Chapter 3
Artificial Intelligence for Digital Twin

Abstract Artificial intelligence (AI) is a promising technology that enables machines to learn from experience, adjust to environments, and perform humanlike tasks. Incorporating AI with digital twin (DT) makes DT modelling flexible and accurate, while improving the learning efficiency of AI agents. In this chapter, we present the framework of AI-empowered DT and discuss some key issues in the joint application of these two technologies. Then, we introduce the incorporation paradigms of three AI learning approaches with DT networks.

3.1 Artificial Intelligence in Digital Twin

AI is a branch of computer science that enables learning agents to perform tasks that typically rely on human intelligence. Nowadays, the blooming of AI technology has brought powerful capabilities in environmental cognition, knowledge learning, action decision, and state prediction to smart machines, vehicles, and various types of Internet of Things (IoT) devices.

However, despite great advancements led by AI for industry, transportation, healthcare, and other areas, AI is not always glamorous. In fact, the AI learning process consists of continuous interactions between agents and the environmental system. The agents make decisions and take actions according to the current observed environment states, and these actions then react to and change the environment states, which triggers a new round of agent learning until the process finally converges. The interactive learning approach, which relies on real physical systems, is often costly and inefficient. For instance, when applying AI directly to real vehicles to train autonomous driving policies, vehicles can cause traffic accidents. Another example is leveraging AI to optimize the operation of cellular networks. Due to the large scale of cellular networks and their many subscribers, it takes a long time for AI agents to obtain feedback on state changes after performing actions, which seriously undermines AI learning effectiveness. Incorporating AI with simulation software seems a feasible approach to speed up the system feedback for AI actions. However,

Y. Zhang, *Digital Twin*, Simula SpringerBriefs on Computing 16,
https://doi.org/10.1007/978-3-031-51819-5_3

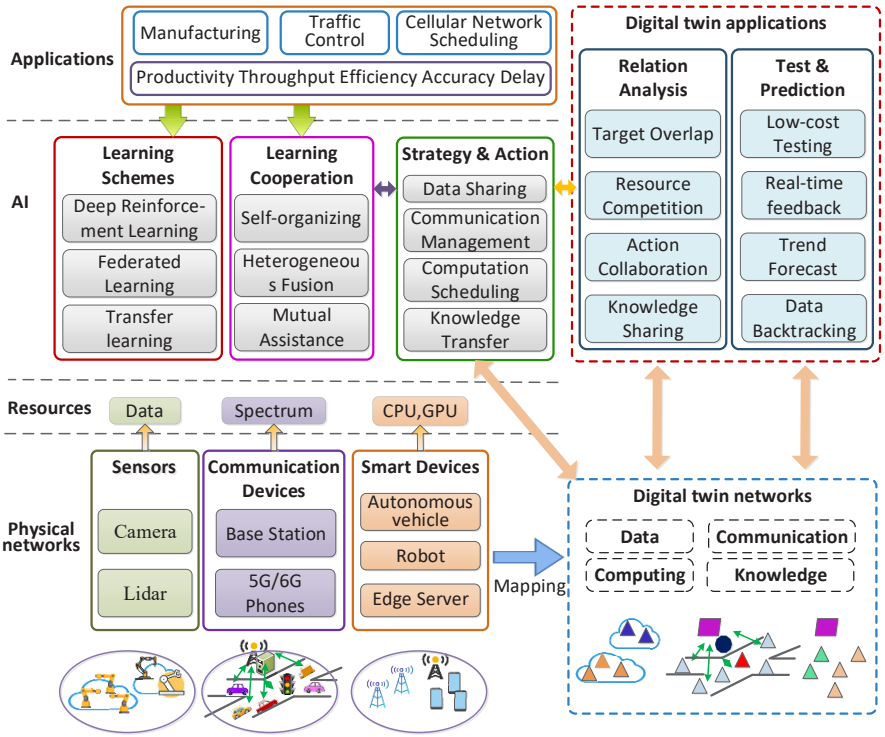

Fig. 3.1 AI-empowered DT framework

due to nonlinear factors and uncertainty, it is hard to build a high-fidelity simulation environment for a highly dynamic and complex system. Thus, the strategies and actions learned in a simulation environment cannot be directly deployed to machines in the real world.

To cope with this problem, we resort to DT technology. DT mirrors the forms, states, and characteristics of physical objects in the real world with high fidelity and real time into virtual space. This mirror model eases our cognition of complex physical systems and makes operations on virtual entities equivalent to those on physical ones. Moreover, by leveraging the precise reflection capability of DT and the intelligent adaptability of AI, the combination of DT and AI can benefit both parties. On the one hand, with the aid of DT, AI learning methods can obtain high-fidelity state information from physical objects for model training, verify the effect of the learning strategy at low cost, and implement the life cycle management of complex systems. On the other hand, AI learning can continuously monitor the accuracy of DT models, dynamically adjust the DT mapping mechanism, and maintain the consistency between virtual space and physical space.

To fully explore the benefits of incorporating AI and DT, we present the framework of AI-empowered DT shown in Fig. 3.1. This framework is mainly composed of two types of networks, namely, physical networks and DT networks. The physical

networks are composed of various types of physical devices and different types of resources served or consumed by these devices. As ubiquitous devices in the physical networks, sensors such as cameras and lidars collect the real-time state from the physical environment. The state data carry the characteristics of the real world, such as the operating conditions of industrial equipment and driving behaviour of smart vehicles, which are useful for failure detection and traffic planning. Another type of device involves communication infrastructures and user terminals, for instance, cellular radio base stations and mobile phones. The data interaction between these devices mostly adopts wireless communications, which consume spectrum resources. Thus, managing the communication equipment mainly involves scheduling of channel resources. Furthermore, in the physical network, smart devices play an important role in providing computation resources. Smart devices such as autonomous vehicles, edge servers, and robots can be equipped with very powerful CPU and GPU computing capabilities, compared to handheld user devices. For such data-intensive and computationally intensive tasks, performing local computations on user equipment can consume excessive energy and bring about long delays. Catering to this problem, these tasks can be offloaded to edge service–enabled smart devices for efficient processing.

The data, communication, and computing resources mentioned above can be scheduled to serve various types of tasks in the physical network. However, the highly dynamic topology of mobile devices and communication interference arising in the physical environment pose significant challenges to resource efficiency and application performance. More specifically, the mobility and dynamic topology of devices make environmental data collection more difficult. In addition, due to wireless interference, the received data will deviate from the sender's original data, which can lead to erroneous environmental cognition and resource scheduling decisions. To address these challenges, we turn to DT technology and formulate DT networks.

A DT network is a mapping of a physical network in a virtual space that consists of virtual twins of all the physical units on the physical side. Data, spectrum, and computing resources contained in the DT network form logical entities that can be freely decomposed and flexibly combined. In addition, the resources in the DT network include some of the knowledge and experience that have been already gained and cached, such as the channel history states and known bandwidth allocation strategy in previous radio resource management. Since the DT network operates in a virtual space, there is no interference or error in the information interaction between DT entities, and the coordination of heterogeneous resources can also break the constraints of node locations and realize resource supply and demand services between distant nodes. In addition, based on historical information and knowledge, future network status trends can be accurately predicted, thereby facilitating effective resource management.

Based on the DT networks formulated, two promising types of applications can be achieved. The first is a variety of relational analysis in complex systems and highly dynamic environments, including objective overlap testing in distributed optimization, competition for limited resources by multiple business nodes, action cooperation among multiple nodes, and knowledge sharing collaboration among

a group of machine learning agents. The other type of DT application is strategy testing and future state prediction. DT can provide low-cost policy verification in virtual space and obtain real-time result feedback. Moreover, during the forecasting process, the time axis can be easily and flexibly adjusted, allowing for efficient trend forecasting and data retrospectives.

The AI module for scheduling resources can be divided into two parts. The first part involves learning schemes to determine the architecture as well as the components of AI models, and it can be mainly classified into deep reinforcement learning (DRL), federated learning (FL), and transfer learning (TL). Among these schemes, DRL is of an architecture combining the neural networks of deep learning (DL) and the decision model of reinforcement learning (RL). Based on DRL, FL is a multi-agent DRL framework that can protect the privacy of each agent. TL is a novel concept that aims to utilize the original model to construct a new model to speed up convergence.

The second part of the AI module involves learning cooperation relationships that indicate the cooperation types between learning agents, including self-organizing, heterogeneous fusion, and mutual assistance. These relationships can be further classified into individual learning and cooperative learning. Individual learning always converges faster than cooperative learning, since it does not experience a time delay in information interaction. However, a lack of global information about a system can cause the convergence point to be suboptimal. In contrast, collaborative learning can usually achieve more accurate decision performance, but it often requires longer convergence times, especially for large-scale complex systems.

3.2 DRL-Empowered DT

3.2.1 Introduction to DRL

In earlier years, machine learning methods represented by DL and RL were widely used to solve various problems in networks. DL aims to construct deep neural networks to identify characteristics from the environment, while RL aims to take optimal actions to obtain maximal rewards. More specifically, DL enables machines to imitate human activities such as hearing and thinking and to solve complex pattern recognition problems, making great progress in AI-related technologies. RL allows agents to imitate the capacity of humans making decisions based on the current environment. However, both DL and RL have their drawbacks. For example, DL cannot explain decisions it has already made, and RL cannot identify high-dimensional states of the environment well. Combining DL with RL to design a new machine learning framework called DRL is a promising approach to address the above problem. DRL combines the perceptive ability of DL with the decision-making ability of RL. Moreover, DRL can learn control strategies directly from high-dimensional raw data, which is much closer to human learning compared to previously designed AI approaches.

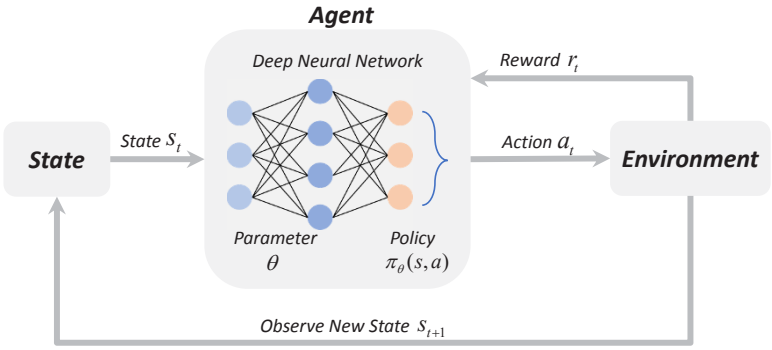

Fig. 3.2 DRL framework

Essentially, DRL is applied to sequential decision making, which can be mathematically formulated as a Markov decision process (MDP). The DRL framework is shown in Fig. 3.2. In each time slot t, the agent observes the current environment state s_t and uses its policy to select an action a_t. A policy can be considered a mapping from any state to an action. After the action a_t is performed, the environment moves to state s_{t+1} in the next time slot with transition probability $P(s_{t+1}|s_t, a_t)$. In addition, a corresponding reward $r_t = R(s_t, a_t)$ is obtained via the immediate reward function, which is the evaluative feedback of the action taken. Given a stationary and Markovian policy π, the next state of the environment, s_{t+1}, is completely determined by the current state, s_t. In this context, the current policy together with the transition probability function determines the long-term cumulative reward. Assuming $\tau = (s_t, a_t, s_{t+1}, a_{t+1}, \cdots, s_T, a_T)$ is a trajectory from an MDP, the long-term cumulative reward can be defined as

$$G(\tau) = \sum_{i=0}^{T-t} \gamma^i R(s_{t+i}, a_{t+i}), \tag{3.1}$$

where $\gamma \in (0, 1]$ is the discount factor that measures the importance of the future reward and T is the length of an episode. For a continuous MDP, we have $T \longrightarrow \infty$. In an MDP, the key issue is to find the optimal policy that maximizes the long-term cumulative reward.

3.2.2 Incorporation of DT and DRL

As a promising AI technology, DRL provides a feasible method for solving complex problems in unknown environments. However, there are still challenges to be resolved in the process of DRL learning and implementation, which are discussed below.

High cost of the trial-and-error learning process: As a zero-knowledge experimental learning method, DRL maximizes the cumulative discounted reward by learning optimal state–action mapping policies through trial and error. However, in some application scenarios, especially in traffic safety–sensitive Internet of Vehicles applications and smart medical care related to patients' lives, the cost of trial and error is too high to be acceptable.

Frequent data transmission in learning: A large amount of state data needs to be input into the DRL system to train models and draw action strategies. For example, the channel spectrum status and real-time communication requirements of users are input for radio resource scheduling. Rapid and dynamic changes in environmental status and user requirements result in intensive data transmission and frequent state updates. Furthermore, as the dimensionality of the input data increases, so too does the time taken for the learning process to reach the convergence. Thus, we find that it is difficult for the DRL method to meet the needs of delay-sensitive business scenarios such as the driving action control of autonomous vehicles and communication management in interactive multimedia applications.

Interaction barriers between multiple agents in distributed DRL: Distributed DRL uses multiple agents to obtain the optimal action policy based on the environmental status. These agents can accelerate the learning process by sharing information when collaboratively working towards a common learning target. However, when the agents use wireless communication to share learning information, wireless signal fading and spectrum interference can lead to transmission errors and retransmission, which not only cause extra communication costs, but can also undermine training efficiency and learning convergence.

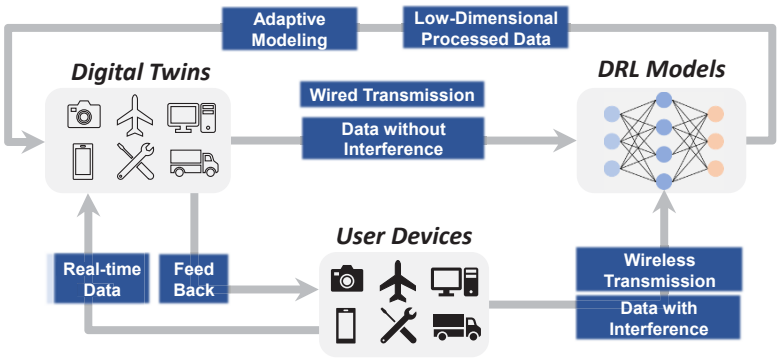

Fig. 3.3 Cooperation of DT and DRL

To address the above challenges, we turn to DT technology. Figure 3.3 illustrates how DT and DRL can cooperate to improve learning efficiency. First, since DT creates a high-fidelity virtual map of physical objects, DRL algorithms applied in the real world can be trained in the DT space. Different from the real training

process in physical space, the trial-and-error process in DT training does not have unacceptable consequences, such as damage or injury to objects or humans due to wrong decisions. Second, the agents of DRL can obtain physical system states from the DT models without relying on communications between the agents and the physical objects, reducing data transmission delays. Compared with traditional DRL implemented in the physical space, the DRL model on the DT side can be trained for more rounds per unit time and converges faster. Finally, by modelling the DT of DRL agents on DT servers, the actual information interaction between agents in the physical space can be mapped to the information sharing between DT servers or within one server in virtual space. This virtual-to-virtual agent communication enables reliable information sharing between two agents and does not consume physical communication resources.

On the DRL side, we note that the features, functions, and behaviours of physical objects are often high dimensional, making it difficult to describe them directly in the DT modelling process. With the help of DRL, these high-dimensional data are extracted and refined by neural networks into lower-dimensional data that are easier to process. Furthermore, DRL can help handle some of the unique problems of DT, such as DT placement and DT migration algorithms, and make DT technology adaptable to different time-varying environments.

Numerous recent studies have investigated the cooperation of DRL and DT. Among these works, the resource management of sixth-generation (6G) networks has attracted much attention from researchers. In [25], the authors considered the dynamic topology of the edge network and proposed a DT migration scenario. They adopted a multi-agent DRL approach to find the optimal DT migration policy by considering both the latency of updating DT and the energy consumption of data transmission. In [26], the authors proposed an intelligent task offloading scheme assisted by DT. The mobile edge services, mobile users, and channel state information were mapped into DT to obtain real-time information on the physical objects and radio communication environments. Then, a reliable mobile edge server with the best communication link quality was selected to offload the task by training the data stored in the DT with the double deep-Q learning algorithm. In [27], the authors proposed a mobile offloading scheme in a DT edge network. The DT of the edge server maps the state of the edge server, and the DT of the entire mobile edge computing system provides training data for offloading decisions. The Lyapunov optimization method was leveraged to simplify the long-term migration cost constraint in a multi-objective dynamic optimization problem, which was then solved by actor–critic DRL. This solution effectively diminishes the average offloading latency, the offloading failure rate, and the service migration rate while saving system costs with DT assistance.

DT technology and DRL can be seamlessly fused to achieve intelligent manufacturing. In [28], the authors proposed a DT- and RL-based production control method. This method replaces the existing dispatching rule in the type and instance phases of a micro smart factory. In this method, the RL policy network is learned and evaluated by coordination between DT and RL. The DT provides virtual event logs that include states, actions, and rewards to support learning. In [29], the authors proposed the automation of factory scheduling by using DT to map manufacturing

cells, simulate system behaviour, predict process failures, and adaptively control operating variables. Moreover, based on one of the cases, the authors presented the training results of the deep Q-learning algorithm and discussed the development prospects of incorporating DRL-based AI into the industrial control process. By applying the DRL method, process knowledge can be obtained efficiently, manufacturing tasks can be arranged, and optimal actions can be determined, with strong control robustness.

In addition to the above work, previous studies have applied DT and DRL to emerging applications. In [30], the authors analysed a multi-user offloading system where the quality of service is reflected through the response time of the services; they adopted a DRL approach to obtain the optimal offloading decision to address the problem of edge computing devices overloading under excessive service requests owing to the computational intensity of the DT-empowered Internet of Vehicles. In [31], the authors discussed the feedback of traditional flocking motion methods for unmanned aerial vehicles (UAVs) and proposed a DT-enabled DRL training framework to solve the problem of the sim-to-real problem restricting the application of DRL to the flocking motion scenario.

3.2.3 Open Research Issues

Although the cooperation of DRL and DT has shown great potential in some scenarios, there are still problems that warrant investigation. The first problem is resource scheduling. The volume of data of physical objects in DT is huge, and the deployment of DRL at the edge also requires computing resource services. Therefore, reducing redundant data and designing lightweight DRL models are significant issues in the combination of DT and DRL.

Another issue is environmental dynamics. The DT modelling process can involve a dynamic and time-varying environment, with a wide variety of physical objects, and the data and computing requirements required for the corresponding modelling processing can also differ. In addition, the high-speed movement of physical objects and the dynamic changes of wireless channels will further exacerbate the uncertainty of environmental characteristics. Although DRL can provide an optimal strategy for DT resource scheduling, a continuously and dynamically changing environment can seriously undermine learning efficiency. Therefore, improving the flexibility and adaptability of DRL to dynamic DT modelling is an important issue to be addressed.

3.3 Federated Learning (FL) for DT

3.3.1 Introduction to FL

The proliferation of AI learning techniques has provided unprecedented powerful applications to areas including smart manufacturing, autonomous driving, and intelligent healthcare. With these diverse AI applications, two critical challenges have emerged that must be addressed. The first challenge is learning scalability. In a system with many widely distributed nodes, using a traditional centralized AI mechanism in the learning process can generate significant amounts of data to be collected and in overhead transmission, creating a great burden on the processing capability of a few centralized agents. Another challenge centres around privacy protection. The system states or data resources gathered for learning related to factory production techniques, route navigation preferences, and an individual's personal physical condition invariably contain sensitive information, requiring a strong privacy guarantee.

FL has been widely regarded as an appealing approach to address the above challenges. FL is a privacy-protected model-training technology with an emphasis on leveraging distributed agents to collect data and leverage local training resources. Unlike centralized AI, which depends purely on the capability of a few central agents, in FL multiple geodistributed agents perform model training in parallel without sharing sensitive raw data, thus helping ensure privacy and reducing communication costs.

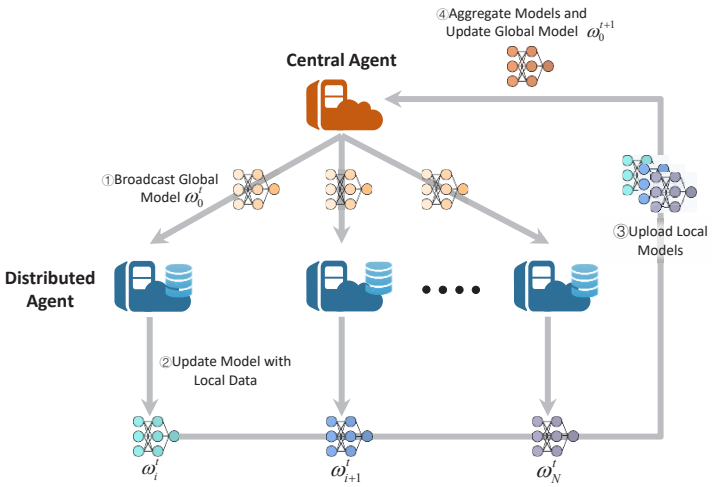

Fig. 3.4 Main flow of the FL process

Figure 3.4 shows the main flow of the FL process. First, a central agent initializes a global model, denoted as ω_0, and broadcasts this model to the other distributed

agents. Then, after each distributed agent receives ω_0, it takes locally collected data to update the parameters of this model and achieves a local model that minimizes the loss function, defined as

$$F(\omega_i^t) = \sum_{x_i \in D_i} f(\omega_i^t, x_i) \Big/ |D_i|, \tag{3.2}$$

where ω_i^t is the local model of agent i in learning iteration t, and D_i is the local data set of agent i. This loss function is used to measure the accuracy of the local model and guide the model update in a gradient descent approach, which is written as

$$\omega_i^{t+1} = \omega_i^t - \xi \cdot \nabla F(\omega_i^t), \tag{3.3}$$

where ξ is the learning step. Next, each distributed agent uploads its local model to the central agent and waits for an aggregation step, which can be written as

$$\omega_0^{t+1} = \sum_{j=1}^{N} \alpha_i \cdot \omega_i^t \Big/ N, \tag{3.4}$$

where α_i is the coefficient of agent i and N is the number of collaborating learning agents. When the aggregation is completed, the central agent will republish the updated global model to the distributed agents. The iterations repeat in this manner until the global model converges or reaches a predetermined accuracy.

3.3.2 Incorporation of DT and FL

Although FL is a promising paradigm that enables collaborative training and mitigates privacy risks, its learning operation still has several challenges and limitations.

Complexity and uncertainty of model characteristics: Large-scale dynamic systems usually have diverse features that correlate with each other, which means it is very difficult for FL to extract them from system events. Moreover, during the learning operation, unplanned events such as weather changes, traffic accidents, and equipment failures, can further confuse the training inputs and undermine model convergence.

Asynchrony between heterogeneous cooperative agents: As a distributed AI framework, FL leverages multiple geographically distributed agents to train their local models in parallel and then aggregates a parametric model in a central agent. There is heterogeneity in the training environment where each agent is located in terms of the number of physical entities, the size of the region, the frequency of event changes, and the differences in agents' processing capacity. This heterogeneity makes it hard to synchronize the aggregation of FL across multiple distributed agents. Although previous works have been devoted to the design of asynchronous FL mechanisms, most of them have improved the learning convergence at the cost

of model accuracy. How to achieve both learning efficiency and model precision is still an open question.

Interaction bottleneck between collaborative agents: Considering the distributed training and central aggregation characteristics of FL, frequent interactions are required between the client agents and the central agent, especially for learning systems with high-dimensional feature parameters and highly dynamic environments. In such a case, where wireless communications are used to realize the interactions between agents, the efficiency of local model aggregation and global model distribution can be severely undermined due to the data transfer bottleneck caused by the limited wireless spectrum and disturbed

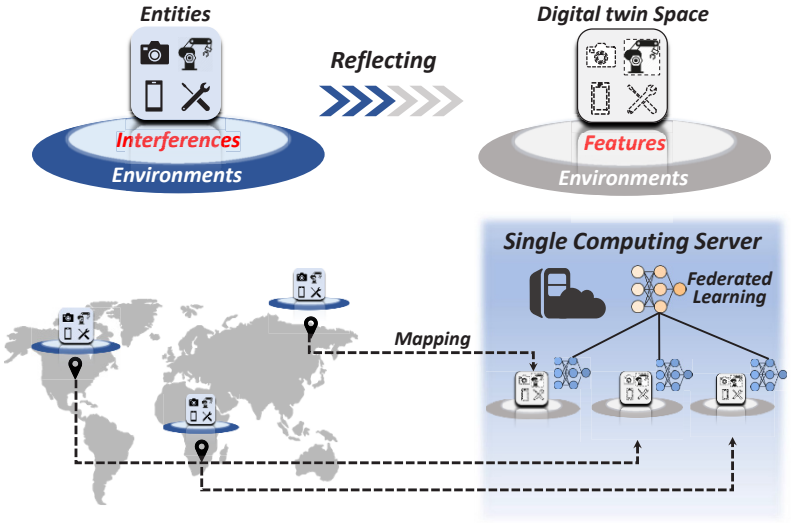

Fig. 3.5 Benefits of applying DT in FL

To address the above challenges, we turn to DT technology. Figure 3.5 illustrates the benefits of applying DT in FL. First, reflecting complex physical entities and environments into DT space can eliminate unnecessary interference factors, thereby helping FL to mine the core features of the system and further explore their interrelationships. Second, for the problem of asynchronous heterogeneous training regions, using a mirrored virtual environment built by DT to replace all or part of the regional systems affected by slow response can greatly improve these regions' local model convergence speeds. The training between regions is thus synchronized, and both learning efficiency and accuracy can be achieved. Finally, the DT mappings of multiple regions can be constructed on a single computing server, and the real data communications between the agents located in different regions in the physical space can be mapped to the interactions between multiple learning processes in the virtual

space. Therefore, DT can free the collaborative agents in FL from the constraints of physical communication resources.

We note that the many benefits provided by DT to FL depend on the ability of the twin models in the virtual space being able to map physical entities and networks accurately and in real time. Due to the potential dynamics of physical networks, the DT mapping strategy needs to be adjusted accordingly. Considering the large-scale and distributed characteristics of the physical entities, using FL to optimize the mapping strategy seems an appealing approach. More specifically, in the integration of DT and FL, DT mapping accuracy can be included as an element of the learning reward, and the parameters of the DT mapping strategy can be added to the learning action space.

Recently, research attempts have focused on applying DT with FL. Among these works, the Industrial IoT (IIoT), which enables manufacturers to operate with massive numbers of assets and gain insights into production processes, has turned out to be an important application scenario. In [32], the authors intended to improve the quality of services of the IIoT and incorporated DT into edge networks to form a DT edge network. In this network, FL was leveraged to construct IIoT twin models, which improves IIoT communication efficiency and reduces its transmission energy cost. In [33], the authors used DT to capture the features of IIoT devices to assist FL and presented a clustering-based asynchronous FL scheme that adapts to the IIoT heterogeneity and benefits learning accuracy and convergence. In [34], the authors focused on resource-constrained IIoT networks, where the energy consumption of FL and digital mapping become the bottleneck in network performance. To address this bottleneck, the authors introduced a joint training method selection and resource allocation algorithm that minimizes the energy cost under the constraint of the learning convergence rate.

In preparation for the coming 6G era, DT technology and FL can be seamlessly fused to trigger advanced network scheduling strategies. In [9], the authors presented an FL-empowered DT 6G network that migrates real-time data processing to the edge plane. To further balance the learning accuracy and time cost of the proposed network, the authors formulated an optimization problem for edge association by jointly considering DT association, the training data batch size, and bandwidth allocation. In [35], the authors applied dynamic DT and FL to air–ground cooperative 6G networks, where a UAV acts as the learning aggregator and the ground clients train the learning model according to the network features captured by DTs.

In the area of cybersecurity, blockchain has emerged as a promising paradigm to prevent the tampering of data. Since both the ledger storage of blockchain and the model training process of FL are distributed, blockchain can be introduced into DT-enabled FL. In [36], the authors utilized blockchain to design a DT edge network that facilitates flexible and secure DT construction. In this network, a double auction–based FL and local model verification scheme was proposed that improves the network's social utility. In [37], the authors proposed a blockchain-enabled FL scheme to protect communication security and data privacy in digital edge networks, and they introduced an asynchronous learning aggregation strategy to manage network resources.

In addition to the above work, previous studies have applied DT and FL to emerging applications. In [38], the authors used the COVID-19 pandemic as a new use case of these two technologies and proposed a DT–FL collaboratively empowered training framework that helps the temporal context capture historical infection data and COVID-19 response plan management. In [39], the authors applied these two technologies to edge computing–empowered distribution grids. A DT-assisted resource scheduling algorithm was proposed in an FL-enabled DT framework that outperforms benchmark schemes in terms of the cumulative iteration delay and energy consumption.

3.3.3 Open Research Issues

The incorporation of FL with DT is a promising way to improve learning efficiency while guaranteeing user privacy. However, there are still unexplored questions in the joint application of these two technologies. The first question worth investigating is the operation matching between DT and FL. The training process of FL requires many iterations, which consume massive computing resources and generate a certain time delay. Since DT modelling also depends on intensive computation, competition for resources arises between DT and FL. Effective resource scheduling is thus a critical research challenge. Moreover, the key advantage of DT is the ability to accurately map the physical world into virtual space in real time. When using FL to improve DT modelling accuracy, how to make the slow iterative learning direct the DT mapping strategy in a timely manner is still a problem for future research.

Another unexplored question concerns privacy. To reflect physical systems and objects fully and accurately, DT modelling inevitably needs to extract massive amounts of system data and user information, which can lead to privacy leakage. On the other hand, the use of FL is an attempt to protect users' private information. How to ensure privacy protection while improving the accuracy of DT modelling is also a challenge to be addressed.

3.4 Transfer Learning (TL) for DT

3.4.1 Introduction to TL

In traditional distributed intelligence networks, multiple machine learning agents equipped on edge servers, smart vehicles, and even powerful IoT devices, work independently. In some application scenarios, multiple agents in similar environments can learn with the same goal. If these agents start training at different times, agents that start later may learn their strategies from scratch. A complete training process always incurs a great deal of resource consumption and long training delays,

posing a critical challenge for resource-constrained devices serving delay-sensitive computing tasks.

TL, which is a branch of AI with low learning costs and high learning efficiency, provides a promising approach to meet these challenges. Unlike the traditional machine learning agent that tries to learn a new mission from scratch, a TL agent receives prior knowledge from other agents that have performed similar or related missions, and then starts learning with the aid of this knowledge, thus achieving faster convergence and better solutions.

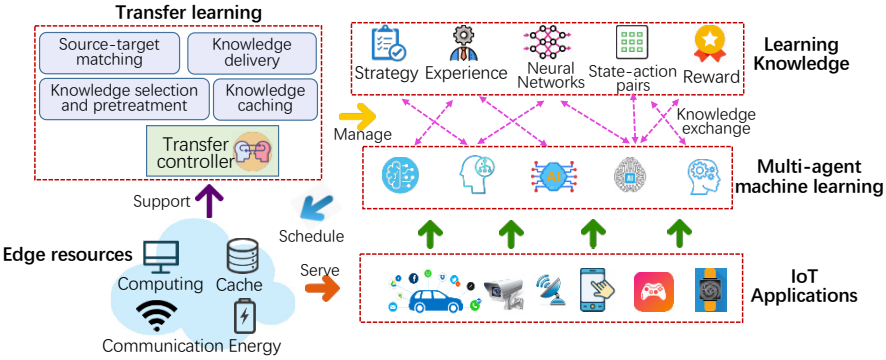

Fig. 3.6 TL framework

Figure 3.6 illustrates the TL framework. At the bottom of this figure are shown various types of modelling training and strategy learning tasks generated by IoT devices. Multiple agents with TL capabilities are deployed to handle these tasks. We note that FL-inspired learning is a gradual process that consists of continuous environment awareness, constant action exploration, and persistent strategy improvement. As the learning proceeds, valuable knowledge, such as neural network parameters, state–action pairs, action exploration experience, and the evaluation of existing strategies, is generated and recorded. This knowledge not only is the basis for the learning of the local agent in subsequent stages, but also can be shared with other agents, which can then jump directly from the initial learning stage, without any experience, to an intermediate stage with certain prior knowledge.

In the FL framework, a transfer controller module manages the sharing process, including the pairing of the transfer source and target agents, knowledge building and pretreatment, the knowledge data delivery, and the caching among the agents. It is worth noting that edge resources play a vital role in the FL framework. On the one hand, these resources can serve in IoT applications, such as vehicular communications, popular video caching, and sensing image recognition, while multi-agent machine learning is leveraged for resource scheduling. On the other hand, we resort to TL to improve machine learning efficiency and reduce scheduling time costs. However, the knowledge sharing process can create the need for extra communication, computing, and cache resources. Thus, there exists a trade-off in resource

allocation, that is, whether to use the resources to directly enhance IoT application performance or for learning efficiency improvement and service delay reduction.

TL can offer many benefits in multi-agent distributed learning scenarios, the main advantage being the reduction in training time of the target agent of the knowledge sharing process. The shared prior knowledge can effectively guide the agent to quickly converge to and reach optimal action strategies without time-consuming random exploration. In addition, TL can save training resource consumption. Each training step requires analysis and calculation. A faster training process means fewer steps, as well as lower computing and energy resource consumption. Moreover, for machine learning approaches that record large amounts of state–action pairs, the reduced training process provided by the TL also reduces the record sizes, thereby saving on cache resources.

3.4.2 Incorporation of DT and TL

Despite the benefits provided by TL, unaddressed challenges remain in TL scheme implementation, especially in application scenarios with multiple associated heterogeneous agents. Due to the associations between such agents, multiple TL node pairs can be formed. Thus, the first challenge is the choice of transferring source when the target mission has multiple potential knowledge providers. For example, when multiple UAVs are agents in training terrain models based on sensing data, these UAVs hover and cruise at different altitudes and can have overlapping or even the same modelling area. The beneficial prior knowledge of a UAV agent performing a learning mission can exist in multiple neighbouring UAVs. Source determination is a prerequisite before the learning implementation. However, it is difficult to determine the appropriate transferring pairs solely according to the physical characteristics and superficial associations in the physical world. Another challenge is what knowledge should be transferred. The prior knowledge learned by heterogeneous agents can take various forms and provide diverse learning gains between different transferring pairs. Knowledge selection and organization are the basis of effective TL. However, since knowledge is an abstract concept, it is hard to measure and schedule it accurately in physical space.

Incorporating DT with TL is a feasible approach to address the above challenges. In terms of the effect of DT on TL, by leveraging the comprehensive mapping ability of DT from a physical system to virtual space, multi-agents' environmental characteristics, neural network structure, and learning power, as well as their current training stages can be clearly presented in a logical form. This logical representation allows the TL scheduler to find optimal TL source–destination agent pairs based on the similarity of environmental features or the matching of knowledge supply and demand. Moreover, DT models existing in the virtual space are suitable for describing the knowledge attributes acquired by each agent. For example, knowledge can be logically represented as a tuple DT model composed of an owner, information items, the application scope, transfer gains, transfer costs, and other elements.

From the perspective of the role played by TL in the DT process, especially in scenarios of distributed multi-DT models, TL can share the construction experience of the completed DT model, such as the model structure, constituent elements, and update cycle, with the DT models that have been or have yet to be formed. This knowledge transfer scheme greatly shortens DT construction delays and improves DT model accuracy. Moreover, since DT processes consume considerable communication and computing resources, TL can also be used in several similar DT environments to reuse resource scheduling strategies.

TL has been used in many areas to improve the efficiency of distributed learning. For instance, in [40], the authors proposed a deep uncertainty–aware TL framework for COVID-19 detection that addresses the problem of the lack of medical images in neural network training. In [41], the authors introduced a TL-empowered aerial edge network that uses multi-agent machine learning to draw optimal service strategies while leveraging TL to share and reuse knowledge between UAVs to save on resource costs and reduce training latency. In [42], TL was used in action unit intensity estimation, where known facial features were inherited in new estimation scenarios at minimal extra computational cost.

Along with the development of DT technology, a few studies have been dedicated to the incorporation of DT and TL. In [43], the authors focused on anomaly detection in dynamically changing network functions virtualization environments. They used DT to measure a virtual instance of a physical network in capturing real-time anomaly–fault dependency relationships while leveraging TL to utilize the learned knowledge of the dependency relationships in historical periods. In [44], the authors introduced a DT and deep FL jointly enabled fault diagnosis scheme that diagnoses faults in both the development and maintenance phases. In this scheme, the previously trained diagnosis model can be migrated from virtual space to physical space for real-time monitoring. Considering that DT models are usually customized for specific scenarios and could lack sufficient environmental adaptability, the authors in [45] leveraged TL to explore an adaptive evolution mechanism that improves remodelling efficiency under the premise of limited environmental information.

3.4.3 Open Research Issues

As recent emerging technologies, DT and TL, as well as their incorporation, still have open research issues to be explored. The first issue concerns knowledge transfer between heterogeneous training models. Training models can differ among TL agents, in terms of their learning methods, neural network structures, and knowledge cache organization. Although DT can describe these training models logically and consistently in virtual space, during TL implementation, how to preprocess and match the knowledge between source and target agents to improve the transfer effect is still a key challenge.

The second issue involves resource scheduling in DT-empowered TL. Various types of resources play a key role in TL for knowledge data delivery, storage, and

processing, and DT's model building and updating also consume these resources. Competition for constrained resources can thus take place during cooperation between FL and DT. How to coordinate resource scheduling between the two and improve the efficiency of knowledge transfer while ensuring modelling accuracy is therefore also a key question to be addressed.

Finally, an issue to be considered is DT construction that adapts to TL operations. TL usually occurs between multiple agents distributed in a large-scale system, whereas DT systems always construct models on a small number of centralized servers. How to solve the contradiction between the distributed architecture of TL and the centralized construction of DT requires further exploration.