacting DS are also coupled along certain dimensions as interaction conditions. The interactions of n-d DS are more complicated to be studied in self-labeling scenarios. Currently, the derivation needs an integration first to solve the system of differential equations and then compare the relative relationships of the four methods (forward and backward self-labeling, fully supervised, and semi-supervised.) A future direction is to utilize some mathematical tools to bypass the integration step as integrals of n-d DS is challenging. Additionally, a more comprehensive comparison between self-labeling and semi-supervised learning can be accomplished. Another interesting topic to be explored is to theoretically

quantify the impact of inaccurate interaction time inference and inaccurate effect state detector on the model learning. This quantification will need to be integrated with generic ML learning theory and be used as a reference for the design of interaction time model and effect state detector.

For methodology, an influential aspect for the future is the exploration of the usage of knowledge graph in new knowledge acquisition. The new knowledge can represent found anomalies during deployment. This dissertation provides one way to do so by combining existing unsupervised pattern recognition and causal discovery methods. A future idea in this topic to mimic how humans gain new knowledge by utilizing existing knowledge to ask questions. As we grow up, this is always the fundamental way we learn the world via interaction. Therefore, it is expected to develop a method of expanding knowledge graph by asking questions. Using ChatGPT as an example, most of the time human users ask questions to GPT and expect for answers. Very limited times ChatGPT will ask clarifications to human users. Thus asking questions is an effective means to gain knowledge. A pathway has been conceived based on the interactive causality methodology to expand KG. Initially, we still expect humans with different roles in different application scenarios, such as technicians in manufacturing scenarios, to ask questions to the domain KG based on users experience and knowledge. An algorithm with a graph search engine can be designed to explore the existing KG for related nodes based on users questions. These found nodes can be utilized by users to develop solutions for the asked questions. The developed solution can potentially introduce new nodes represented by new data streams that can be added to the KG and used for future self-labeling purpose. More advanced, this entire pathway can be automated and initiated by AI asking questions.

In terms of applications, the proposed self-labeling has great potential in traditional fields with limited datasets and AI expertise such as smart manufacturing and precision agriculture. The multimodal nature of cyber physical systems (CPS) paves the way to explore

pervasive applications of self-labeling. The rich multi-modal signals in CPS can be utilized for obtaining additional observing channels. CPS involve various interactions among the elements inside where causation can easily be found as additional observing channels. Many manufacturing processes, such as welding and assembly, can be enhanced with adaptive AI perception system by extracting and modeling the causation among these processes. For example, an adaptive AI system can be designed to recognize the interactions occurred during manual welding. The interactions involve how hands interact with welding guns (machines) and filling rods (materials) and how welders (power level, temperature) interact with materials. Self-labeling can be applied on these interactions to enhance AI perception to avoid human errors. In assembly, the interaction among operators, robots, and parts can be studied for self-labeling to achieve more efficient human robot collaboration with AI-enhanced intention recognition capability.

Another field with great potential is in autonomous driving. The self-labeling technique can be applied to enhance driving intention recognition with adaptability for many unseen situations. For example, certain vehicle behaviors can be self-labeled as intentions (causes) for other perceivable effects. In addition, as this self-labeling requires data streams and known causal graphs, there is a need to establish a standard metric and benchmark datasets shared with the community. The required dataset is different from conventional static datasets such as image recognition benchmarks and currently there is no such a public dataset available. Therefore it is expected that a benchmark from a typical application scenario in CPS can be established for examining various self-labeling algorithm development.

Additionally, for many of these CPS applications, a key is to model the domain knowledge and extract causal graphs of interactive events used for self-labeling, which can potentially be facilitated by large language models to summarize causality from inputs of natural languages.

Bibliography

[1] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft. Universal digital twin - a dynamic knowledge graph. *Data-Centric Engineering*, 2:e14, 2021.

[3] A. Al-Shdifat and C. Emmanouilidis. Development of a context-aware framework for the integration of internet of things and cloud computing for remote monitoring services. *Procedia Manuf*, 16:31–38, 2018.

[4] K. Alexopoulos, S. Makris, V. Xanthakis, K. Sipsas, and G. Chryssolouris. A concept for context-aware computing in manufacturing: the white goods case. *Int J Comput Integr Manuf*, 29(8):839–849, 2016.

[5] K. Alexopoulos, K. Sipsas, E. Xanthakis, S. Makris, and D. Mourtzis. An industrial internet of things based platform for context-aware information services in manufacturing. *Int J Comput Integr Manuf*, 31(11):1111–1123, 2018.

[6] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. Moura. Event detection for non intrusive load monitoring. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pages 3312–3317, 2012.

[7] B. R. Andrus, Y. Nasiri, S. Cui, B. Cullen, and N. Fulda. Enhanced story comprehension for large language models through dynamic document-based knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10436–10444, Jun. 2022.

[8] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] A. Arbabi, D. R. Adams, S. Fidler, and M. Brudno. Identifying clinical terms in medical text using ontology-guided machine learning. *JMIR Med Inform*, 7(2):e12596, May 2019.

[10] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021.

[11] Y. M. Asano, C. Rupprecht, and A. Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[12] N. Azouz and H. Pierreval. Adaptive smart card-based pull control systems in context-aware manufacturing systems: Training a neural network through multi-objective simulation optimization. *Appl Soft Comput*, 75:46–57, 2019.

[13] R. Bakirov and B. Gabrys. Investigation of expert addition criteria for dynamically changing online ensemble classifiers with multiple adaptive mechanisms. In *Int. Conf. Artif. Intell. Appl. Innov.*, pages 646–656, 2013.

[14] N. Batra, R. Kukunuri, A. Pandey, R. Malakar, R. Kumar, O. Krystalakos, M. Zhong, P. Meira, and O. Parson. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, page 193–202, 2019.

[15] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[16] L. Bombelli, J. Lee, D. Meyer, and R. D. Sorkin. Space-time as a causal set. *Phys. Rev. Lett.*, 59:521–524, Aug 1987.

[17] B. Bozorgtabar and D. Mahapatra. Attention-conditioned augmentations for self-supervised anomaly detection and localization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14720–14728, Jun. 2023.

[18] N. R. Bramley, T. Gerstenberg, R. Mayrhofer, and D. A. Lagnado. Time in causal structure learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 44(12):1880, 2018.

[19] N. Brännström. Averaging in weakly coupled discrete dynamical systems. *Journal of Nonlinear Mathematical Physics*, 16(4):465–487, 2009.

[20] S. L. Bressler and A. K. Seth. Wiener–granger causality: A well established methodology. *NeuroImage*, 58(2):323–329, 2011.

[21] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, July 2017.

[22] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[23] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. 2023.

[24] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss. Automatic labeling to generate training data for online lidar-based moving object segmentation. *IEEE Robotics and Automation Letters*, 7(3):6107–6114, 2022.

[25] C.-Y. Cheng. A novel approach of information visualization for machine operation states in industrial 4.0. *Comput Ind Eng*, 125:563 – 573, 2018.

[26] C. Cimini, F. Pirola, R. Pinto, and S. Cavalieri. A human-in-the-loop manufacturing control architecture for the next generation of production systems. *J Manuf Syst*, 54:258–271, 2020.

[27] V. de. Learning classification with unlabeled data. In *Adv. Neural Inf. Process. Syst.*, volume 6, 1994.

[28] F. Demrozi, M. Jereghi, and G. Pravadelli. Towards the automatic data annotation for human activity recognition based on wearables and ble beacons. In *2021 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, pages 1–4, 2021.

[29] S. Deng, H. Rangwala, and Y. Ning. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1585–1595, New York, NY, USA, 2020. Association for Computing Machinery.

[30] A. M. Deshpande, A. K. Telikicherla, V. Jakkali, D. A. Wickelhaus, M. Kumar, and S. Anand. Computer vision toolkit for non-invasive monitoring of factory floor artifacts. *Procedia Manuf*, 48:1020–1028, 2020.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[33] R. Drake, M. Yildirim, J. Twomey, L. Whitman, J. Sheikh-Ahmad, and P. Lodhia. Data collection framework on energy consumption in manufacturing. *Proceedings of 2006 Institute of Industrial Engineering Research Conference*, 01 2006.

[34] S. Du, G. Song, L. Han, and H. Hong. Temporal causal inference with time lag. *Neural Computation*, 30(1):271–291, 2018.

[35] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proc. 31st Conf. Uncertain. Artif. Intell.*, page 258–267, 2015.

[36] B. Edrington, B. Zhao, A. Hansel, M. Mori, and M. Fujishima. Machine monitoring system based on mtconnect technology. *Procedia CIRP*, 22:92–97, 2014.

[37] T. Elsaleh, S. Enshaeifar, R. Rezvani, S. T. Acton, V. Janeiko, and M. Bermudez-Edo. Iot-stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services. *Sensors*, 20(4), 2020.

[38] C. Emmanouilidis, P. Pistofidis, A. Fournaris, M. Bevilacqua, I. Durazo-Cardenas, P. N. Botsaris, V. Katsouros, C. Koulamas, and A. G. Starr. Context-based and human-centred information fusion in diagnostics. *IFAC-PapersOnLine*, 49(28):220–225, 2016.

[39] J. Feydy. *Geometric data analysis, beyond convolutions*. PhD thesis, Université Paris-Saclay Gif-sur-Yvette, France, 2020.

[40] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In A. L. C. Bazzan and S. Labidi, editors, *Advances in Artificial Intelligence – SBIA 2004*, pages 286–295, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[41] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), Mar. 2014.

[42] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano, K. Kim, E. Wang, M. Lingelbach, A. Curtis, K. Feigelis, D. M. Bear, D. Gutfreund, D. Cox, A. Torralba, J. J. DiCarlo, J. B. Tenenbaum, J. H. McDermott, and D. L. K. Yamins. Threedworld: A platform for interactive multi-modal physical simulation, 2020.

[43] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[44] A. Ghassami, N. Kiyavash, B. Huang, and K. Zhang. Multi-domain causal structure learning in linear systems. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[45] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Int. Conf. Learn. Represent.*, 2018.

[46] M. Gil, M. Albert, J. Fons, and V. Pelechano. Engineering human-in-the-loop interactions in cyber-physical systems. *Inf. Softw. Technol.*, 126:106349, 2020.

[47] I. Goldenberg and G. I. Webb. Survey of distance measures for quantifying concept drift and shift in numeric data. *Knowledge and Information Systems*, 60(2):591–615, Aug 2019.

[48] H. F. Gollob and C. S. Reichardt. Taking account of time lags in causal models. *Child Development*, 58(1):80–92, 1987.

[49] H. M. Gomes, M. Grzenda, R. Mello, J. Read, M. H. Le Nguyen, and A. Bifet. A survey on semi-supervised learning for delayed partially labelled data streams. *ACM Comput. Surv.*, feb 2022.

[50] M. Gong. Bridging causality and learning: How do they benefit from each other? In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5150–5153, 7 2020.

[51] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf. Domain adaptation with conditional transferable components. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 2839–2848, 20–22 Jun 2016.

[52] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[53] M. Grzenda, H. M. Gomes, and A. Bifet. Delayed labelling evaluation for data streams. *Data Mining and Knowledge Discovery*, 34(5):1237–1266, Sep 2020.

[54] M. Grzenda, H. M. Gomes, and A. Bifet. Performance measures for evolving predictions under delayed labelling classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[55] Y. Guo, Y. Sun, and K. Wu. Research and development of monitoring system and data monitoring system and data acquisition of cnc machine tool in intelligent manufacturing. *Int J Adv Robot Syst*, 17(2), 2020.

[56] S. Gupta, P. A. Martinek, W. Schwarting, J. S. Hardy, and B. W. Mairs. Automatic labeling and learning of driver yield intention, Sept. 13 2016. US Patent 9,443,153.

[57] S. Han, N. Mannan, D. C. Stein, K. R. Pattipati, and G. M. Bollas. Classification and regression models of audio and vibration signals for machine state monitoring in precision machining systems. *J Manuf Syst*, 61:45–53, 2021.

[58] J. Hastings, M. Glauer, A. Memariani, F. Neuhaus, and T. Mossakowski. Learning chemistry: exploring the suitability of machine learning for the task of structure-based chemical ontology classification. *Journal of Cheminformatics*, 13:1–20, 2021.

[59] L. Hattam and D. V. Greetham. Energy disaggregation for smes using recurrence quantification analysis. In *Proceedings of the Ninth International Conference on Future Energy Systems*, page 610–617, 2018.

[60] P. Hitzler. A review of the semantic web field. *Commun. ACM*, 64(2):76–83, jan 2021.

[61] L. Horváth. Contextual knowledge content driving for model of cyber physical system. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1845–1850, 2018.

[62] S. S. Hosseini, K. Agbossou, S. Kelouwani, and A. Cardenas. Non-intrusive load monitoring through home energy management systems: A comprehensive review. *Renew Sust Energ Rev*, 79:1266–1274, 2017.

[63] L. Hu, H. Zheng, L. Shu, S. Jia, W. Cai, and K. Xu. An investigation into the method of energy monitoring and reduction for machining systems. *J Manuf Syst*, 57:390–399, 2020.

[64] A. Jaber, M. Kocaoglu, K. Shanmugam, and E. Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[65] A. Jaber, M. Kocaoglu, K. Shanmugam, and E. Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020.

[66] S. M. Jameel, M. A. Hashmani, H. Alhussain, and A. Budiman. A fully adaptive image classification approach for industrial revolution 4.0. In *Int. Conf. Reliab. Inf. Commun. Technol.*, pages 311–321, 2018.

[67] S. M. Jameel, M. A. Hashmani, M. Rehman, and A. Budiman. An adaptive deep learning framework for dynamic image classification in the internet of things environment. *Sensors*, 20(20), 2020.

[68] J. Jiao, F. Zhou, N. Z. Gebraeel, and V. Duffy. Towards augmenting cyber-physical-human collaborative cognition for human-automation interaction in complex manufacturing and operational environments. *Int. J. Prod. Res.*, 58(16):5089–5111, 2020.

[69] Y. Jin, E. Tebekaemi, M. Berges, and L. Soibelman. Robust adaptive event detection in non-intrusive load monitoring for energy aware smart facilities. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4340–4343, 2011.

[70] E. Kaasinen, S. Aromaa, P. Heikkilä, and M. Liinasuo. Empowering and engaging solutions for operator 4.0 – acceptance and foreseen impacts by factory workers. In F. Ameri, K. E. Stecke, G. von Cieminski, and D. Kiritsis, editors, *Advances in Production Management Systems. Production Management for the Factory of the Future*, pages 615–623, Cham, 2019. Springer International Publishing.

[71] E. Kaasinen, F. Schmalfuß, C. Özturk, S. Aromaa, M. Boubekeur, J. Heilala, P. Heikkilä, T. Kuula, M. Liinasuo, S. Mach, R. Mehta, E. Petäjä, and T. Walter. Empowering and engaging industrial workers with operator 4.0 solutions. *Comput Ind Eng*, 139:105678, 2020.

[72] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[73] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[74] D. J. Kedziora, K. Musial, and B. Gabrys. Autonoml: Towards an integrated framework for autonomous machine learning, 2020. arXiv:2012.12600.

[75] M. Kocaoglu, A. Jaber, K. Shanmugam, and E. Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[76] M. Kocaoglu, S. Shakkottai, A. G. Dimakis, C. Caramanis, and S. Vishwanath. Applications of common entropy for causal inference. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[77] L. Kocarev and U. Parlitz. Generalized synchronization, predictability, and equivalence of unidirectionally coupled dynamical systems. *Phys. Rev. Lett.*, 76:1816–1819, Mar 1996.

[78] T. D. Le, T. Hoang, J. Li, L. Liu, H. Liu, and S. Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(5):1483–1495, 2019.

[79] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[80] H.-C. Lee. Review of inductively coupled plasmas: Nano-applications and bistable hysteresis physics. *Appl Phys Rev*, 5(1):011108, Mar. 2018.

[81] J. Leng, P. Jiang, C. Liu, and C. Wang. Contextual self-organizing of manufacturing process for mass individualization: a cyber-physical-social system approach. *Enterp Inf Syst*, 14(8):1124–1149, 2020.

[82] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang. 3d human action representation learning via cross-view consistency pursuit. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2021.

[83] Z. Li, G. Zheng, A. Agarwal, L. Xue, and T. Lauvaux. Discovery of causal time intervals. In *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, pages 804–812, 2017.

[84] P. A. Lindahl, M. T. Ali, P. Armstrong, A. Aboulian, J. Donnal, L. Norford, and S. B. Leeb. Nonintrusive load monitoring of variable speed drive cooling systems. *IEEE Access*, 8:211451–211463, 2020.

[85] E. Lindgren, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath. Experimental design for cost-aware learning of causal graphs. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[86] E. Lindgren, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath. Experimental design for cost-aware learning of causal graphs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[87] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(10):2684–2701, 2020.

[88] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2020.

[89] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[90] B. Lu and V. C. Gungor. Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks. *IEEE Trans Ind Electron*, 56(11):4651–4659, 2009.

[91] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019.

[92] Y. Lu, X. Xu, and L. Wang. Smart manufacturing process and system automation – a critical review of the standards and envisioned scenarios. *J Manuf Syst*, 56:312–325, 2020.

[93] A. C. J. Luo. *Dynamical System Interactions*, pages 623–683. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[94] A. Madni, M. Sievers, and C. Madni. Adaptive cyber-physical-human systems: Exploiting cognitive modeling and machine learning in the control loop. *INSIGHT*, 21:87–93, 2018.

[95] S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[96] T. Mar, V. Tikhanoff, G. Metta, and L. Natale. Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot. In *2015 IEEE Int. Conf. Robot Autom.*, pages 3200–3206.

[97] P. B. M. Martins, J. G. R. C. Gomes, V. B. Nascimento, and A. R. de Freitas. Application of a deep learning generative model to load disaggregation for industrial machinery

power consumption monitoring. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, 2018.

[98] A. Merkhouf and M. I. Boulos. Integrated model for the radio frequency induction plasma torch and power supply system. *Plasma Sources Sci Technol*, 7(4):599–606, nov 1998.

[99] C. Mesterharm. On-line learning with delayed label feedback. In *Algorithmic Learning Theory*, pages 399–413, 2005.

[100] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, page 1003–1011, 2009.

[101] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[102] F. Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5):183–197, 1991.

[103] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *Proc. Eur. Conf. Comput. Vis.*, pages 801–816, 2016.

[104] M. Paluš, A. Krakovská, J. Jakubík, and M. Chvosteková. Causality, dynamical systems and the arrow of time. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075307, 2018.

[105] N. Panten, E. Abele, and S. Schweig. A power disaggregation approach for fine-grained machine energy monitoring by system identification. *Procedia CIRP*, 48:325 – 330, 2016.

[106] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2536–2544, June 2016.

[107] J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2009.

[108] J. Plasse and N. Adams. Handling delayed labels in temporally evolving data streams. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2416–2424, 2016.

[109] G. J. Qi and J. Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2020.

[110] C. Qian, Y. Zhang, C. Jiang, S. Pan, and Y. Rong. A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing. *Robot Comput Integr Manuf*, 61:101841, 2020.

[111] A. Raj, G. Dwivedi, A. Sharma, A. B. Lopes de Sousa Jabbour, and S. Rajak. Barriers to the adoption of industry 4.0 technologies in the manufacturing sector: An inter-country comparative perspective. *Int. J. Prod. Econ.*, 224:107546, 2020.

[112] S. Rajbhandari, J. Aryal, J. Osborn, A. Lucieer, and R. Musk. Leveraging machine learning to extend ontology-driven geographic object-based image analysis (o-geobia): A case study in forest-type mapping. *Remote Sensing*, 11(5), 2019.

[113] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[114] A. U. Rehman, T. T. Lie, B. Vallès, and S. R. Tito. Event-detection algorithms for low sampling nonintrusive load monitoring systems based on low complexity statistical features. *IEEE Trans Instrum Meas*, 69(3):751–759, 2020.

[115] A. U. Rehman, S. Rahman Tito, T. T. Lie, P. Nieuwoudt, N. Pandey, D. Ahmed, and B. Vallès. Non-intrusive load monitoring: A computationally efficient hybrid event detection algorithm. In *2020 IEEE International Conference on Power and Energy (PECon)*, pages 304–308, 2020.

[116] Y. Ren and G.-P. Li. A contextual sensor system for non-intrusive machine status and energy monitoring. *J. Manuf. Syst.*, 62:87–101, 2022.

[117] Y. Ren and G.-P. Li. An interactive and adaptive learning cyber physical human system for manufacturing with a case study in worker machine interactions. *IEEE Transactions on Industrial Informatics*, 18(10):6723–6732, 2022.

[118] K. Ruan and X. Di. Learning human driving behaviors with sequential causal imitation learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4583–4592, Jun. 2022.

[119] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[120] B. Schölkopf. *Causality for Machine Learning*, page 765–804. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.

[121] B. Schölkopf. Causality for machine learning, 2019. arXiv:1911.10500.

[122] A. Sharma and E. Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

[123] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2019.

[124] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608, 2020.

[125] J. Sossenheimer, O. Vetter, E. Abele, and M. Weigold. Hybrid virtual energy metering points – a low-cost energy monitoring approach for production systems based on offline trained prediction models. *Procedia CIRP*, 93:1269–1274, 2020.

[126] J. Sossenheimer, T. Weber, D. Flum, N. Panten, E. Abele, and T. Fuertjes. *Non-intrusive Load Monitoring on Component Level of a Machine Tool Using a Kalman Filter-Based Disaggregation Approach*, pages 155–165. Springer International Publishing, Cham, 2019.

[127] V. M. Souza, D. F. Silva, G. E. Batista, and J. Gama. Classification of evolving data streams with infinitely delayed labels. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 214–219, 2015.

[128] T. Stankovski, A. Duggento, P. V. E. McClintock, and A. Stefanovska. Inference of time-evolving coupled dynamical systems in the presence of noise. *Phys. Rev. Lett.*, 109:024101, Jul 2012.

[129] T. Stankovski, T. Pereira, P. V. McClintock, and A. Stefanovska. Coupling functions: dynamical interaction mechanisms in the physical, biological and social sciences. *Philosophical Transactions of the Royal Society A*, 377(2160):20190039, 2019.

[130] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[131] K. Su, X. Liu, and E. Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2020.

[132] K. Suzuki, Y. Yokota, Y. Kanazawa, and T. Takebayashi. Online self-supervised learning for object picking: Detecting optimum grasping position using a metric learning approach. In *2020 IEEE/SICE Int. Symp. Syst. Integr.*, pages 205–212.

[133] Y. Svetashova, B. Zhou, T. Pychynski, S. Schmidt, Y. Sure-Vetter, R. Mikut, and E. Kharlamov. Ontology-enhanced machine learning: A bosch use case of welding quality monitoring. In J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, editors, *The Semantic Web – ISWC 2020*, pages 531–550, Cham, 2020. Springer International Publishing.

[134] R. Syed. Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. *Information & Management*, 57(6):103334, 2020.

[135] Y. S. Tan, Y. T. Ng, and J. S. C. Low. Internet-of-things enabled real-time monitoring of energy efficiency on manufacturing shop floors. *Procedia CIRP*, 61:376–381, 2017.

[136] F. Tao, Q. Qi, A. Liu, and A. Kusiak. Data-driven smart manufacturing. *J Manuf Syst*, 48:157–169, 2018.

[137] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, and J. Lloret. Context-aware cloud robotics for material handling in cognitive industrial internet of things. *IEEE Internet Things J*, 5(4):2272–2281, 2018.

[138] J. Wang, S. Gao, Z. Tang, D. Tan, B. Cao, and J. Fan. A context-aware recommendation system for improving manufacturing process modeling. *J Intell Manuf*, Oct 2021.

[139] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.*, 48:144–156, 2018.

[140] K.-J. Wang, Y.-H. Lee, and S. Angelica. Digital twin design for real-time monitoring – a case study of die cutting machine. *Int J Prod Res*, 0(0):1–15, 2020.

[141] P. Wang, H. Liu, L. Wang, and R. X. Gao. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. *CIRP Annals*, 67(1):17–20, 2018.

[142] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo, H. Qi, Z. Wu, Y.-F. Li, S. Nakamura, W. Ye, M. Savvides, B. Raj, T. Shinozaki, B. Schiele, J. Wang, X. Xie, and Y. Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[143] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, , Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. 2023.

[144] Z. Wang, M. Ritou, C. D. Cunha, and B. Furet. Contextual classification for smart machining based on unsupervised machine learning by gaussian mixture model. *Int J Comput Integr Manuf*, 33(10-11):1042–1054, 2020.

[145] Z. Wang, M. Ritou, C. D. Cunha, and B. Furet. Contextual classification for smart machining based on unsupervised machine learning by gaussian mixture model. *Int J Comput Integr Manuf*, 33(10-11):1042–1054, 2020.

[146] W. Wichakool, Z. Remscrim, U. A. Orji, and S. B. Leeb. Smart metering of variable power loads. *IEEE Trans Smart Grid*, 6(1):189–198, 2015.

[147] T. Wuest, C. Irgens, and K.-D. Thoben. An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *J Intell Manuf*, 25(5):1167–1180, Oct. 2014.

[148] Q. Xiong, J. Zhang, P. Wang, D. Liu, and R. X. Gao. Transferable two-stream convolutional neural network for human action recognition. *J Manuf Syst*, 56:605–614, 2020.

[149] R. Yadav, A. K. Pradhan, and I. Kamwa. Real-time multiple event detection and classification in power system using signal energy transformations. *IEEE Trans Industr Inform*, 15(3):1521–1531, 2019.

[150] M. Yahya, J. G. Breslin, and M. I. Ali. Semantic web and knowledge graphs for industry 4.0. *Applied Sciences*, 11(11), 2021.

[151] H. Yan, Y. Guo, and C. Yang. Augmented self-labeling for source-free unsupervised domain adaptation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[152] T. Yang, Q. Guo, L. Xu, and H. Sun. Dynamic pricing for integrated energy-traffic systems from a cyber-physical-human perspective. *Renew. Sust. Energ. Rev.*, 136:110419, 2021.

[153] J. Yu and X. Zhou. One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. *IEEE Trans. Ind. Informat.*, 16(10):6347–6358, 2020.

[154] B. Zhang, Y. Wang, W. Hou, H. WU, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, volume 34, pages 18408–18419, 2021.

[155] C. Zhang, K. Zhang, and Y. Li. A causal view on robustness of neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 289–301, 2020.

[156] K. Zhang, M. Gong, and B. Schoelkopf. Multi-source domain adaptation: A causal view. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015.

[157] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 819–827, 17–19 Jun 2013.

[158] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14471–14481, June 2022.

[159] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing. Dags with no tears: Continuous optimization for structure learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[160] R. Y. Zhong, L. Wang, and X. Xu. An iot-enabled real-time machine status monitoring approach for cloud manufacturing. *Procedia CIRP*, 63:709–714, 2017.

[161] P. Zhou, C. Xiong, X. Yuan, and S. C. H. Hoi. A theory-driven self-labeling refinement method for contrastive representation learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 6183–6197, 2021.

[162] Q. Zhu, Z. Chen, and Y. C. Soh. A novel semisupervised deep learning method for human activity recognition. *IEEE Trans. Ind. Informat.*, 15(7):3821–3830, 2019.

[163] I. Zliobaite, A. Bifet, M. Gaber, B. Gabrys, J. Gama, L. Minku, and K. Musial. Next challenges for adaptive learning systems. *SIGKDD Explor. Newsl.*, 14(1):48–55, Dec. 2012.

[164] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012.

[165] I. Zolotová, P. Papcun, E. Kajáti, M. Miškuf, and J. Mocnej. Smart and cognitive solutions for operator 4.0: Laboratory h-cpps case studies. *Comput Ind Eng*, 139:105471, 2020.

[166] Álvaro Segura, H. V. Diez, I. Barandiaran, A. Arbelaiz, H. Álvarez, B. Simões, J. Posada, A. García-Alonso, and R. Ugarte. Visual computing technologies to support the operator 4.0. *Comput Ind Eng*, 139:105550, 2020.

[167] D. Şahinel, C. Akpolat, O. C. Görür, F. Sivrikaya, and S. Albayrak. Human modeling and interaction in cyber-physical systems: A reference framework. *J Manuf Syst*, 59:367–385, 2021.