

Graph Learning for Cognitive Digital Twins in Manufacturing Systems

Trier Mortlock[†], Deepan Muthirayan[†], Shih-Yuan Yu[†], Pramod P. Khargonekar[†], Mohammad A. Al Faruque^{††}

Department of Electrical Engineering and Computer Science[†]

Department of Mechanical and Aerospace Engineering[†]

University of California, Irvine, California, USA

(tmortloc, dmuthira, shihyuay, pramod.khargonekar, alfaruqu)@uci.edu

Abstract—Future manufacturing requires complex systems that connect simulation platforms and virtualization with physical data from industrial processes. Digital twins incorporate a physical twin, a digital twin, and the connection between the two. Benefits of using digital twins, especially in manufacturing, are abundant as they can increase efficiency across an entire manufacturing life-cycle. The digital twin concept has become increasingly sophisticated and capable over time, enabled by rises in many technologies. In this paper, we detail the cognitive digital twin as the next stage of advancement of a digital twin that will help realize the vision of Industry 4.0. Cognitive digital twins will allow enterprises to creatively, effectively, and efficiently exploit implicit knowledge drawn from the experience of existing manufacturing systems. They also enable more autonomous decisions and control, while improving the performance across the enterprise (at scale). This paper presents graph learning as one potential pathway towards enabling cognitive functionalities in manufacturing digital twins. A novel approach to realize cognitive digital twins in the product design stage of manufacturing that utilizes graph learning is presented.

Index Terms—Digital Twin, Manufacturing Systems, Cyber-Physical Manufacturing Systems, Cognitive Systems, Industry 4.0, Graph Learning.

I. FUTURE OF MANUFACTURING

The evolution of manufacturing can be presently defined by four major transformations: (i) the industrial revolution during the 18th and 19th centuries; (ii) mass production in the first half of the 20th century; (iii) information technology-based automation of production in the second half of the 20th century; and (iv) the ongoing fourth industrial revolution. This fourth industrial revolution has many monikers, namely, Industry 4.0, Smart Manufacturing, connected industries — as part of Society 5.0 in Japan, Made in China 2025, etc. This wave of future visions of manufacturing is often hinged on harnessing the power of computing in manufacturing systems. Most of these visions aspire to bring together wireless (and wired) communications, smart sensors, cyber-physical systems, internet-of-things (IoT) [1], advanced robotics [2], additive manufacturing, simulation and high-performance computing, advanced data analytics, machine learning and artificial intelligence, cloud computing, and cybersecurity [3]. The goal of the fourth industrial revolution is to achieve personalized,

affordable, efficient, resilient, adaptive, and sustainable products and production across distributed factories and supply chains.

Digital twins are one of the many technologies that will help shape the future of manufacturing. Digital twins integrate the cyber and physical worlds — allowing for seamless communication between digital models and real, in-operation systems. Digital twins have been shown to provide distinct advantages in manufacturing [4]. Digital twins leverage many auxiliary technologies and systems such as modeling and simulation, IoT sensors, standards and interoperability among digital technologies, computing, and data from different stages of the product lifecycle. These enabling technologies that constitute increasingly sophisticated and powerful digital twins are illustrated in Fig. 1.

The deployment of digital twins in manufacturing can have an impact throughout the entire product life cycle — including, but not limited to, product design and optimization, testing, production system design and operation, supply chain management and control, prognostics, maintenance, aftermarket services, and cybersecurity. As with many modern research fields in computing, the applications of digital twins in manufacturing continues to rapidly progress and increase. Gartner has listed the digital twin as one of the top ten technology trends for 2019 and the years to come [6]. Furthermore, according to recent reports, the digital twins’ market size is projected to grow from US\$3.1 billion in 2020 to US\$48.2 billion in value by 2026 [7]. Historically, the aerospace and automotive industries have been dominant users of digital twin technologies; however, other manufacturing sectors are also beginning to leverage the many advantages that digital twins can offer [7]. Concurrently, digital twins have given rise to many exciting and challenging research questions.

In this paper, we extend upon the cognitive digital twin framework introduced in [5]. Specifically, we propose graph learning frameworks and algorithms as a potential pathway for cognitive digital twins. Graph learning is a subset of machine learning that focuses on algorithms and methods involving data that can be structured as graphs. Graph learning techniques, which have gained increasing interest in recent years [8], have many benefits in modeling non-Euclidean data that has inherent graph structures. The pervasive nature of graphs in manufacturing systems can be observed at all scales and stages — from process and fabrication flow graphs to high-level organization supply chains. Hence, graph learning

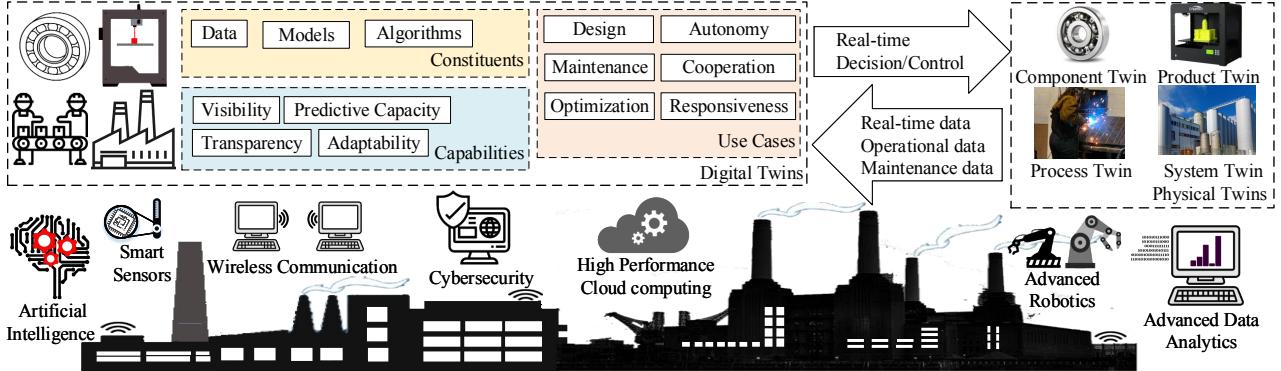


Fig. 1: Digital Twin in the context of cyber-physical production systems [5].

can be leveraged in future Industry 4.0 systems. Furthermore, these methods can be used to realize the vision of cognition in digital twins. We make the important clarification that graph learning is not the only solution to enabling cognition in digital twins, nor does it completely enable all levels of cognition across a system. It is simply just yet another computational method that takes advantage of data-driven modelling that can be used in systems moving towards further autonomy.

The main contribution of this work is a novel, graph learning framework for enabling cognition in digital twins. The framework we propose is a query-based framework that can potentially answer any type of query. The framework involves three main steps namely, (i) the *Graph Formation* step, which can aggregate data from diverse data types and from diverse products in a graph representation that can be used for reasoning or solving a specific problem, (ii) *Graph Operations* that process the graph further to select the key features, and (iii) the *Learning Objective* that specifies the problem to be solved to answer the query. This framework can leverage both physical and virtual data simultaneously, improve the core operations of manufacturing systems, and address some of the open research gaps in the field.

The remainder of this paper is structured as follows: Section II introduces digital twins and their importance in Industry 4.0; in Section III, the concept of cognitive digital twins is detailed; Section IV describes concepts of graph learning and its application in manufacturing along with a general graph learning framework for digital twins with illustration for the product design stage; and Section VII gives concluding remarks.

II. DIGITAL TWINS AND MANUFACTURING

The term *Digital Twin* was first used in 2002 by John Vickers of the National Aeronautics and Space Administration (NASA). During a later 2010 report, Vickers also gave the first formal definition of the digital twin, specified for air vehicles, as *an integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin* [9]. The most basic and concise definition of a digital twin can be stated as in [10]: “A digital twin has a digital or a virtual part, a physical part and a connection between them.”

Since its inception, the digital twin concept has evolved and expanded to various products, processes, and domains. Digital twins have been proposed for many applications or use cases such as predicting real-time behavior, monitoring, decision support, planning, production optimization, and control [11], [12]. The concept of digital twins in cyber-physical production systems is studied in [13], however strictly for data acquisition. In [14], a digital twin-based cyber-physical production system is examined, placing specific emphasis on the interconnection between the physical and cyber-shop floors. Our focus encompasses these works in a larger view of cyber-physical manufacturing (see Fig. 1) while providing a flexible framework for enabling cognitive functions throughout the system’s life cycle.

Fig. 1 illustrates the digital twin in the context of a cyber-physical production systems (CPPS) and the overarching use cases (or functions). The top part of the figure illustrates the digital twin ecosystem: (i) major building blocks and constituents of digital twins (data, models, and algorithms); (ii) capabilities digital twins can enable (visibility, transparency, predictive capacity, and adaptability); and (iii) the various use cases ranging from design to autonomy and cooperation. The bottom part of the figure shows the key enabling technologies of this ecosystem. The top-right side of the figure depicts the physical manufacturing system. CPPS related data (e.g., operational data and maintenance data) are collected in real-time and provided to the digital twins. Digital twins in response provide real-time feedback (e.g., decision and control) to the physical manufacturing system. This provides real-time two-way seamless communication between the physical manufacturing system and the corresponding digital twin.

The use cases of digital twins span the entire life cycle of a product: product design and optimization, testing, production system design and operation, supply chain management and control, prognostics, maintenance, aftermarket services, cybersecurity (Fig. 1). For example, the digital twin of a component or a product can be used to simplify and streamline the design process by enabling virtual testing of the specifications to ensure that the product meets the standards (verify) and the performance requirements (validate) [12]. The digital twin of a product can be used to detect the early onset of faults (predictive maintenance), help diagnose the fault, and provide customized solutions for performance optimization, mainte-

nance, and compliance [12]. A manufacturing company can offer new services based on digital twins for optimizing the performance of the product during operation. Thus, computing through digital twins provides organizations with opportunities for offering new products and services. The digital twin of a production system can be used to optimize the system for throughput (performance), reduce waste (efficiency), improve quality [12]. Digital twins can also enable processes involved in CPPS to be adaptable and responsive to disruptive events [15].

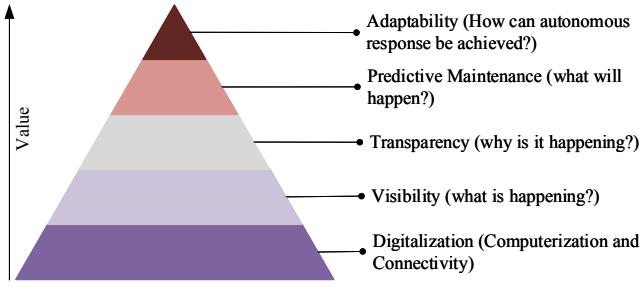


Fig. 2: Pathway to Industry 4.0 [5].

In our previous paper we envisioned that cognitive digital twins are the key for the various stages of the Industry 4.0 development path (proposed in [16]): visibility (what is happening), transparency (why is it happening), predictive capacity (what will happen), and adaptability (how can an autonomous response be achieved) (see Fig. 2, which is an adaptation of Fig. 3 in [16]). Visibility is vital for assessing the shop floor changes and operating conditions for the overall system's adaptability and responsiveness. This will impact decisions in the production pipeline, allowing the system to adapt quickly and effectively, therefore reducing downtime and costs. Transparency and predictive capacity are essential for understanding and inferring how to respond, and thus directly affect manufacturing performance under changing circumstances. Finally, adaptability enables the system to respond to a changing situation by itself with self-corrections based on feedback, instead of a human-in-the-loop making the decisions. When this happens across interacting physical processes, the system can achieve seamless cooperation in how it responds and operates.

There are many different algorithms and implementations that may be used to realize these discussed capabilities in digital twins (e.g., Neural Networks, Dynamic Bayesian Networks, Hidden Markov Models, Finite Element Analysis, and Deep Learning [15]). Graph learning, another example, can enhance visibility by learning relationships between systems and subsystems — creating a higher-level abstracted vision of an entire system. Graph representations can also provide more explainable decisions than traditional machine learning algorithms as relations between components of a system can be mapped and visually interpreted. Also, graph learning can help in prediction tasks — for example, if a tool in a manufacturing plant breaks down or requires repair, the graph structure can enable informed predictions on other tools that may need preventative repairs or reinforcements. Large scale graphs can

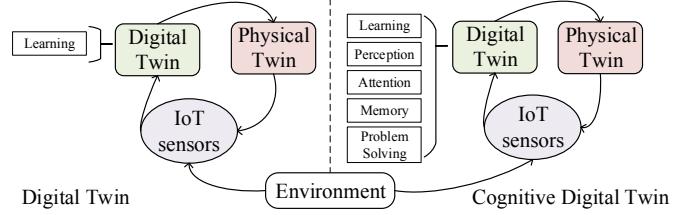


Fig. 3: Standard digital twin and cognitive digital twin [5].

enable higher levels of autonomy in digital twins by allowing abstraction through a holistic view of a system, much like how humans perceive their environment. *We posit that graph learning, further detailed in Section IV, provides a pathway for cognition in digital twins.*

III. COGNITIVE DIGITAL TWIN

The definition of *cognitive digital twin* is inspired by major advances in cognitive science, machine learning, and artificial intelligence. Neisser's classic definition of cognition [17] includes “*all the processes by which the sensory input is transformed, reduced, elaborated, stored, recovered and used ...*”. Fundamental aspects of cognition include attention (selective focus), perception (forming useful precepts from raw sensory data), memory (encoding and retrieval of knowledge), reasoning (drawing inferences from observations, beliefs, and models), learning (from experiences, observations, and teachers), problem-solving (achieving goals), knowledge representation, etc.

The standard view of the digital twin and the conceptual framework of the cognitive digital twin proposed in [5] are depicted in Fig. 3. The digital twin on the left of Fig. 3 is the standard digital twin, which has a digital part, a corresponding physical part, and a connection between them. This version of the digital twin has the ability to learn. The digital twin we propose is shown on the right of Fig. 3, which in addition to having the ability to learn, is endowed with the other elements of cognition such as perception, attention, memory, reasoning, problem-solving, etc. In the following, we describe these capabilities in the context of a digital twin.

A. Cognitive Capabilities

1) *Perception*: in cognitive psychology could be defined as *the organization, identification, and interpretation of sensation to form a mental representation* [18]. We extend this definition to define perception in cognitive digital twin as *the process of forming useful representations of data related to the physical twin and its physical environment for further processing*. It is well established that machine learning techniques are less effective in learning representations of high dimensional and large data volume [19]. Since CPPS (and IoT) generate multi-modal, high-dimensional, large volumes of data, we posit that perception is a key cognitive capability to form useful *precepts* upon which further cognitive processing can occur in a digital twin. Perception in a digital twin will enable visibility in manufacturing systems.

2) *Attention*: can be viewed as the allocation of limited resources or a selection mechanism [20]. We adopt the latter view and define attention in a cognitive digital twin as *the process of focusing selectively on a task or a goal or certain sensory information either by intent or driven by environmental signals and circumstances*. Attention can be perceptual or non-perceptual and controlled or otherwise (see [21] for a detailed taxonomy of attention). Attention enables focus on the essential information from the raw sensor data and memory. So, it can simplify and improve the process of perception and decision making in a cognitive digital twin. Attention will help monitor or select a task to focus on, paving the way for autonomy in manufacturing systems.

3) *Memory*: we define memory in a cognitive digital twin adopting the view of memory provided in [22]: *is a single process that reflects a number of different abilities: holding information briefly while working with it (working memory), remembering episodes of the physical twin's life (episodic memory), and knowledge of facts of the environment and its interaction with the physical twin (semantic memory), where remembering includes the steps: encoding information (learning it, by perceiving it and relating it to past knowledge), storing it (maintaining it over time), and then retrieving it (accessing the information when needed)*. Thus, memory (both working memory and the remembered episodes and knowledge), are an essential ingredient for the algorithms complementing the digital twin to autonomously control the physical processes related to the various stages of a physical twin because memory allows the algorithm to remember the context and additionally allows the digital twin to leverage past knowledge.

4) *Reasoning*: in cognitive psychology can be broadly defined as the “process of drawing meaningful conclusions for informing problem-solving or decision making” [23]. Reasoning can be broadly classified under deduction, induction, and probabilistic reasoning [24]. *Thinking* and *reasoning* are cornerstones of human intelligence and so have been extensively studied in cognitive psychology [23], [25]. We define reasoning in cognitive digital twins adopting the definition proposed in [26]: *drawing conclusions consistent with a starting point — a perception of the physical twin and its environment, a set of assertions, a memory, or some mixture of them*. Thus, reasoning directly impacts understanding (transparency) and is central to decision making.

5) *Problem-solving*: we define problem-solving in cognitive digital twin as *the process of finding a solution for a given problem or achieving a given goal from a starting point*. Thus, problem-solving is central to decision making and autonomy.

6) *Learning*: we define learning in cognitive digital twin as *the process of transforming experience of the physical twin into reusable knowledge for a new experience*. Hence, learning is essential for adaptability (or autonomy) and responsiveness of the physical system that the digital twin represents and becomes a key ingredient for intelligence in digital twins.

IV. GRAPH LEARNING FOR COGNITIVE DIGITAL TWINS

Data driven models based on artificial intelligence and machine learning have become increasingly popular for enabling

some cognitive capabilities in digital twins at various levels of manufacturing systems [27]. In this section, we present graph learning as just one of many solutions that can be used to enable cognition in digital twins. In the subsequent subsections, we briefly provide preliminaries on graph learning along with motivation for the use of graphs in manufacturing systems. In Section IV-E, we present specialized graph learning frameworks and discuss how they can enable cognitive capabilities in digital twins and offer potential solutions for important research challenges within the field of manufacturing systems.

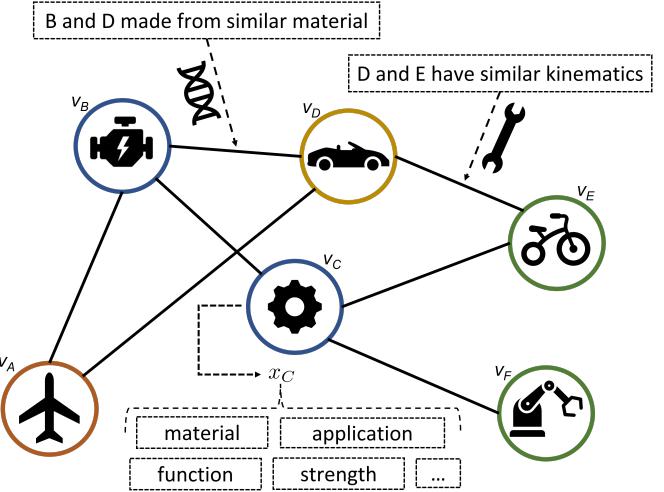


Fig. 4: A generic undirected graph formed in the context of product design.

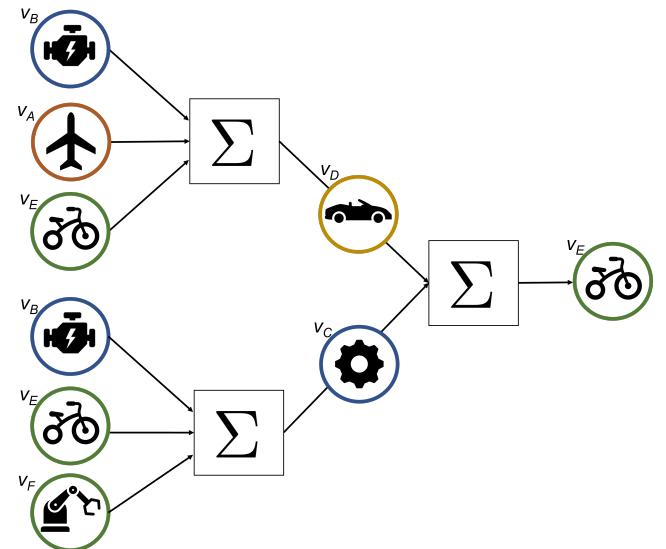


Fig. 5: Neighborhood node aggregation of node E .

A. Preliminaries on Graph Learning

We define a graph with n nodes as, $\mathcal{G} = \{\bar{V}, \bar{A}\}$, where $\bar{V} = [v_1, v_2, \dots, v_n]$ is the matrix of the node embedding in the graph, v_i is the node embedding for node i . The edges of a graph are represented by the adjacency matrix, \bar{A} , where each

TABLE I: Related Works of Graphs Learning in Manufacturing

Manufacturing Stage	Graph Function	References
Design	perform automation engineering tasks using functional lifting	[28]
	design of distributed circuits	[29]
	unsupervised 3D shape retrieval	[30]
Optimization	optimize scheduling performance of flexible manufacturing systems	[31]
	find the optimum scheduling policy for job-shop problems	[32]
Monitoring	detect and isolate faulty components in industrial systems	[33]
	improve product failure prediction	[34]
	share multi-level manufacturing knowledge within a system	[35]
	predict the remaining useful life estimation of industrial equipment	[36]

value a_{ij} corresponds to the relation between nodes i and j . The feature matrix of the graph, \bar{X} is composed of each nodes feature vectors, x_i , which can hold unique data pertaining to each node. Fig. 4 shows a basic graph of six nodes with various connections. This example is set in a manufacturing design setting where each node is a product, and the edges are possible connections between products. Additionally, some examples of features of product C are also shown.

Graphs can be classified as (i) *directed* or *undirected* and (ii) *weighted* or *unweighted*. Directed graphs are graphs where the edges represent relations that are uni directional. The opposite is the undirected graph where the edge is a bi-directional relation. Weighted graphs are graphs where the edges have an associated weight that specifies the relative importance of the relation. The concepts we present later hold for graphs of any type.

Graph learning broadly includes two steps: the formation of the graph and finding a lower dimensional representation of the graph. Graph neural networks are a widely popular and successful deep learning technique for generating lower dimensional representation of graph structure data. The first *Graph Neural Network* (GNN) model was developed in [37], and since then, numerous graph approaches have been proposed such as *Recurrent Graph Neural Networks* (RecGNNs), *Convolutional Graph Neural Networks* (ConvGNNs), *Graph Autoencoders* (GAEs) and *Spatio-temporal Graph Neural Networks* (STGNNs) [8]. One of the key techniques contributing to graph learning's rise in popularity is notably the Graph Convolutional Network (GCN), developed by [38] in 2016.

Graph neural networks typically include the following two operations: (i) message passing and (ii) node embedding. Message passing is the function that aggregates features from a node's neighbors. This operation updates the features of nodes with the information from their respective neighbors and its current feature values. Multiple message passing operations will then result in a final node feature that is a function of information from nodes across the graph. Figure 5 shows one such instance of message passing for the graph in Fig. 4. The node embedding operation encodes the final feature vector as a lower dimensional representation. These two operations form the key parts of most general graph neural network approach.

B. Graph Learning in Manufacturing

Graph learning has some fundamental advantages, specifically: (i) modeling of non-Euclidean data; (ii) expressing data in an insightful manner to understand relationships between entities of a system; (iii) abstracting data for higher-level reasoning tasks; (iv) capturing dependencies between the different model instances of a system. Since graphs are prevalent in manufacturing systems (see Section II), it follows that graph learning can provide significant advantages in manufacturing. In Table I, works that illustrate the use of graph learning in various phases of manufacturing systems are listed. In [28]–[30], it is shown that graphs of similar or related products can be used to leverage information sharing and save time and money for verification, validation, and testing in the design stage. In [31], [32], it is shown that graph learning can be used to optimize scheduling performance during operation. In monitoring systems, graph learning has been shown to improve detection and prediction of failures in both products and equipment [33], [34], [36]. While these demonstrations highlight the opportunities that lie for graph learning in manufacturing systems, the graph learning systems that these works develop are restricted to specific applications. In the following sections, we present a novel and general graph learning framework for any digital twin application. We focus on the design stage of manufacturing to illustrate our ideas.

C. Product Design

Benefits of cognition can be realized throughout manufacturing, like shown in [39]. They propose a cognitive production planning and control system that it is able to react and autonomously adapt its production planning process to increase manufacturing efficiency. They employ cognitive functions of perception, knowledge, and planning within their architecture. Cognitive IoT is another application that can be used in manufacturing systems as sensors themselves can exercise cognitive capabilities [40]. While cognition can impact various stages of Industry 4.0 and related technologies, here we focus on the product design stage to illustrate our key ideas. The potential impact of this focus is far-reaching — advancements in product design can lead to more robust designs, cost efficient research and development, and overall more streamlined designs. The product design stage of manufacturing can be defined as steps starting from the idea generation of a product to the physical testing, verification, and validation of the

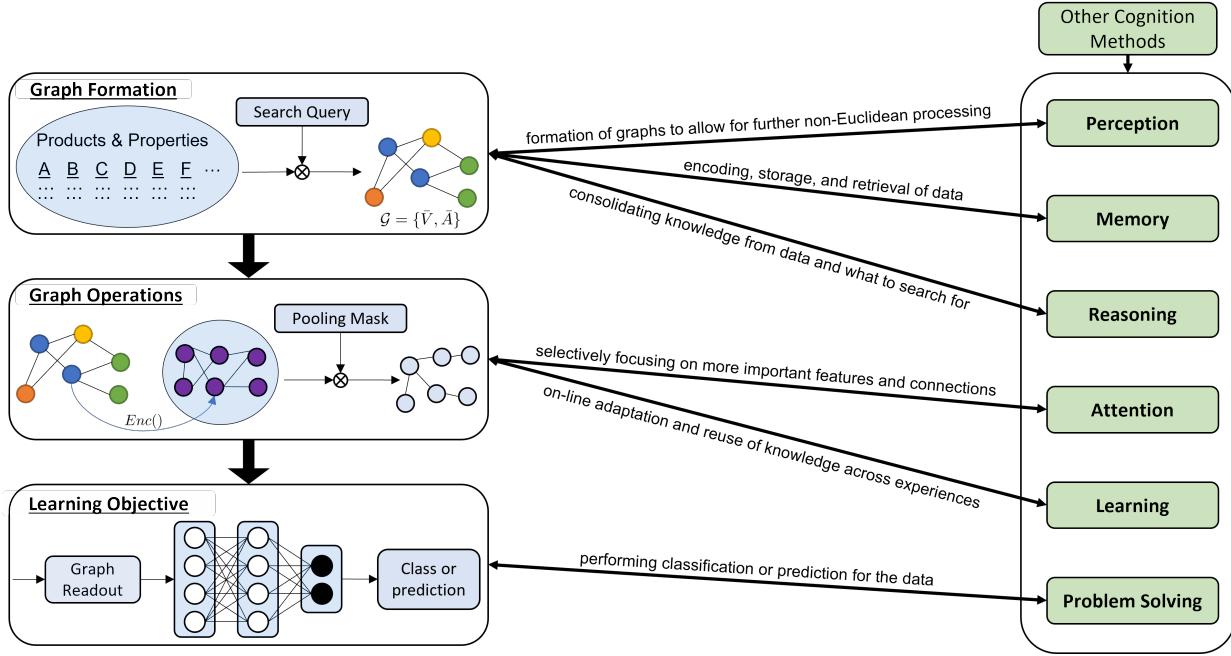


Fig. 6: An example of a cognitive digital twin using graph learning during the product design stage of manufacturing. The graph learning framework is composed of three steps: (i) **Graph Formation**: products and their properties are filtered by a user defined design principle query to form a graph of the products and their relations; (ii) **Graph Operations**: higher level abstractions or relations are obtained from the generated graph; and (iii) **Learning Objective**: defines the problem to be solved (e.g., classification, prediction) for answering the query and the general metrics and specifications to optimize and fine-tune the model.

product before manufacturing. Current practices in this domain often target their efforts in two separate streams, one that focuses on physical data and the other that focuses on virtual data. Thus both do not take advantage of the benefits a digital twin can offer [41]. In the following subsections, we discuss the application of graph learning to the *search* operation of the design phase. We discuss the benefits that arise and comment on the limitations and future research directions.

D. The Search Operation

The *search* operations in the product design stage can be defined as processes that involve transferring useful knowledge and information from other products or systems that can aid in the design of the product under development. Currently, researchers and engineers spend a large amount of time building a new product, process, or system without leveraging the knowledge that already exists for a similar or somewhat closer product, process, or system. For example, there are resources such as GrabCad [42], which contains computer-aided design (CAD) models of many engineering systems, that can be used as a base design for creating new models. These CAD models are part of the digital twin of the product that goes through the manufacturing process to be converted into the physical twin. Various methods have been proposed for exploring 3D models (a partial digital twin model defining the physical twin geometry and manufacturing information). For example, authors in [43] created a 3D search engine that utilizes the spherical harmonics descriptor for acquiring the signature vector and use the Euclidean distance among

these vectors to find a similar polygon model. Authors in [44] utilized the shape similarity metric on 3D models to enable the discovery of parts and assemblies existing in the company's database. Authors in [45] utilized a similarity measure based on various attributes (such as name, description, etc.) to calculate a score for discovering CAD models in repositories. While these research efforts focused on searching and discovering similar 3D models based on geometry information, the digital twin models comprise more than just geometry information of the physical twins and thus can be used for searching and responding to more diverse search queries. *We posit that graph learning techniques by virtue of their capability to learn general relationships are critical to building a general purpose product search engine.*

E. Graph Learning Framework

In Fig. 6, we portray the graph learning framework for enabling cognition in digital twins. Cognitive functions allow humans to transfer their knowledge and experiences from one domain to a completely different domain. While this is a big challenge for current machine learning methodologies, graph learning is a potential machine learning technique that can bridge this gap. In Fig. 6, the functional steps of the graph learning framework are shown on the left and the cognitive functions it enables are shown on the right.

The framework we propose is a query-based framework. The query can be either any user generated query or an automated query from the digital twin. The query can potentially cover a range of tasks or problems like finding functionally

similar products or generating configuration for a new set of product specifications, or even generating novel insights for new products. Thus, the scope of this framework is very broad. The graph learning framework will allow the digital twin to answer such higher level queries by identifying complex product to product or product to sub-component relationships and solving problems by processing this information. In the **Graph Formation** step, data of products and their properties are mined and organized as a graph based on the query, allowing them to be processed further. The constructed graph captures the essential relations and provides a powerful abstraction or representation that can enable reasoning or problem solving using graph operations to answer the query. The **Graph Operations** step, can model very general and complex mathematical functions. Graph operations can further aggregate information and form condensed representations as shown in the figure, which is useful for generating the final decision or insight to answer the query. Graph neural networks, in principle, can model any mathematical function, and so are a potential approach for this step. Lastly, the **Learning Objective** specifies the problem to be solved for answering the query and the general metrics and specifications to optimize and fine-tune the model for solving this problem. Specifying the correct metrics for training and verifiability are critical to ensure the digital twin can effectively and reliably optimize the model for the given problem (e.g. classification, prediction). Accurate problem specification is also critical for generating the best response to the query. This step's output and subsequent feedback can also be used to iteratively refine and improve the model.

By leveraging this framework, digital twins can learn to organize information from diverse domains by their complex relations based on the query (which can be a search query). While physical testing is not completely substitutable, this method can make the design process more efficient by allowing the designer to leverage prior knowledge. For example, a designer targeting a specific design objective can initiate their query, and the framework can enable aggregation of knowledge from various domains to generate new insights for the design of the product. This will also enable digital twins to leverage physical and virtual data simultaneously, and find improved solutions for the specific queries, addressing a key gap that we alluded to in Section IV-C. We also envision that future cognitive digital twins will be able to make these queries autonomously. We note that full cognition in manufacturing will leverage a combination of other methods, as indicated in the top right of Fig. 6, and that graph learning is one of the key methodologies. In the next section, we showcase a real-world application that illustrates some functionalities of the proposed framework.

V. USE CASE

In this section, we review experiments and results of an application to illustrate the proposed methodology. In a previous work, our group presented a specialized graph learning framework that uses the concept of functional lifting for future automation engineering in the product design space

[28]. Automation engineering in this context can be defined as automation of the *design, creation, development and management of production systems in factories, process plants, and supply chains that realize the production of products*. Automated functional lifting refers to the strategy of inferring functions of products based on abstractions of common engineering data. In these experiments, the goal is to produce accurate classification labels for different product types using engineering data artifacts. This classification enables more efficient search operation within and across different products with similar functionalities. To realize these operations, a graph learning framework with a structural graph convolutional neural network (SGCNN) is proposed.

The dataset was generated by scraping GrabCAD [42] for six different categories of 3D CAD models with related functions: *Car, Engine, Robotic Arm, Airplane, Gear, and Wheel*. Functional information such as model's name, author, description of the model, name of parts in the model, tags, likes, timestamps, and comments on the models was also extracted. The total number of models per category was 2,271 for *Car*, 1,597 for *Engine*, 2,013 for *Robotic Arm*, 2,114 for *Airplane*, 1,732 for *Gear*, and 2,404 for *Wheel*. These were later broken down into a testing-training split that is detailed in Section V-B.

A. Graph Learning Framework

The graph learning framework proposed in [28] is depicted in Fig. 7. Their framework can be viewed as an application of our cognitive digital twin framework. To illustrate this, we present their proposed framework as it aligns with the different stages of our cognitive digital twin framework: Graph Formation, Graph Operations, Learning Objective. We detail these stages in their framework below.

Graph Formation

In this stage, the input dataset is processed into graph structures based on a search query which is dependent on the target design application. Specifically, the Sample Generator generates subgraphs, \mathcal{G}_n , based on the queried schema of interest, which in this setting is the information from the extracted 3D CAD models. This schema is designed to extract functional information from each sample. The embedding of sample information into the feature vectors is done by the well-known *word2vec*, and the induced subgraphs are composed of the functional information for each sample. These subgraphs are aimed at capturing the meaning of the engineering design and enable functional lifting.

Graph Operations

In this stage, both graph pooling and graph convolutions are performed over the subgraphs. This involves aggregating information from neighbor nodes within a subgraph, appending this information to the original subgraphs, and then processing the subgraphs through convolutional layers. The output after these subsequent convolutions is processed through a nonlinear activation function in order to perform the classification. The final output is the probability of the various classes. The processing done during this stage enables functional lifting by abstracting the common engineering data across subgraphs in a

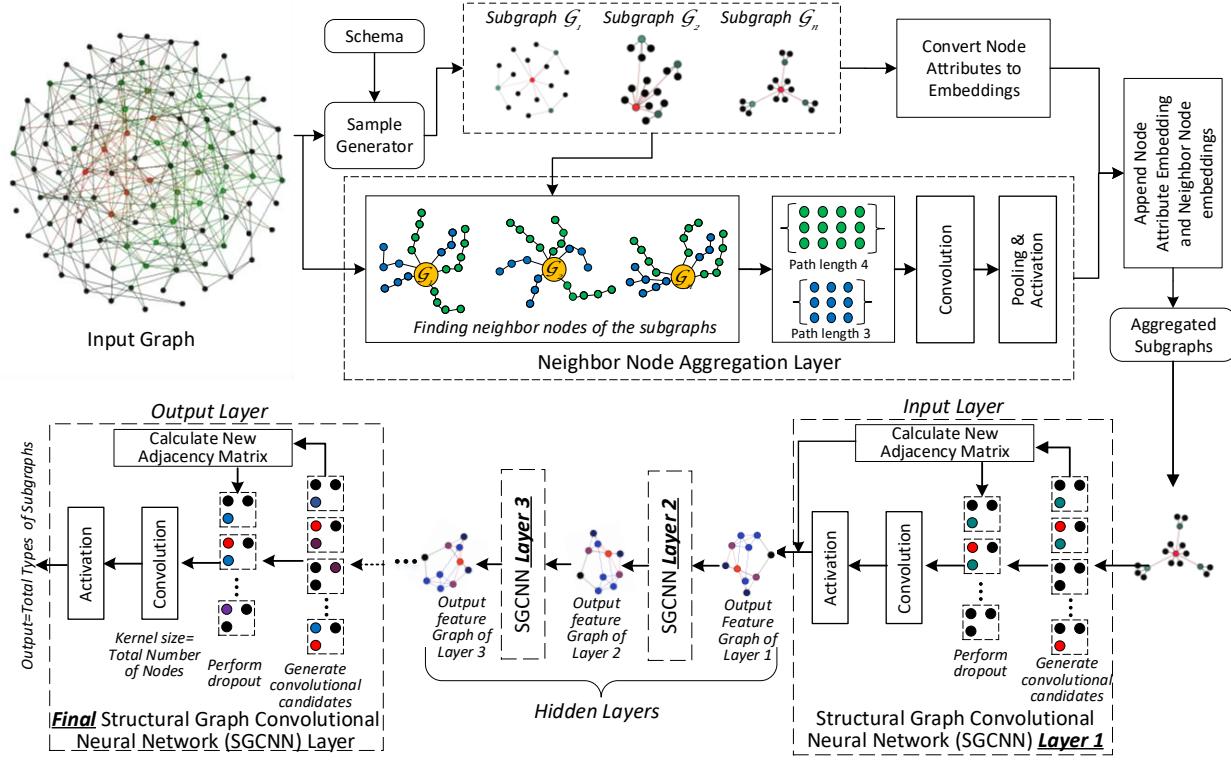


Fig. 7: SGCNN Architecture [28].

way that they can mathematically utilized. The Neighborhood Node Aggregation Layer performs convolutions and pooling to aggregate information from each node's neighbors into the created subgraphs. Here, the common method `node2vec` that combines a breadth/depth first search over graphs is utilized. The feature vector for each node, x_n , is calculated as follows:

$$x_n = \sigma(f_{pool}(\bar{W} \circledast \bar{N}) + b), \quad (1)$$

where $\sigma(\cdot)$ is the activation function; f_{pool} is a pooling function; \bar{W} is a 1 by d trainable weight matrix with d as a defined search depth; \circledast represents the convolution operation; \bar{N} is n by d matrix of the neighbor node's features (as illustrated roughly in Fig. 5); and b is a bias variable.

The aggregated subgraphs are then fed into the SGCNN layers which consist of subsequent convolutions, graph pooling, and non-linear activations. In order to perform the 2D convolutions over the subgraphs, an attribute matrix, \bar{R} , is calculated as:

$$\bar{R} = \bar{X} \circ (\bar{A} + \bar{I}), \quad (2)$$

where \bar{X} is the feature matrix for the subgraph (as described in Section IV-A); \circ denotes the Hadamard Product of two matrices; \bar{A} the adjacency matrix of the subgraph (as described in Section IV-A); and \bar{I} denotes the identity matrix.

Before the convolutions are applied, a special graph pooling algorithm is used to down sample \bar{R} to a k by k matrix, \bar{R}^k , where k is the size of the convolutional kernel. The trainable k by k kernel matrix, \bar{W}^k , is passed over the subgraph as

defined in the following equation:

$$x^k = \phi(\bar{W}^k \circledast \bar{R}^k + b), \quad (3)$$

where $\phi(\cdot)$ is the non-linear activation function. This graph convolution operation is able to effectively aggregate local features for each node, in a similar manner that traditional convolutions over images are applied. The new subgraph, adjacency, and attribute matrix are then calculated using the previous subgraph and passed to the next SGCNN layer accordingly.

Learning Objective

The overall graph learning objective for classification is specified by the cross-entropy loss function, H , over the output of the final SGCNN layer:

$$\max H, H = \sum_{i \in \text{classes}} y_i \log_e(\hat{y}_i), \quad (4)$$

where y_i is the indicator of the ground truth validity of the label i and \hat{y}_i is the inferred probability of the label i . It is clear that when the learning objective is maximized, the predicted probabilities will closely match the ground truth labels. During training of the model, the weights of the graph pooling and convolution operations at each layer are updated. As in traditional supervised learning training approaches, the model has access to the ground truth labels to compute the loss function. During testing of the model, the performance is then evaluated against subgraphs it has not previously been trained on.

For training and evaluation of the SGCNN architecture a train-test split of 11,304 to 2,827 samples was used. Through

extensive experiments, the best SGCNN model was identified to be four layers. It was also observed that as both the number of hidden layer features and output kernel size were increased the model achieved better classification scores. It was shown that the model was able to correctly classify roughly 91% of the subgraphs based on their functionality. We refer readers to [28] for further information regarding the model structure and tuning of hyperparameters and ablation studies.

B. Cognitive Capabilities

The cognitive capabilities are utilized throughout the framework's core operations. *Perception* is utilized in the representation of the data on subgraphs, and the subsequent embedding of features that are abstracted to a level for the model to perform further computations. *Memory* and *learning* are utilized continuously in the update and the optimization of the model parameters during the training process. In the graph operations, *attention* and *reasoning* are effectively employed to reuse knowledge across different samples while selectively focusing on the most relevant functional attributes. Thus, the framework implements several key cognitive functions.

C. Further Extensions

An additional output that the graph framework could produce is the clustering of the nearest-neighbor subgraphs that serve similar design functions. This could allow the digital twin to learn from similar designs at a functional level when tasked with searching for similar products, as opposed to searching for only similar products of the same basic design. This would require a multi-task learning approach because with this additional objective the model must be trained for accurate classification and clustering. Even further, this framework can be extended for higher level cognitive functions like reasoning, decision making and problem solving by enabling a feedback loop that self-generates queries. This can potentially enable complete autonomy and fully realize the capabilities of cognitive digital twins.

VI. DISCUSSION

In this section we discuss current challenges and limitations, extensions to other areas of manufacturing, and an outlook on future research directions in the field.

Challenges:

There are some specific challenges with regards to the proposed framework. Formation of graphs, problem and metric specification, and hyper-parameter tuning of graph operations can require expert knowledge. In the use case in Section V, extensive studies on the optimal model and hyperparameters, such as various activation functions, number of hidden features per layer, kernel size, and number of total layers were needed. Techniques for formation of graphs are still relatively under-explored and therefore may be burdensome in some applications. Additionally, there are some scenarios where the data is not easy to represent as a graph. Further abstraction of systems and/or processes may be necessary to arrive at a graph-level representation. Nonetheless, applications of graph

learning approaches to these situations are still under explored. All of these are open research challenges.

Another notable challenge is the availability of data. Due to constraints in the sharing of propriety information, some companies may be prohibited from collaboration with others. Additionally, financial and competitive motivations may also limit progress along this direction. However, as Industry 4.0 continues to develop, the amount of available data will continue to increase as well. This will enable digital twins using data-driven approaches to become more robust and efficient over time.

Other Applications in Manufacturing:

The framework we presented can be extended to the entire life cycle of a product. During actual manufacturing, information about operations, products, logistics, and diagnostics are exchanged between various systems. This data can be modeled on graphs and graph learning can be leveraged to create new solutions to key manufacturing problems. Likewise, high-level organizational supply chains and consumer-provider supply and demands can be modelled on graphs and optimized towards specific performance objectives. Another powerful extension stems from a cognitive digital twin's ability to draw on its own past experiences (or experiences from other digital twins) to learn and adapt over time. Graph learning can incorporate temporal features, features that vary over time, which can allow a digital twin to learn from past data and observe specific time-dependent trends. This can enable iterative learning over design cycles and more intelligent design tools. Likewise, the growing areas of reconfigurable/flexible manufacturing can benefit from similar methods.

Future Research Directions:

The research community has immense opportunities to contribute in the areas of mathematical representations, algorithms, tools, and methodologies for developing and using cognitive digital twins. In this context, we formulate and pose a few research questions.

- What are the appropriate mathematical representations of digital twins that can enable the incorporation of cognitive capabilities? Examples here include differential equations, discrete-event dynamic systems, logic-based models, graph models, connectionist network models, etc. How can such models be used for simulations, state estimation, and control and decision making?

- How can high-performance computing and numerical simulation tools be leveraged to enable cognitive capabilities in digital twins? For example, can numerical simulations (along with experimental data) create large memory banks that can be used for interpreting and acting on real-time streaming data from IoT sensors? Can they be used for real-time response to changes in the manufacturing system environment?

- How can we enable public searchability of the digital twin models? More specifically, how to embed metadata in complex digital twin models (parts, processes, and systems) so that they can easily be searched over the internet during the design phase?

- How do we make the knowledge sharing scalable in digital twin models? Scaling may fall under the scope of generalizing knowledge sharing across multiple domains. Scalability is non-

trivial and is a challenge due to the complexity of cross-domain knowledge sharing. However, the digital twin models may be capable of sharing knowledge across non-overlapping domains (for example, across manufacturing systems utilizing different technologies, between the aging model and quality prediction model, etc.).

Among these future research directions, we believe that there is not one sequential path that takes precedence over others in the future development of cognitive digital twins. Different applications and studies will call for varying degrees of focus on these directions and may involve pursuing multiple directions simultaneously. Having said that, prioritizing development of strong theoretical and mathematical methods will be necessary to address many of these directions. Public searchability and scalability could be developed in tandem as they can work synergistically together to enable a proliferation of cognitive digital twin usage and prevalence.

VII. CONCLUSION

This paper presented a comprehensive vision of future cognitive digital twins that can enable advancements in Industry 4.0. The benefits of using digital twins in manufacturing were detailed. Six core cognitive capabilities, *perception, attention, memory, reasoning, problem-solving, and learning*, were described along with their ability to influence complex manufacturing decisions and future autonomy. A novel, query-based graph learning framework for enabling cognition in digital twins was presented to fill existing research gaps in the field. We believe that to realize the full potential of cognitive digital twins, collaboration across different research communities is essential. Domain expertise in many areas will be necessary in the development of the future of manufacturing.

ACKNOWLEDGMENT

This research was supported by NSF awards CMMI-1739503 and ECCS-1839429. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect views of our funding agencies.

REFERENCES

- [1] N. Jazdi, "Cyber Physical Systems in the Context of Industry 4.0," *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–4, 2014.
- [2] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, "Industry 4.0: A Review on Industrial Automation and Robotic," *Jurnal Teknologi*, vol. 78, no. 6-13, 2016.
- [3] S. R. Chhetri, N. Rashid, S. Faezi, and M. A. Al Faruque, "Security Trends and Advances in Manufacturing Systems in the Era of Industry 4.0," *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1039–1046, 2017.
- [4] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. Al Faruque, "Digital Twin of Manufacturing Systems," *Center for Embedded and Cyber-Physical Systems*, vol. CECS Technical Report CECS TR 17-07, 2017.
- [5] M. Al Faruque, D. Muthirayan, S.-Y. Yu, and P. Khargonekar, "Cognitive Digital Twin for Manufacturing Systems," *Design, Automation, and Test in Europe Conference*, 2021.
- [6] "Gartner Top 10 Strategic Technology Trends for 2019," 2019. [Online]. Available: <https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019/>
- [7] "Digital Twin Market Worth \$48.2 Billion by 2026," 2020. [Online]. Available: <https://www.marketsandmarkets.com/PressReleases/digital-twin.asp>
- [8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [9] E. Negri, L. Fumagalli, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *Procedia Manufacturing*, vol. 11, pp. 939–948, 2017.
- [10] J. Hultgren and K. Lundström, "The Concept of Digital Twins in the Manufacturing Industry: A Study Untangling the Digital Twin Concept to Utilize Its Benefits," *Master's Thesis, Chalmers University of Technology*, 2020.
- [11] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in Manufacturing: A Categorical Literature Review and Classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [12] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital Twin in Industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.
- [13] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335–340, 2017.
- [14] K. Ding, F. T. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors," *International Journal of Production Research*, vol. 57, no. 20, pp. 6315–6334, 2019.
- [15] K. Y. H. Lim, P. Zheng, and C. H. Chen, "A State-of-the-art Survey of Digital Twin: Techniques, Engineering Product Lifecycle Management and Business Innovation Perspectives," *Journal of Intelligent Manufacturing*, pp. 1–25, 2019.
- [16] A. Ignatov and P. Korolev, "The Global Energy Association and the Perspectives of Future Development of the Energy Sector within the Fourth Industrial Revolution," *IOP Conference Series: Earth and Environmental Science*, vol. 390, no. 1, p. 012052, 2019.
- [17] R. L. Solso, M. K. MacLin, and O. H. MacLin, "Cognitive Psychology". Pearson Education, New Zealand, 2005.
- [18] D. Schacter, D. Gilbert, D. Wegner, and B. M. Hood, "Psychology: European Edition". Macmillan International Higher Education, 2011.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [20] M. I. Posner, "Cognitive Neuroscience of Attention". Guilford Press, 2011.
- [21] K. Oberauer, "Working Memory and Attention – A Conceptual Analysis and Review," *Journal of Cognition*, vol. 2, no. 1, 2019.
- [22] K. B. McDermott and H. L. Roediger, "Memory (Encoding, Storage, Retrieval)," *General Psychology*, pp. 117–153, 2018.
- [23] R. J. Sternberg and J. P. Leighton, "The Nature of Reasoning". Cambridge University Press, 2004.
- [24] J. St.B. T. Evans, "The Cognitive Psychology of Reasoning: An Introduction," *The Quarterly Journal of Experimental Psychology*, pp. 561–567, 1993.
- [25] J. Baron, "Thinking and Deciding". Cambridge University Press, 2000.
- [26] P. N. Johnson-Laird, "Mental Models and Human Reasoning," *Proceedings of the National Academy of Sciences*, vol. 107, no. 43, pp. 18243–18250, 2010.
- [27] S. Abburu, A. J. Berre, M. Jacoby, D. Roman, L. Stojanovic, and N. Stojanovic, "COGNITWIN-Hybrid and Cognitive Digital Twins for the Process Industry," in *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2020, pp. 1–8.
- [28] J. Wan, B. S. Pollard, S. R. Chhetri, P. Goyal, M. A. A. Faruque, and A. Canedo, "Future Automation Engineering Using Structural Graph Convolutional Neural Networks," *arXiv preprint arXiv:1808.08213*, 2018.
- [29] G. Zhang, H. He, and D. Katabi, "Circuit-GNN: Graph Neural Networks for Distributed Circuit Design," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7364–7373.
- [30] W. Nie, Y. Zhao, A.-A. Liu, Z. Gao, and Y. Su, "Multi-graph Convolutional Network for Unsupervised 3D Shape Retrieval," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3395–3403.
- [31] L. Hu, Z. Liu, W. Hu, Y. Wang, J. Tan, and F. Wu, "Petri-net-based Dynamic Scheduling of Flexible Manufacturing System via Deep Reinforcement Learning with Graph Convolutional Network," *Journal of Manufacturing Systems*, vol. 55, pp. 1–14, 2020.
- [32] J. Park, J. Chun, S. H. Kim, Y. Kim, and J. Park, "Learning to Schedule Job-shop Problems: Representation and Policy Learning using Graph

Neural Network and Reinforcement Learning,” *International Journal of Production Research*, pp. 1–18, 2021.

[33] H. Khorasgani, A. Hasanzadeh, A. Farahat, and C. Gupta, “Fault Detection and Isolation in Industrial Networks using Graph Convolutional Neural Networks,” in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019, pp. 1–7.

[34] S. Kang, “Product Failure Prediction with Missing Data using Graph Neural Networks,” *Neural Computing and Applications*, pp. 1–10, 2020.

[35] L. He and P. Jiang, “Manufacturing Knowledge Graph: a Connectivism to Answer Production Problems Query with Knowledge Reuse,” *IEEE Access*, vol. 7, pp. 101 231–101 244, 2019.

[36] J. Narwariya, P. Malhotra, V. TV, L. Vig, and G. Shroff, “Graph Neural Networks for Leveraging Industrial Equipment Structure: An application to Remaining Useful Life Estimation,” *arXiv preprint arXiv:2006.16556*, 2020.

[37] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The Graph Neural Network Model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[38] T. N. Kipf and M. Welling, “Semi-supervised Classification with Graph Convolutional Networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[39] M. F. Zaeh, G. Reinhart, M. Ostgathe, F. Geiger, and C. Lau, “A Holistic Approach for the Cognitive Control of Production Systems,” *Advanced Engineering Informatics*, vol. 24, no. 3, pp. 300–307, 2010.

[40] J. Ploennigs, A. Ba, and M. Barry, “Materializing the promises of cognitive iot: How cognitive buildings are shaping the way,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2367–2374, 2017.

[41] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital Twin-driven Product Design, Manufacturing and Service with Big Data,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3563–3576, 2018.

[42] “GrabCAD.” [Online]. Available: <https://grabcad.com/>

[43] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, “A Search Engine for 3D Models,” *ACM Trans. Graph.*, vol. 22, no. 1, p. 83–105, Jan. 2003.

[44] D. Ramanujan and W. Z. Bernstein, “VESPER: Visual Exploration of Similarity and Performance Metrics for Computer-aided Design Repositories,” *International Manufacturing Science and Engineering Conference*, vol. 51371, p. V003T02A034, 2018.

[45] F. Neumann and M. Atten, “Novel Approach for Shape-based Similarity Search Enabled by 3D PDF,” *Int. J. Comput. Appl. Technol.*, vol. 58, pp. 165–173, 2018.



Trier Mortlock is a Ph.D. student in the Department of Mechanical and Aerospace Engineering at the University of California, Irvine. He is a member of the Autonomous and Intelligent Cyber-Physical Systems (AICPS) Laboratory. He received his B.S. in Mechanical Engineering from the University of California, Berkeley and his M.S. in Mechanical and Aerospace Engineering from the University of California, Irvine. He has served in the United States Army Reserve since 2018, where he is presently a Cyber Operations Officer. His research interests include cyber-physical systems, autonomous vehicle security, manufacturing system security, and the intersection of machine learning and sensor fusion.



Deepan Muthirayan is currently a Post-doctoral Researcher in the department of Electrical Engineering and Computer Science at University of California at Irvine. He obtained his Ph.D. from the University of California at Berkeley (2016) and B.Tech/M.tech degree from the Indian Institute of Technology Madras (2010). His doctoral thesis work focused on market mechanisms for integrating demand flexibility in energy systems. Before his term at UC Irvine he was a post-doctoral associate at Cornell University where his work focused on online

scheduling algorithms for managing demand flexibility. His current research interests include control theory, machine learning, topics at the intersection of learning and control, online learning, online algorithms, game theory, and their application to smart systems.



Shih-Yuan Yu received the B.S. and M.S. degrees in Computer Science and Information Engineering from the National Taiwan University (NTU) in 2014. He worked at MediaTek for 4 years. Currently he is a Ph.D. student in the University of California, Irvine. Now his research interests are about design automation of embedded systems using data-driven system modeling approaches. It covers incorporating machine learning methods to identify potential security issues in systems.



Pramod Khargonekar received B. Tech. Degree in electrical engineering in 1977 from the Indian Institute of Technology, Bombay, India, and M.S. degree in mathematics in 1980 and Ph.D. degree in electrical engineering in 1981 from the University of Florida, respectively. He was Chairman of the Department of Electrical Engineering and Computer Science (1997-2001) and the Claude E. Shannon Professor of Engineering Science at The University of Michigan. From 2001-2009, he was Dean of the College of Engineering and Eckles Professor of Electrical and Computer Engineering at the University of Florida (2001-2016). After serving as Deputy Director of Technology at ARPA-E (2012-13), he was appointed Assistant Director for the Directorate of Engineering (ENG) (2013-2016). Currently, he is Vice Chancellor for Research and Distinguished Professor of Electrical Engineering and Computer Science at the University of California, Irvine. His research and teaching interests are centered on theory and applications of systems and control.



Mohammad Abdullah Al Faruque (M’06, SM’15) received his B.Sc. degree in Computer Science and Engineering (CSE) from Bangladesh University of Engineering and Technology (BUET) in 2002, and M.Sc. and Ph.D. degrees in Computer Science from Aachen Technical University and Karlsruhe Institute of Technology, Germany in 2004 and 2009, respectively. He is an Associate Professor at University of California, Irvine (UCI) and Director of the Embedded and Cyber-Physical Systems Lab. He was with Siemens Corporate Research and Technology in Princeton, NJ as a Research Scientist. His current research is focused on the system-level design of embedded and Cyber-Physical-Systems (CPS) with special interest in low-power design, CPS security, data-driven CPS design, etc. Besides many other awards, he is the recipient of the School of Engineering Mid-Career Faculty Award for Research 2019, the IEEE Technical Committee on Cyber-Physical Systems Early-Career Award 2018, and the IEEE CEDA Ernest S. Kuh Early Career Award 2016.