electronics

MDPI

Review

Distributed Machine Learning and Native AI Enablers for End-to-End Resources Management in 6G

Orfeas Agis Karachalios [1,2,*], Anastasios Zafeiropoulos [2], Kimon Kontovasilis [1] and Symeon Papavassiliou [2]

[1] National Center for Scientific Research Demokritos, Institute of Informatics and Telecommunications, 153 10 Athens, Greece; kkont@iit.demokritos.gr
[2] School of Electrical and Computer Engineering, National Technical University of Athens, 157 80 Athens, Greece; tzafeir@cn.ntua.gr (A.Z.); papavass@mail.ntua.gr (S.P.)
* Correspondence: okarachalios@iit.demokritos.gr

Abstract: 6G targets a broad and ambitious range of networking scenarios with stringent and diverse requirements. Such challenging demands require a multitude of computational and communication resources and means for their efficient and coordinated management in an end-to-end fashion across various domains. Conventional approaches cannot handle the complexity, dynamicity, and end-to-end scope of the problem, and solutions based on artificial intelligence (AI) become necessary. However, current applications of AI to resource management (RM) tasks provide partial ad hoc solutions that largely lack compatibility with notions of native AI enablers, as foreseen in 6G, and either have a narrow focus, without regard for an end-to-end scope, or employ non-scalable representations/learning. This survey article contributes a systematic demonstration that the 6G vision promotes the employment of appropriate distributed machine learning (ML) frameworks that interact through native AI enablers in a composable fashion towards a versatile and effective end-to-end RM framework. We start with an account of 6G challenges that yields three criteria for benchmarking the suitability of candidate ML-powered RM methodologies for 6G, also in connection with an end-to-end scope. We then proceed with a focused survey of appropriate methodologies in light of these criteria. All considered methodologies are classified in accordance with six distinct methodological frameworks, and this approach invites broader insight into the potential and limitations of the more general frameworks, beyond individual methodologies. The landscape is complemented by considering important AI enablers, discussing their functionality and interplay, and exploring their potential for supporting each of the six methodological frameworks. The article culminates with lessons learned, open issues, and directions for future research.

Keywords: 6G networks; resource management; end-to-end; network management and orchestration; distributed machine learning; reinforcement learning; federated learning; pervasive AI; native AI enablers

Citation: Karachalios, O.A.; Zafeiropoulos, A.; Kontovasilis, K.; Papavassiliou, S. Distributed Machine Learning and Native AI Enablers for End-to-End Resources Management in 6G. *Electronics* **2023**, *12*, 3761. https://doi.org/10.3390/electronics12183761

Academic Editor: Christos J. Bouras

Received: 26 July 2023
Revised: 1 September 2023
Accepted: 4 September 2023
Published: 6 September 2023

1. Introduction

1.1. Background and Motivation

It is expected that 6G will provide support for networking scenarios of unprecedented breadth, ambition, and complexity. Relevant use cases encompass Industry 4.0, healthcare, holographic communications, virtual reality and meta-verse applications, vehicular networks, and haptic communications, all of which present stringent and diverse requirements in terms of bandwidth, latency, reliability, connectivity, security, and privacy while being computationally intensive [1]. Moreover, the association of 6G with a paradigm shift towards the combined provision of connectivity, computational capabilities, and ubiquitous intelligence [1–3] creates an even more challenging landscape that calls for the ubiquitous availability of a multitude of heterogeneous resources and means for their efficient and coordinated management in an end-to-end fashion across various relevant domains or resource clusters with complicated dependencies [1,4]. Conventional algorithmic approaches

fall short in handling the complexity, dynamicity, and end-to-end scope of the problem, and solutions based on AI become necessary. In fact, it is envisaged that such AI-based management solutions will be integrated with native facilities in the network architecture, and this vision poses additional challenges, not only in terms of the required integration, but also with respect to the structure of the AI models themselves [1,5,6].

Exploiting AI for network management in general and resources management (RM) in particular is not a 6G novelty, as research in this direction has already made headway in the context of 5G networks; see, e.g., [4,7–9] and references therein. Relevant examples include AI-based solutions for spectrum management, power allocation, user association, beamforming, mobility management, cache management, virtual network function (VNF) placement, and end-to-end RM and orchestration. Various methodologies using ML have been adopted, from supervised learning (SL) to reinforcement learning (RL), according to learning frameworks exhibiting a varying degree of centralization. However, most approaches proposed to date are tailored to particular use cases, frequently focusing on a narrow set of resources and/or network domains, without regard for an end-to-end scope. Also, many approaches employ representations/learning that cannot scale to address the greater complexity, diversity, and sophistication of 6G. Finally, the current approaches give little or no consideration to how they function in the context of network-native AI-enabling facilities.

1.2. Contributions

In view of the shortcomings just mentioned, there is a need for reexamining and reassessing AI-based methodologies for RM proposed in the literature, seeking new insight along a combined viewpoint that jointly considers the following: the structure and effectiveness of candidate AI-based solutions for RM; the suitability of these solutions for use as building blocks towards managing resources end-to-end; the potential of the solutions for addressing the greater challenges posed by the 6G vision; the degree to which the solutions are capable for functioning in the context of emerging network-native AI-supporting facilities. This is the approach we take in this survey article, with the aim of providing a systematic demonstration that the 6G vision promotes the paradigm of employing appropriate distributed ML frameworks that interact through native AI enablers in a highly composable fashion towards a versatile and effective end-to-end RM framework.

We start with a concise account of 6G challenges in connection with RM, considering aspects relevant to specific types of resources and network domains, as well as aspects of a more general nature. The account of challenges gives rise to the formulation of three criteria that can be used as benchmarks for the suitability of candidate AI-based RM methodologies in the context of 6G and for their potential as parts of end-to-end RM mechanisms.

Subsequently, we exploit the gained insight in the course of a focused literature survey on RM methodologies leveraging AI. We concentrate our attention on a subset of methodologies that are more pertinent to the 6G vision based on their distributed decision-making capability, a feature that matches with the disaggregated characteristics of 6G networks and has an inherently greater potential for scalability. A second scope-defining characteristic arises from the orientation of the article towards RM tasks and the associated need for closed-loop control and automation [4], which naturally points to RL-based methodologies. Furthermore, in the interest of introducing additional structure in the survey, we classify the distributed RL-based methodologies in accordance with six distinct distributed RL frameworks (two of which also employ federated learning (FL)) that exhibit varying degrees of interaction between the decision-making units. By leveraging these frameworks, the literature survey does not only identify the strengths and weaknesses (with respect to the aforementioned three criteria) of the individual methodologies reviewed, but also highlights the potential and limitations of the frameworks themselves, in terms of managing different types of resources or managing resources end-to-end. The comparison between frameworks brings forward certain trade-offs that are discussed.

To complement the combined viewpoint targeted by the article, we then turn to network-native AI enablers that are part of the 6G vision. Specifically, we consider three complementary enablers: the AI plane that establishes AI workflows to manage data collection and status monitoring and perform AI model training, planning, and AI model deployment to the network [1,10]; knowledge reuse facilities that allow for model libraries, model selection, and transfer learning [11]; and digital twin networks (DTNs), that will enable network performance prediction and model training [12,13]. We describe how these enablers support the function of AI models in general, but also, importantly, discuss the potential of the enablers and the implications involved when they provide support for each of the six aforementioned distributed RL frameworks. Finally, we exploit the insight gained from all previous steps towards discussing lessons learned and identifying open issues and promising directions for future research.

Recapitulating, the article makes the following contributions:

- It provides a concise account of 6G challenges that yields three criteria for benchmarking the suitability of candidate ML-powered RM methodologies for 6G, also in connection with an end-to-end scope.
- Through a focused literature survey, it reviews distributed RL-based methodologies for RM, evaluating them with respect to the three criteria. The considered methodologies are categorized into six methodological frameworks, and this approach invites broader insight into the potential and limitations of the more general frameworks, beyond individual methodologies.
- It considers three important network-native AI enablers that are part of the 6G vision, discussing their functionality and interplay and exploring their potential for supporting each of the six methodological frameworks.
- It exploits the insight gained from previous steps towards discussing lessons learned and identifying open issues and promising directions for future research.

1.3. Related Work

To the best of the authors' knowledge, this is the first time that the combined consideration of the issues targeted by the paper is pursued in its entirety and through a unified perspective and methodology. Nevertheless, there are a few earlier works addressing some aspects of end-to-end RM in conjunction with some notion of AI enablers, often coming under the label of pervasive AI (PAI), and we now comment upon such related work. Publication [11] discusses end-to-end RM in the narrower context of slicing, and transfer learning is examined as an enabler for the application of ML in next-generation networks. A taxonomy of AI methodologies is provided, together with a discussion of the applicability and limitations associated with these methodologies. However, neither the interactions between the operating infrastructure and a knowledge reuse mechanism nor other AI enablers are discussed. Works [14,15] provide reviews of end-to-end distributed intelligence from the "network-for-AI" viewpoint, targeting next-generation networks. This aspect of distributed intelligence focuses on how the network technologies can support AI applications running on the network, as opposed to the "AI-for-network" viewpoint, adopted here, that focuses on how AI can be leveraged for managing the network itself. However, the notion of native AI enablers is common in both viewpoints, and these surveys provide a complementary perspective to this article, discussing certain things, such as distributed inference and computation and communication co-design. In [16], the interaction between entities virtualizing the network and its users with pervasive intelligence is discussed. DTNs are presented as an intermediate component between infrastructure operation and network control powered by AI. Additionally, composability of AI solutions is discussed. However, transfer learning and AI workflows are not considered as part of the framework. Finally, publication [17] provides a very detailed examination of distributed algorithms in the context of IoT applications and explores trade-offs relating to their applicability but does not discuss the interplay with specific AI enablers.

Differently from these previous works, we adopt a more holistic approach that exploits a review of 6G challenges in connection with RM, also considering an end-to-end scope, towards formulating criteria for assessing the suitability of candidate AI-powered methodologies for managing resources in 6G, as well as for narrowing down the space of candidate methodologies. Moreover, when reviewing the methodologies in scope, we proceed by classifying them in accordance with a few distinct methodological frameworks, and this approach invites broader insight about the potential and limitations of the more general frameworks, beyond the particular methodologies reviewed. Finally, this approach is carried over to the consideration of the network-native AI enablers, which are treated comprehensively, but also in connection with their potential for supporting each one of the methodological frameworks.

A summary of the previous discussion on related work appears in Table A1 within Appendix A.

1.4. Paper Structure

The rest of the article is structured as follows: Section 2 provides the account of 6G challenges. Section 3 formulates the three benchmarking criteria, introduces the six methodological frameworks for the classification of the RM methodologies, and proceeds with the review of the methodologies in scope. Different types of resources are treated in individual subsections, always in association with the same set of methodological frameworks, while another subsection explores unifying and distinguishing features across resource types and domains and discusses relevant trade-offs. Section 4 considers the three AI enablers, discusses their functionality and interplay, and explores their potential for supporting each of the six methodological frameworks employed in the review. Section 5 exploits all gained insight towards discussing lessons learned and identifying open issues and promising directions for future research. Finally, Section 6 concludes. The main text is supplemented with two appendices. Appendix A includes two tables, with one summarizing the discussion about related work in Section 1.3 and the other briefly describing the content of survey articles referred to in the text and addressing relevant aspects in a different context. Appendix B contains a very brief account of ML preliminaries. Only an elementary coverage is provided, with the sole purpose of making the article more self-contained and accessible to a wider readership. For additional details and a deeper coverage, the reader is directed to other sources of information.

2. Challenges Relating to RM in 6G

The 6G vision poses unique challenges to the management and operation of the network infrastructure due to stringent application requirements, huge scale, high dynamics, and an increasing tendency for disaggregation [3]. Indeed, challenging applications were already a driver in shaping 5G, giving rise to a classification into enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC), and massive machine-type communication (mMTC). In 6G, the relevant challenges are expected to be significantly greater [1]. For example, virtual reality applications will demand extremely high transmission bandwidth and computing resources while operating in real-time [18]. Vehicular networks will require high connectivity and reliability with their operation being informed by AI technologies [18]. In general, computational capacity and intelligence provision under conditions of low latency will be necessary for 6G applications. As a consequence, computational resources and AI services (catering to the needs of applications) will be hard to host in the core network or cloud and should be disaggregated towards the edge instead. More generally, deployment and management of services over distributed resources across the computing continuum will be required [19].

To cope with the so-called communication–computation convergence [1], heterogeneous resources from different domains will have to be jointly allocated and managed, which is hard to model when considering the associated complex interactions. Part of the difficulty is due to the increased problem space dimensionality when multiple domains

are considered jointly. Another part stems from a higher requirement for the coordinated collection of analytics and for detailed status exchanges across domains, both of which need to be sufficiently frequent to accommodate the anticipated highly dynamic conditions, and from the latency constraints associated with these operations. These factors increase the complexity of RM, and appropriate network support will be required to manage resources in a coordinated fashion. Equally importantly, complexity also arises due to the scale of the network [3,20] and to new technologies that gradually become part of the infrastructure, in addition to the heterogeneity and spatial distribution of resources. To deal with the increasing complexity, it is generally accepted that AI will be used to manage the network itself apart from being offered as a service to applications [1,3,21,22]. AI will become part of the service-based architecture, being available at every level of the network hierarchy, from edge to core. A new AI plane will be introduced [1,3,23], in addition to the user and control planes, with the purpose of providing services to consumers, both applications and network management entities. It is envisaged that 6G networks and AI technologies will support the transition from connected things to connected intelligence [24].

Given this context, each combination of resource and domain has its own challenges to consider. Radio resources in the radio access network (RAN) are distributed between access points and are scarce, while at the same time having to satisfy stringent requirements from all types of applications. In addition, new air interface technologies will present an extremely dynamic environment, that, while providing higher capabilities, will be hard to model and optimize [20]. Power allocation is coupled to Radio RM (RRM) and presents additional challenges, such as energy efficiency for Internet of Things (IoT) devices [4,25]. Therefore, multiple potentially competitive objectives must be optimized concurrently in the highly distributed RAN, and it is expected that AI algorithms will be the solution. Data collection and online training are necessary for these to function and present new challenges in terms of communication and computational costs. Caches at the edge are also a scarce resource that must be smartly allocated. Additionally, privacy concerns arise since user preferences must be collected and analyzed for models to make predictions. Implementing proactive edge caching faces challenges associated with the distributed and localized character of the edge network, where multiple edge servers must coordinate in order to achieve better quality of experience (QoE) and reduced traffic [4].

As mentioned earlier, computational resources will be vital for future applications, and these resources are expected to be available both at the edge and core of the network [1,3,9]. VNFs and containerized network functions (CNFs) will provide processing capacity to users. Generated tasks will need to be offloaded to the appropriate edge location to achieve low latency and high resource utilization, while more computationally intensive tasks will be sent to the core network, which creates a highly dynamic environment together with the large number of tasks [4]. User mobility necessitates quick service migration and minimal communication overhead to achieve high reliability in applications, such as vehicular networks, while the heterogeneity of edge nodes and limited capacity complicates task offloading further [26]. All these factors complicate the network management and orchestration (NMO) of edge VNFs/CNFs, which will have to make decisions in a decentralized fashion to avoid large delays and traffic congestion while minimizing communication and energy costs.

Apart from intra-domain RM, coordination and cooperation across domains is expected. Intelligent solutions to determine the best trade-off between performance, isolation, and efficiency become necessary due to the scale of the network. Data governance will also constitute a big challenge, as data need to become available within each domain and across domains for consumption by management functions [1,4]. Finally, the AI plane will need to be established for AI to be functional at every domain. The AI workflow set up by the AI plane includes the planning, optimizing, deploying, monitoring, and maintaining of AI models on the network infrastructure [1].

In the next section, we exploit the insight gained from the consideration of the aforementioned challenges towards a focused survey of candidate ML-powered RM methodologies and their assessment in terms of suitable benchmarking criteria.

3. Distributed ML for RM in 5G and Beyond

3.1. Overview, Benchmarking Criteria, and Methodological Frameworks

Research efforts on using AI for network RM problems have been increasing steadily in the past few years, tackling problems from the RAN to the core. Relevant methodologies leverage both more conventional ML, as well as deep learning (DL). For an overall account of employing such methodologies to network management and optimization tasks and for the introduction of relevant concepts, the reader may consult, e.g., [7,25,27–30] (Table A2 in Appendix A provides some further context about each of these references). Here, our aim is to concentrate on a subset of methodologies that are more pertinent to the 6G vision, in alignment with the challenges discussed in the previous section. Specifically, using as a starting point existing surveys, we examine the targeted subset of methodologies and assess their suitability for 6G in terms of the following benchmarking criteria:

(a) Whether they are scalable to larger topologies [1] (**scalability**);
(b) Whether they can become a part of a larger end-to-end framework for RM [4,14,16] (**composability/modularity**);
(c) Whether their accumulated knowledge can be extracted and reused [11] (**transferability**).

The criterion (a) addresses aspects relating to how much the communication costs associated with training increase at larger scales, how much convergence is slowed down as the scale grows, and whether the ML models of the methodology must be redesigned and retrained from scratch to work with larger topologies. The criterion (b) assesses whether the methodology can be aware of and operate jointly with other algorithms managing other aspects of the system, either in other domains or in different parts of the same domain. Finally, the criterion (c) examines whether the information from trained models (such as collected experiences or neural network parameters) can be reused. As we will see later, these criteria are also relevant to the potential of a methodology for operating in the context of network-native AI facilities.

With respect to the targeted subset of methodologies, we focus our attention on methodologies with a distributed decision-making capability, a feature that matches with the increasingly disaggregated characteristics of 6G networks and has an inherently greater potential for scalability (cf. criterion (a) above). Indeed, distributed methodologies provide many benefits in the context of 6G, as agents may be installed in separate network nodes and configure the network locally, allowing for distributed control. Compared to a fully centralized unit that manages clusters of nodes, they provide low latency in decision making and, in many cases, reduce communication costs by transmitting only processed forms of data. Because they reduce the size of the control problem, convergence becomes faster at a viable cost to optimality. When considering AI workflows (discussed in Section 4), it is easier to replace an agent in control of a smaller area of the network (with a favorable impact with respect to criterion (b)). Finally, they are more lightweight than centralized models, so they can be hosted in hardware with limited capacity.

Another scope-defining characteristic arises from the orientation of the article towards RM tasks and the associated need for closed-loop control and automation. This naturally points to RL-based methodologies; however, using a combination of SL or unsupervised learning (UL) in conjunction with a decision algorithm is also possible. Still, closed-looped control that must account for the long-term impact of decisions is more amenable to RL when compared to SL or UL. Moreover, when SL or UL are complemented by a decision-making algorithm, the latter encapsulates a fixed set of rigid modeling assumptions, while RL is model-free. In addition, such solutions rely on the stationarity of the data distribution, being less suitable for a dynamic 6G environment. In general, the stationarity assumption affects RL too, but given the focus on distributed methodologies, one considers multi-agent RL (MARL), which can deal with non-stationary environments.

In view of the aforementioned considerations, in the following we consider distributed RL methodologies. FL is also examined, as it is a distributed approach orthogonal to RL and can be combined with it, resulting in federated RL (FRL). For other types of approach, the reader may consult, e.g., [4,7–9,31,32] (Table A2 in Appendix A provides some further context about each of these references). Briefly, distributed RL methodologies introduce multiple agents to manage the network resources, with each one performing actions on the network to maximize a long-term reward in a cooperative or competitive fashion. FL includes the following steps: models are trained on distributed nodes using local data and each one communicates the learned model to a central entity. There, models are aggregated, and the updated model is pushed back to the distributed nodes.

In the interest of introducing additional structure in the subsequent review, and motivated by relevant considerations in [17], we classify the distributed RL-based methodologies in the following six distinct frameworks, according to differences between methodologies in their architecture, training, and execution model, as well as to how decision-making units/agents collaborate:

1. Centralized training, decentralized execution (**CTDE**), where actors collect data from the environment and a centralized critic is trained with the data collected. This may allow for communication between agents, considering interactions explicitly. Often, the critic is discarded after training, but this disables the potential for any further online training. See Figure 1a.
2. Fully decentralized communicating agents (**FDC**). Agents are trained locally on each node and information is exchanged in some form between agents, such as the state and action of other agents or a global reward. Learned communication is another variant, where agents learn what sort of information to transmit. See Figure 1b.
3. Fully decentralized independent agents (**FDI**). Agents do not exchange information and each agent trains independently from the others. While this reduces communication costs, it does not consider interactions explicitly, which may lead to suboptimal solutions. See Figure 1c.
4. Horizontal federated RL (**HFRL**) trains models that share the same state and action space but that collect data from different nodes. It may allow for interactions between agents. See Figure 1d.
5. Vertical federated RL (**VFRL**) can provide flexibility in state and action space structure. Essentially, agents can be heterogeneous and have different action spaces. It may also allow for interactions between agents. See Figure 1e.
6. Team learning (**TeamL**) allows for the formation of heterogeneous agents into teams that complement each other by doing different types of tasks. Agents within a team share information and reward signals. This allows for a certain degree of composability. See Figure 1f.

For a brief summary of the interactions between entities in these frameworks, see also the first two columns of Table 1 in Section 3.6.

In the following subsections we review the application of distributed RL methodologies to RM tasks, in connection with different types of resources. By maintaining associations with the aforementioned frameworks, the review does not only identify the strengths and weaknesses of the individual methodologies considered, but also highlights the potential and limitations of the more general frameworks, in terms of managing different types of resources or managing resources end-to-end.

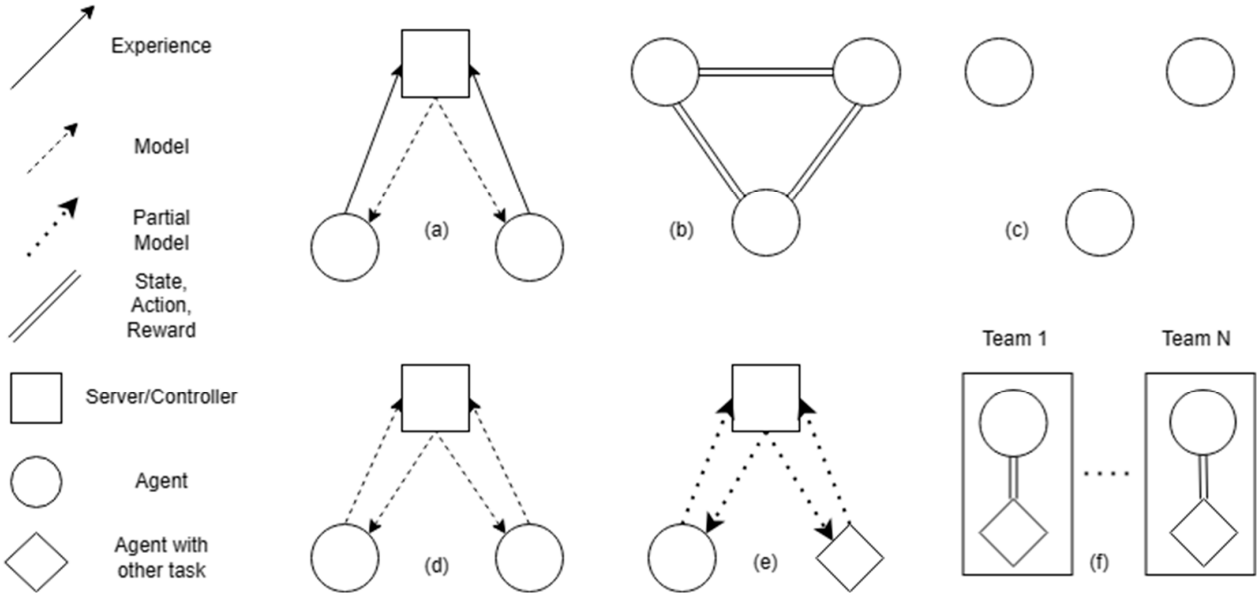

Figure 1. Distributed RL frameworks, where the communication between components is visualized. (**a**) CTDE; (**b**) FDC; (**c**) FDI; (**d**) HFRL; (**e**) VFRL; (**f**) TeamL.

3.2. Radio Resources and Power Allocation

Resources available at the RAN are the spectrum and power, where the spectrum resources are virtualized in such a way that they are easier to customize and assign efficiently. Additional tasks are closely associated with the management of the resources just mentioned, such as channel state information (CSI) prediction, beamforming, interference management, or handoff management. We do not deal with such aspects here, and instead concentrate on the core RM task. Readers interested in additional aspects are referred to, e.g., [6–8,31,32].

In [33,34], the CTDE approach is followed, where in [33] spectrum and energy efficiency are optimized concurrently, while in [34] user association and power allocation are optimized. Benchmarks show that CTDE converges faster than a fully centralized algorithm.

The FDI approach is taken in [35], where agents correspond to base stations and each of them trains its own policy targeting low latency. However, the training occurs offline and there is no AI workflow associated with the models. This is a shortcoming because, although the independent agents do not interact, there is still a need for a mechanism to exercise pre-emptive control in case individual agents fail as a result of instability due to changed conditions (a phenomenon that may become more likely due to the lack of interaction between agents).

In [36], FDC agents optimize user association and power allocation, sharing observations with a fixed amount of neighbors and computing a global reward. Their performance is close to conventional sophisticated centralized algorithms. In [37], FDC agents are used to execute the task of user association and resource allocation in RANs, where the state is shared between agents. Again, in both [36,37], the training is performed implicitly offline, and no associated AI workflow is developed. The number of agents is hardcoded into the problem description, and this creates difficulties when transferring knowledge from models trained in smaller systems to larger ones. A more recent FDC approach [38] includes a design for AI workflows in the context of O-RAN, including model construction, training, deployment, inference, evaluation, and update. Multiple agents execute their actions separately but optimize a global objective by sharing their rewards.

In [39], a team of heterogeneous RL agents is trained to optimize both power and resource allocation in an O-RAN setting. Three types of training are compared, differing in the sequence of agents' actions and in the type of rewards (individual or team ones) used for training. The algorithm designed according to the TeamL framework performed

best. It is noted that this study considered the compatibility of agents but did not examine scalability and transferability.

Another way to combine heterogeneous agents is examined in [40] using VFRL. This approach was benchmarked against a central agent and two independent agents, and the VFRL algorithm was shown to converge fast and perform comparatively well.

3.3. Edge Caching

The proper management of cache resources located at the edge is critical to avoiding unnecessary traffic overloads between core and edge, while also providing low latency to users [41]. To achieve this, user preferences must be predicted based on previous usage patterns. This should happen within the edge, otherwise data will need to be transmitted to the core for training, partially canceling the benefits of caching. In addition, privacy concerns dictate that learning should happen as close to the end-devices as possible. Due to the dynamicity of content in mobile networks, methodologies that allow for online learning are desired. Again, in considering the problem, we focus on distributed RL methodologies. For other approaches that are centralized or use SL, the reader may consult, e.g., [7,41].

FDC agents are used in [42], assuming an offline training phase, shared state, and common reward. Edge caching in [43,44] is also treated with an FDC algorithm, sharing a global state between agents, with the difference being that it is compressed through learning in order to minimize communication costs. Again, it is assumed that the necessary components are pre-trained, and no AI workflows are described. The model is also sensitive to the number of agents, so the model cannot be reused as a starting point for larger topologies.

By contrast, ref. [45] uses HFRL to train online and communicate a shared model between base stations through an aggregating server, making it more scalable. However, agents do not cooperate through the exchange of information. In [46], HFRL is used again, but agents do exchange information, achieving faster convergence compared to non-cooperating agents.

3.4. Edge and Core Computing

Edge and core computing need to coordinate and perform task offloading in the most efficient and performant way, due to the edge being resource-constrained and the core being associated with larger latencies. At the same time, VNFs need to be placed at the appropriate location in order to serve incoming tasks. Works covering various aspects of multi-access edge computing (MEC) planning, management, orchestration, and optimization can be found in [9,47].

Centralized deep RL (DRL) is compared to HFRL in [48], where comparable performance is achieved but with greatly reduced communication costs. In [49], both FDC agents and a HFRL approach were used to offload tasks among small base stations (SBS) representing vehicles, where both demonstrated superior performance compared to the FDI agents and to more conventional algorithms. In fact, the independent learning algorithm showed worse convergence properties compared to the two. However, the FDC algorithm had increasing communication costs when scaled-up, while the HFRL algorithm was associated with reduced costs. The models were able to tackle both performance and energy-efficiency. The HFRL algorithm had the additional benefit of being more easily transferable, since it is independent of the number of agents, at the cost of performance compared to the FDC algorithm.

An FDC algorithm was also used in [50] for task offloading and compared to centralized DRL, an exhaustive approach, and a genetic algorithm. Centralized DRL had a worse performance due to insufficient training. The conventional approaches performed somewhat better than FDC agents but were much slower during inference.

In the context of non-orthogonal multiple access (NOMA)-assisted MEC, ref. [51] examines the joint task of partially offloading tasks to edge servers while using minimal

channel resources under FDI. While there are no extra communication costs when scaling up, the convergence of the algorithm may not be sufficiently fast.

In [52], placement of tasks on an edge or fog environment is solved by applying a modified distributed DRL algorithm that employs the CTDE paradigm. Convergence is accelerated due to the actors exploring in parallel, but communication costs are expected to increase when scaling-up. Interoperation with a cloud environment is not examined.

3.5. End-to-End

End-to-end management in the form of slicing has been widely researched in recent years. Slicing is composed of several phases [53], but we focus on the end-to-end task of allocating resources for each slice and the relevant runtime. Other aspects of end-to-end management include fault management, performance monitoring, and others [26], which do not fall under the category of RM. For a review of non-distributed RL methodologies on end-to-end slicing, see [54], while for more general ML approaches, see [55].

In [56], a distributed heterogeneous DRL algorithm is applied, where each agent is associated with a slice type and resource type, essentially following the CTDE paradigm. Generalization was achieved by training under different kinds of network topologies and leveraging the heterogeneity of agents. The algorithm showed superior performance compared to a relatively simple empirical algorithm as the system was scaled-up. However, as the number of slice types increases, as in the 6G vision, it is an open question as to how fast the algorithm can reach convergence.

By contrast, a sophisticated architecture is presented in [57], where a combination of prediction algorithms using FL and control algorithms using MARL is used to achieve closed-loop control. The control part can use cross-domain input to make decisions, while the prediction part leverages FL to dramatically reduce communication costs when transferring information from a local to global scope. Another mechanism is employed in [58,59], where the FDC approach is essentially taken, and each agent coordinates with the others by means of certain parameters enforcing the resource constraints or by sharing reward and state space. Sharing more information improves convergence.

In [60], the FDI approach is followed, where each agent manages a subnetwork. However, agents utilize transfer learning among them, allowing for faster convergence and similar performance as algorithms trained from scratch.

3.6. Unifying and Distinguishing Features across Resource Types and Domains

The insight from the review of the methodologies in the previous subsections is collectively summarized into Table 1, classified in accordance with the respective methodological frameworks. Moreover, each framework is assessed in terms of the criteria formulated in Section 3.1, namely scalability, composability/modularity, and transferability. In addition to the elements included in the table, we now proceed with a discussion of further noteworthy aspects.

RRM requires coordination between several different tasks, so more advanced architectures become necessary that consider composability. Some of the reviewed works included a management framework for AI models in the context of O-RAN, which aids composability by providing interfaces to be exploited by end-to-end NMO. However, transferability between topologies and different network sizes is not properly dealt with. This is because interactions between agents introduce communication that increases both the costs and the state space, decreasing the benefits of distributed learning. We will later examine promising methodologies that allow for transferability while increasing scalability as well.

In edge caching, interactions between agents are desired, since content should be duplicated as little as possible while maintaining low-latency decision making, so communicating decentralized architectures should be preferred. Again, interactions between agents complicate the decentralization process. In addition, methodologies must deal with optimizing different objectives, such as data offloading or delay, and being more scalable by exploiting the spatial structure of wireless networks [61]. Similar arguments work for edge

and core computing, since the state of each agent should consider neighbouring agents to decide task offloading, while a centralized solution suffers from scalability issues.

Table 1. Distributed RL frameworks: features, potential, and related works.

Framework	Communication	Scalability	Composability/ Modularity	Transferability	Related Works
CTE [1] (baseline)	Experience	Very low: convergence slows down with state–action space, and high communication costs to collect data centrally.	No	Low: features depend on system size.	(Mentioning only works that employ CTE as baseline for comparison to distributed RL schemes) RRM: [33,34,40] Compute: [48,50] End-to-end: [57]
1-6 CTDE	Experience	Low: reduced state space, but communication costs are high and no interactions between agents during execution (high potential for instability when scaling up).	No	High: agents have local actions and state.	RRM: [33,34] Compute: [52] End-to-end: [56]
FDC	State, action, reward	Medium: reduced state space and agents interact (good stability), but communication costs increase.	No	Medium: communication may depend on number of neighbors.	RRM: [36–38] Caching: [42–44] Compute: [49,50] End-to-end: [58,59]
FDI	None	Low: reduced state space and no communication costs but no interactions (high potential for instability when scaling up).	Yes (at expense of scalability)	High: agents have local actions and state.	RRM: [35] Compute: [49,51] End-to-end: [60]
HFRL	Model	Medium: reduced state space and communication rounds can be optimized but may lack interactions.	No	Medium to high: agents have local actions and state unless they include interactions.	Caching: [45,46] Compute: [48,49]
VFRL	Partial model	Medium: reduced state space and communication rounds can be optimized but may lack interactions.	Yes	Medium: agents have local actions and state unless they include interactions	RRM: [40]
TeamL	Intra-team state, action, reward	Medium: reduced state space but has inter-team and intra-team communication costs.	Yes	Medium: depends on the interaction between teams.	RRM: [39]

[1] Centralized training and execution: not a distributed RL framework; included as a baseline.

End-to-end slicing differs in nature from other RM problems because they are inherently cross-domain. Multiple types of resources must be allocated in tandem while considering resource constraints, so the allocation of resources to different slices must be considered jointly. Current methods assume a small number of slices that are centralized, so their scalability in the 6G context may be limited. Also, the spatio-temporal distribution of demands has not been fully accounted for [16]. More generally, the exact form of network slicing is still a subject of research, with different efforts adopting different designs [62]. For example, a hierarchical approach may be taken, where a higher-level controller decides on a more aggregate level for the slice allocations while lower-level controllers decide exactly how resources are allocated to devices [54]. However, the appropriate level of granularity for network management decisions at different levels in such a hierarchy is an open issue [16], with a high impact on scalability. Finally, the behaviour of end-to-end management and orchestration based on AI is as yet a largely unknown territory, and rigorous mathematical tools should be used to understand it [63].

When examining the work on all the domains together, we make the observation that, in general, composability is not yet a high concern in research efforts. A reason for this

may be the need for introducing interfaces between agents and communication protocols. These need to be standardized, something that takes time. Additionally, more composable systems would tend to be associated with agents using larger state spaces and having increased communication costs, which would complicate the balance between convergence time and scalability.

Another observation is that there is no clear winner among the frameworks, as the suitability of each depends very much on the problem conditions. For example, CTDE may be suitable when deployed on similar but uncoupled network domains, while FDC may be necessary when decision-making units are strongly coupled, as in the case of edge caching and computing.

4. Supporting Distributed Frameworks with Native AI Enablers

4.1. Native AI Enablers and Their Functionality

We saw in Section 3 that introducing multiple local agents that control network resources can be effective at a local level in achieving faster convergence by reducing the state and action space while also reducing latency and communication costs, if the system scope is small enough [40,46,49,50]. However, as more domains and tasks are added to the system, interactions among agents proliferate and the state space increases for each task. Increasing interactions increase communication costs and may require retraining of the agent from scratch, while state space expansion increases convergence time. It is important to find a balance between including all information when agents communicate and keeping the agent state space small to boost scalability [17,64]. A similar balance is required when considering the alternative of reducing interactions, as this could lead to suboptimal performance and unexpected behavior.

To facilitate the generation of models with increased performance in a safe way, create mechanisms that select appropriate models for each situation, and exercise effective pre-emptive control in cases of failure, certain AI enablers have surfaced. The first of these is an outward-facing interface provided by the AI plane, which sets up the processes managing the lifecycle of AI models [1,23]. These processes monitor system performance, collect data for training, schedule offline and online training of models, and finally replace models either proactively or responsively, while also optimizing the runtime of AI models. By doing so, they enable automation and establish closed-loop control. The second enabler is a knowledge reuse system that undertakes the management of knowledge, consisting of the storage and retrieval of collected data, maintenance of model libraries, and model training. The third enabler is DTNs, which can be used to predict performance and train new models.

In more detail, the first enabler, the AI plane (or Network Intelligence (NI) plane as is alternatively referred to) establishes the AI workflows necessary for AI models lifecycle management and closed-loop control. Based on recent work (see [10,65] and references therein), the management of Network Intelligence, essentially a composition of AI models achieving a shared goal, will be organized in small parallel control-loops responsible for inference, offline and online training. These control-loops are broken into steps that can be expressed in terms of the so-called monitor–analyze–plan–execute based on knowledge (MAPE-K) framework (depicted in Figure 2, which is described in detail later in this section, when discussing the interplay of the three enablers). Crucially, NI should be able to coordinate with other NI instances towards promoting composability, in alignment with the respective criterion of Section 3. However, a trade-off between creating manageable end-to-end workflows and achieving scalability arises, since NI must operate within a certain timeframe and network scope that constrains the amount of time and information available for training. Training will be triggered either by a new phase of planning, in order to deploy updated models, or when a failure occurs or is predicted to occur.

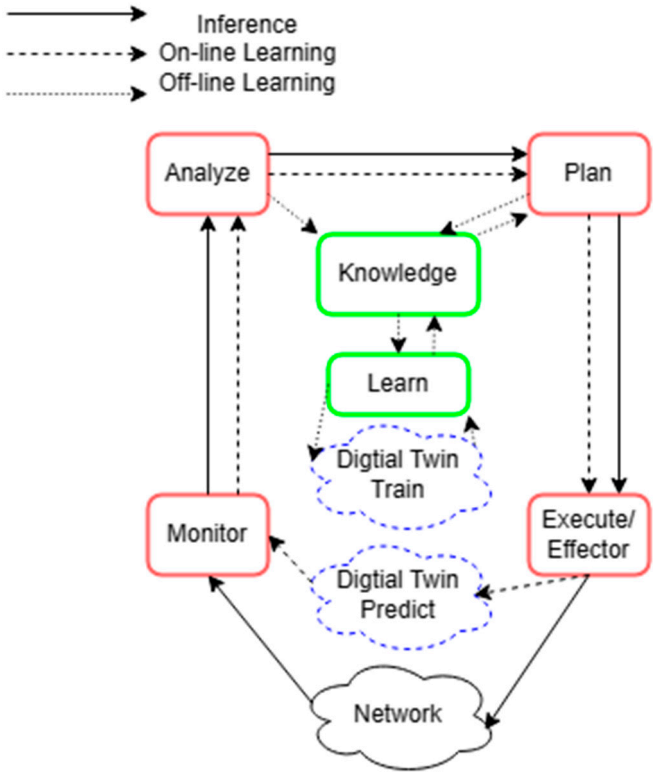

Figure 2. The processes in charge of training, deploying, and maintaining AI models on network infrastructure. The AI plane processes are colored red, the knowledge reuse system green, and the DTNs blue. Knowledge reuse and DTNs are nested within the AI plane and do not interact directly with the network.

The second enabler, namely knowledge reuse, is used to store knowledge in various forms, grow model libraries, and select the appropriate model for the current state of the environment [13]. By managing a model library that is continually enriched offline with data collected from the live environment and simulations (see Figure 2) and dynamically employing appropriate models from the library through the said selection mechanism, knowledge reuse plays an important role in model maintenance and significantly enhances the viability of AI algorithms. There are various ways to store knowledge, including as trained models, collected experience, or exploration strategies in the context of RL, organized by associated task. Mapping functions from source tasks to target tasks can even create learner models that are able to execute tasks jointly [11]. Knowledge reuse will be embedded into AI workflows, and its data collection intervals and storage will be controlled by the AI plane. However, this knowledge should either be stored and consumed close to its production, or it should be transmitted at large time intervals to prevent excessive usage of network resources.

Knowledge reuse requires system resources for data collection/storage and for computation, and also needs to conduct training with respect to multiple environments. Although AI workflows can assist with the data collection and various steps of model maintenance, there is also a need for a safe environment that can be used for training and predicting future performance. The third enabler, DTNs [12,13], undertake this crucial task, as DTNs can learn to mimic system behavior and provide the capability of experimenting by means of "What If" scenarios without impacting real-network performance. In addition, different digital twins with different scopes enable performance predictions in reasonable time frames and for a broad spectrum of targets, from end-to-end performance to more localized tasks at the packet-level. In this manner, DTNs are valuable also for striking the right balance between obtaining precise models at the required level of granularity and making efficient use of the system resources mentioned in the paragraph's beginning. Note that, in

order to further promote system efficiency, DTNs should be hosted close to the devices they simulate, with larger DTNs existing closer to the core where data can be centrally collected, and resources are more abundant [66].

Figure 2 depicts the interaction between the three enablers and the network. As mentioned earlier, workflows follow the MAPE-K framework and evolve in accordance with a number of steps, governed by the AI plane with support from knowledge reuse. These steps are as follows: monitoring the network; analyzing the collected data; planning the deployment of models based on the analysis; executing the plan and applying the effects to the network (the block corresponding to the last step consists of a high-level part for executing the plan and a low-level "effector" part for enforcing the actions associated with the plan through the actual network controllers). Three types of workflow are relevant (see the legend in the figure), as follows:

- Inference, which makes network control decisions and monitors model performance, taking steps to ensure it is acceptable. By monitoring, analyzing, and planning, inference can choose to deploy updated models (made available through online or offline learning, discussed next) to the live environment. Model updates are triggered when a potential for performance enhancements is detected, or when performance drops during inference or is predicted to drop by the online learning workflow.
- Online learning, which improves operating models. By collecting monitoring data from a digital twin and analyzing them, the system may plan to update these models while they are operating. The employed digital twin is connected to the live environment (see the DTN at the bottom of the figure) and enables the training of models under conditions as close as possible to the current ones, while also avoiding the risk of negatively impacting the actual network performance. If the online training workflow detects that the updated model has the potential for improved network performance, a replacement of the model currently used for inference may be triggered. Contrarily, if performance is predicted to drop beyond acceptable levels, a replacement of the inference model will again be triggered, this time selecting the new model among models made available through offline learning.
- Offline learning, which is managed by the knowledge reuse system. It is responsible for processing and organizing monitoring data, model libraries, and scheduling the training of new models. It takes input from analysis to determine how and which models to train. It also coordinates with planning to provide new models upon request or prompt the replacement of models with new ones. The DTN (topmost in the figure) employed in offline learning may be configured to explore scopes of different breadth.

In terms of the criteria set in Section 3, it may be seen that the aforementioned model training and deployment processes create a potential for producing more scalable and composable/modular models. Indeed, through the associated DTNs, the knowledge reuse system explores the ML model space in gradually larger and diverse scopes, keeping the most scalable and composable candidates. For a more concrete example, consider the problem of determining the appropriate size of the input in models that employ communication between multiple agents. Increasing the input size improves composability, since agents become more aware of the global state. However, this comes at the expense of scalability due to increased costs for the communication between the agents. By maintaining and utilizing previous knowledge, this trade-off may be balanced optimally. As another example, consider the adoption of larger models, which can learn more detailed representations of the environment, so that they can generalize over more diverse conditions and, thus, be more scalable to larger network topologies. However, larger models require more training time to reach convergence, so early deployments would lead to decreased performance instead of an improvement. The knowledge reuse system enables the deployment of larger models by collecting data and maintaining a model library, so that transfer learning techniques can be applied systematically, and model training can be sped up. In order to best exploit this ability of the knowledge reuse system, it is important to adopt model families that have

a greater potential for transferability, so that they may generalize more flexibly over the number and heterogeneity of agents.

To summarize, setting up the associated workflows as previously described enables the exploration of the model space in a way that balances scalability with composability. By leveraging DTNs, exploration can be performed in a safe manner, while the knowledge reuse system, together with input from analysis and planning, organizes and accelerates the exploration. The AI plane orchestrates the cooperation of the knowledge reuse system with the management and operation of the actual network.

4.2. Interactions between the Distributed RL Frameworks in Table 1 and AI Enablers

The six methodological frameworks, that were discussed in Section 3 and summarized in Table 1 differ in their interactions with AI enablers, as depicted in Table 2. The evaluation is based on the MAPE-K framework steps. For the AI plane, we examine how easy it is to plan/deploy each framework as an initial step and how easy it is to replace units when operation fails. For the DTN, we examine the smallest possible size of the digital twin required for performance predictions. For knowledge reuse, we examine the amount of knowledge needed in terms of decision-making units.

Table 2. Frameworks and their footprint on each AI Enabler.

Framework	AI Plane	Digital Twin (Size)	Knowledge Reuse
CTDE	Easy to plan/hard to replace	Small	Simple, a single unit
FDC	Hard to plan/easy to replace	Large	Simple, a single unit
FDI	Easy to plan/easy to replace	Small	Multiple units, depending on the environment
HFRL	Easy to plan/hard to replace	Large	Simple, a single unit
VFRL	Easy to plan/hard to replace	Large	Simple, a single unit
TeamL	Planning depends on inter-team interactions; easy to replace teams.	Large	Depends on the size of the team

In the CTDE framework, there is a single central learner and multiple actors/executors that are copies of each other. During planning, the single central model is selected and deployed by copying to multiple network locations, which is simple to perform. However, when performance fails, all actors/executors fail together and need to be replaced at the same time, potentially having a big impact on performance since they affect multiple locations or multiple resources. The digital twin may be composed of a single agent for the central learner, and the only knowledge necessary is the single unit that is copied.

In FDC, multiple agents communicate, cooperate, and train independently at different network locations and may manage different resources. During planning, many agents at different locations or in charge of different resources need to be deployed together, be initialized, and establish cooperation, which is a complex task. By contrast, if failure occurs, only the single failed agent needs to be replaced and, since the agents are more independent than in CTDE, this is a simpler task. The digital twin must be composed of many agents in order to learn communication. Finally, assuming convergence has been reached for all agents, knowledge is a single unit.

In FDI, agents do not communicate and instead manage their resources independently. Agents are both easy to plan and replace, since they do not interact and can be replaced independently of each other. The digital twin is the size of a single unit, but the knowledge must be gained under different conditions, because agents do not collect the more diverse set of observations that FDC agents collect and share via communication. To compensate, FDI agents must be trained under a greater array of conditions. Potentially, the service consumer is presented with different agents for each condition.

HFRL and VFRL share behaviors due to their federated nature, with both training in a decentralized fashion. While the training occurs differently than in CTDE, HFRL and VFRL still share knowledge at a central server, become synchronized, and share the same model, so in this respect they are the similar to CTDE. Thus, they are easy to plan for but need to

be replaced as one when failure is detected. The digital twin may be large, since training requires many decentralized trainers. Knowledge stored is a single unit, representing the aggregated model.

Finally, in TeamL each team is composed of heterogeneous agents that communicate, so it is similar to FDC. It is moderate to plan for, depending on the size of the team and inter-team interactions, and replacement in this case can be performed even within a team, exchanging a single team member, as in the FDC scenario. Note that if there are multiple teams, each team can act either as FDC or FDI depending on the information exchanged. The digital twin may be large, and knowledge stored will depend on the size of the team.

It is remarked that, apart from the general characteristics of a given framework, which determine the decision-making units and their interactions, the internal structure of the agents for each specific methodology in the framework also plays a role and must be considered. However, the degree to which a methodology can materialize the potential benefits from utilizing the AI enablers mostly depends on the inter-agent structure and interactions, which are determined by the framework in use. As an illustrating example referring to the AI plane, independent agents will not be able to participate in adaptive localized closed-loop controls, regardless of their internal structure, because they will be unable to capture the non-stationarity of their environment.

5. Lessons Learned, Open Issues, and Future Directions

While AI enablers are necessary for the training and execution of AI models, the structure of the considered methodological frameworks would also require modifications to fully meet the benchmarking criteria set in Section 3. Scalability can be improved by properly incorporating agent interactions. In the context of network intelligence, learned communication, i.e., incorporating communication as part of the ML module in an agent, would potentially increase system performance while minimizing communication costs in a suitable timeframe related to the communication frequency between agents [63].

Another method of improving model performance is online learning, which is already anticipated by the AI plane as envisioned in [65]. By adding more components and associated agents into the network, the system becomes more dynamic, and models need to adapt faster to changing conditions. Scalability is then expected to improve by incorporating online learning, which mitigates the so-called model drift.

Composability could be enhanced by adopting methodologies, such as team-learning or VFRL, and, in general, heterogeneous MARL. This is necessary for network intelligence, as locally communicating NI instances that are responsible for different tasks need to coordinate. Such approaches require a certain degree of standardization between agents that is non-trivial to achieve [1,10].

End-to-end composability necessitates trustworthiness of the network and the ability to interpret AI model behavior using mathematically rigorous tools [63]. This is because adopting AI solutions and composing them on such a large scale is expected to be very challenging due to the statistical heterogeneity experienced in large-scale networks, which directly affects the performance and convergence rate of these solutions. This issue is particularly pronounced in highly dynamic environments, such as 6G networks.

Transferability can be increased by adopting new neural network architectures that generalize over network structure and relations between decision-making units, namely graph neural networks (GNNs). This also increases scalability because GNNs can generalize over system size and topology [67]. GNNs have been used for performance prediction and control [68–71] but considering that distributed control is a step further in complexity, there is a lot of room for research.

Finally, it should be considered that the methodological frameworks themselves are governed by parameters [72,73], such as frequency of communication between agents in FDC, the aggregation interval or the weighting of each agent's model in the FL-based frameworks, and the values for these parameters also need to be optimally selected. On a related note, additional improvements in FL-based frameworks can be made by introducing

intermediate layers in a hierarchy towards aggregating models in more steps, such as in fog learning [74].

6. Conclusions

By examining the 6G challenges pertaining to closed-loop control on RM, we narrowed down the scope of relevant methodologies into distributed RL. We classified relevant approaches into six methodological frameworks based on their structure and interactions and analyzed them in terms of their scalability, composability/modularity, and transferability. Examining applications based on these methodological frameworks, certain trade-offs between (i) scalability and convergence time, (ii) scalability and composability, and (iii) end-to-end orchestration and scalability were exposed and examined. AI enablers become essential to balancing these trade-offs, especially in view of the dynamic nature and stringent requirements that will characterize 6G networks. The AI plane, knowledge reuse, DTNs, and their interoperation were discussed and their potential to provide support for the considered distributed RL frameworks was explored. In terms of further improvements, future research points to promising directions, such as learned communication, heterogeneous MARL, and graph neural Networks.

Author Contributions: Conceptualization, O.A.K. and K.K.; methodology, O.A.K., A.Z., K.K. and S.P.; validation, O.A.K., A.Z., K.K. and S.P.; writing—original draft preparation, O.A.K.; writing—review and editing, O.A.K., A.Z., K.K. and S.P.; supervision, K.K. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the ICOS project ("Towards a functional continuum operating system") funded from the European Union's HORIZON Research and Innovation Programme, under grant agreement No 101070177 and the 6GREEN project ("Green technologies for 5/6G Service-Based Architectures") funded from the European Union's HORIZON Research and Innovation Programme, under grant agreement No 101096925.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Summary of Related Work and Brief Discussion of Surveys Treating Relevant Aspects in a Different Context

Table A1. Summary of related work (discussed in Section 1.3).

Ref.	Description	Comparison to this Work
[11]	Focuses on end-to-end RM for the specific context of network slicing. Provides a taxonomy of AI methodologies and discusses aspects of a knowledge reuse system, including the type of data stored and curated and workflows for ML model training. Places emphasis on transfer learning.	This work considers AI workflows and digital twin networks in addition to knowledge transfer, conducts a focused survey of distributed RL methodologies for RM tasks of a much wider scope, organizes/abstracts the surveyed methodologies in a number of frameworks, and discusses how the AI enablers support each framework.
[14]	Focuses on distributed methodologies and network technologies from an end-to-end network-for-AI viewpoint and examines how network technologies support distributed ML operation.	This work takes an AI-for-network approach.
[15]	Focuses on architectures of distributed networked ML from an end-to-end network-for-AI-approach. Discusses optimizations of the algorithm's training process in each architecture, such as communication efficient techniques and asynchronous methods.	This work takes an AI-for-network approach.

Table A1. *Cont.*

Ref.	Description	Comparison to this Work
[16]	Proposes virtualization of user demand using digital twins. Discusses pervasive intelligence and categorizes it into network management and service-oriented AI. Describes a network architecture based on an interplay between AI and virtualization supported by digital twins, but does not consider other AI enablers or specify a scope of RM tasks.	This work considers digital twins integrated with AI workflows and a knowledge reuse system, conducts a focused survey of distributed RL methodologies for RM tasks, organizes/abstracts the surveyed methodologies in a number of frameworks, and discusses how the AI enablers support each framework.
[17]	Focuses on pervasive AI for highly heterogeneous networks with computational facilities, such as IoT networks. Adopting a network-for-AI viewpoint, it focuses on distributed AI architectures and techniques, such as parallelization and model splitting, and considers the optimization of training and inference processes.	This work takes an AI-for-network approach.

Table A2. Surveys treating relevant aspects in a different scope.

Ref.	Description
[4]	Focuses on zero-touch management and categorizes ML applications by network domain, including distributed solutions. Does not abstract overdistributed ML approaches and does not consider AI enablers. Discusses ZTM architecture in detail and surveys extensively how ZTM may benefit from applications of ML.
[7]	Provides an extensive survey of ML applications to networking, categorizing them by network task (including RM) and ML algorithm. Does not abstract over ML approaches, does not focus on distributed solutions, keeps the scope to 4G/5G and does not consider AI enablers. Offers preliminaries on SL, UL, RL, and neural networks.
[8]	Surveys RRM tasks in 5G, including distributed ML solutions. Does not abstract over ML approaches, does not include AI enablers, and does not consider other types of resources. Compares ML approaches via simulation.
[9]	Focuses on edge/core VNF and CNF placement in 5G. Does not abstract over ML approaches, does not discuss AI enablers, and does not consider other types of resources. Thoroughly classifies approaches by task, scenario, algorithm, and objective. Includes non-ML approaches.
[25]	Focuses on 5G/6G technologies, surveys potential applications of ML to networking, and discusses the benefits of ML compared to other approaches. Does not abstract over ML approaches, and does not focus on distributed solutions or AI enablers. A brief introduction to SL, UL, RL, and neural networks is given.
[27]	Offers a detailed tutorial on SL, UL, and RL in the context of deep learning for wireless networks. Includes general theory, the most relevant architectures, and training algorithms. Does not include a survey over specific ML applications to networking, does not examine distributed AI architectures or 6G, does not focus on RM, and does not include AI enablers. Discusses a possible cooperation between ML and conventional algorithms.
[28]	Provides a taxonomy of deep learning applications to networking, per network layer and type of task. Keeps the scope to 5G, does not consider AI enablers, does not focus on RM, and does not abstract oversurveyed approaches. Discusses deep learning theory and compares DL frameworks.
[29]	Focuses on the integration of deep learning into the network, such as deployment of ML and specific ML implementations tailored to networking. Keeps the scope to 5G, does not abstract over approaches, does not examine multi-agent systems, does not include AI enablers, and does not focus on RM. Provides a brief introduction to DL.
[30]	Surveys deep RL for autonomous control in networks and proposes a reference architecture for DRL, including distributed solutions. Keeps the scope to 5G, does not include AI enablers, does not focus on RM, and does not abstract in terms of distributed architectures, focusing on network layers instead. Provides theory on advanced DRL algorithms.
[31]	Focuses on RRM tasks in HetNet scenarios. Keeps the scope to 5G, does not abstract over ML approaches, does not include other type of resources, and does not include AI enablers. Discusses non-ML and ML approaches and examines many aspects of each task/approach combination.
[32]	Focuses on RAN RM tasks and ML applications to RAN physical layer technologies. Keeps the scope to 5G, does not abstract over ML approaches, does not consider other types of resources, and does not include AI enablers. Compares non-ML approaches to ML.

Appendix B. ML Preliminaries

ML is a discipline concerned with creating programs that are capable of learning a task from data, in a way that the task can be performed on new problem instances. The programs are usually called ML models. Learning is most often categorized into three flavors.

Appendix B.1. Supervised Learning (SL)

A training set is given, composed of samples that have features and associated labels. The target of supervised learning is to learn the mapping function between features and labels, i.e., to predict the labels from the features. The initial untrained model is a mapping function selected from an appropriate function space, the structure of which usually entails some kind of bias. One simple example is linear regression, where the label is assumed to be a linear combination of the features.

Given the form of the mapping function/model, a training algorithm is applied, which adjusts the model with the purpose of minimizing the value of an objective function that quantifies the average difference between the labels predicted by the model given the features and the actual labels, over all samples in the training set. Usually, the model is updated by means of an iterative procedure, whose details and complexity vary depending on the function space chosen for the model. Once certain target criteria are met, such as reaching a maximum number of iterations or reducing the value of the objective function below a threshold, the model is considered to be trained.

Many different function spaces and training algorithms exist. Widely employed models include decision trees, k-nearest-neighbors, support vector machines, and neural networks. Applications related to networking encompass channel quality indicator prediction and traffic classification. For these and other networking applications see, e.g., [7,25,28,29]. For a thorough introduction to ML in general and SL in particular, the reader may consult [75].

Appendix B.2. Unsupervised Learning (UL)

Again, a training set is given, but now each sample in the set consists only of features, with no labels attached. The task of learning in this case is to determine the data distribution that generates the samples. A common application is clustering, where data samples must be grouped together in clusters, but applications can also relate to data point generation, such as in image compression. As in SL, a model space is chosen and trained via minimization of an objective, but now the objective quantifies the difference between the predicted and actual distribution of samples, measured in various ways. Commonly used function spaces are k-means and the so-called generative models. Applications in networking include user association and mobility management. For these and other networking applications see, e.g., [7,25,28,29]. For more details on UL in general, Ref. [75] may be consulted.

Appendix B.3. Reinforcement Learning (RL)

In RL, the training is not based on an a priori available sample set. Instead, the interactions of an agent with its environment are considered. At each step, the environment assumes a state within a predetermined state space. The agent observes the current state of the environment and takes an action (from within a predetermined action space) based on the agent's policy (which may be deterministic or stochastic) and the observation. Each such action results in a reward credited to the agent and in the transition of the environment to a new state. Subsequently, the agent makes a new observation and the cycle repeats. Training seeks to directly or indirectly determine the action-taking policy of the agent that maximizes the long-term reward, i.e., the sum of rewards accumulated over a prespecified (often infinite) window of interactions. The terms in the sum are usually weighted exponentially by means of a discount factor that prioritizes more recent rewards.

Since the agent must consider the long-term reward, taking the action with the highest immediate reward is not optimal. For this reason, the agent does not always select an

action on the basis of its current policy, but also exercises "exploration" with the purpose of experiencing "previously unseen" combinations of state/action pairs. Exploration is more frequent during the earlier stages of training. Convergence of the training process is determined by checking the value of an objective that quantifies the extent to which perturbations of the current policy would lead to further improvements towards the (unknown) optimal. For more details on RL, the reader may consult [76].

Recent agent architectures often split the agent into separate actor and learner modules, referred to as an actor–critic pair, where the actor executes the policy and collects observations, and the learner improves the policy by evaluating the taken actions. This more advanced technique is discussed in [30], together with other relevant techniques.

The scenario of having multiple agents that act in the same environment is pertinent to our scope. Each agent can exchange information with other agents as part of observing the environment. Training of all agents may occur in conjunction or independently, depending on whether the policy is somehow shared among agents. Each agent may train to maximize its own reward, or a global reward shared between agents. Agents may be based on the actor–critic architecture, where multiple agents share the same policy and a centralized learner does the training, as in the CTDE framework in Section 3.

Applications of RL in networking include resource allocation/management, packet routing, and many other tasks; see, e.g., [7,25,28–30]. In principle, virtually any decision-making problem can be formulated in terms of RL, making this type of learning extremely pertinent to networking.

References

1. Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge Artificial Intelligence for 6G: Vision, Enabling Technologies, and Applications. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 5–36. [CrossRef]
2. Zhang, S.; Zhu, D. Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities. *Comput. Netw.* **2020**, *183*, 107556. [CrossRef]
3. Tataria, H.; Shafi, M.; Molisch, A.F.; Dohler, M.; Sjoland, H.; Tufvesson, F. 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. *Proc. IEEE* **2021**, *109*, 1166–1199. [CrossRef]
4. Coronado, E.; Behravesh, R.; Subramanya, T.; Fernandez-Fernandez, A.; Siddiqui, M.S.; Costa-Perez, X.; Riggio, R. Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2535–2578. [CrossRef]
5. Nassef, O.; Sun, W.; Purmehdi, H.; Tatipamula, M.; Mahmoodi, T. A survey: Distributed Machine Learning for 5G and beyond. *Comput. Netw.* **2022**, *207*, 108820. [CrossRef]
6. Hu, S.; Chen, X.; Ni, W.; Hossain, E.; Wang, X. Distributed Machine Learning for Wireless Communication Networks: Techniques, Architectures, and Applications. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1458–1493. [CrossRef]
7. Sun, Y.; Peng, M.; Zhou, Y.; Huang, Y.; Mao, S. Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3072–3108. [CrossRef]
8. Bartsiokas, I.A.; Gkonis, P.K.; Kaklamani, D.I.; Venieris, I.S. ML-Based Radio Resource Management in 5G and Beyond Networks: A Survey. *IEEE Access* **2022**, *10*, 83507–83528. [CrossRef]
9. Attaoui, W.; Sabir, E.; Elbiaze, H.; Guizani, M. VNF and CNF Placement in 5G: Recent Advances and Future Trends. *IEEE Trans. Netw. Serv. Manag.* **2023**, *1*. [CrossRef]
10. Camelo, M.; Cominardi, L.; Gramaglia, M.; Fiore, M.; Garcia-Saavedra, A.; Fuentes, L.; De Vleeschauwer, D.; Soto-Arenas, P.; Slamnik-Krijestorac, N.; Ballesteros, J.; et al. Requirements and Specifications for the Orchestration of Network Intelligence in 6G. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Virtual, 8–11 January 2022; pp. 1–9.
11. Zhou, H.; Erol-Kantarci, M.; Poor, V. Knowledge Transfer and Reuse: A Case Study of AI-enabled Resource Management in RAN Slicing. *IEEE Wirel. Commun.* **2022**, 1–10. [CrossRef]
12. Hui, L.; Wang, M.; Zhang, L.; Lu, L.; Cui, Y. Digital Twin for Networking: A Data-driven Performance Modeling Perspective. *IEEE Netw.* **2022**, 1–8. [CrossRef]
13. Zhou, C.; Yang, H.; Duan, X.; Lopez, D.; Pastor, A.; Wu, Q.; Boucadair, M.; Jacquenet, C. Digital Twin Network: Concepts and Reference Architecture; Internet Engineering Task Force; Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-03, 27 April 2023. Available online: https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-03 (accessed on 1 July 2023).
14. Campolo, C.; Iera, A.; Molinaro, A. Network for Distributed Intelligence: A Survey and Future Perspectives. *IEEE Access* **2023**, *11*, 52840–52861. [CrossRef]

15. Liu, X.; Yu, J.; Liu, Y.; Gao, Y.; Mahmoodi, T.; Lambotharan, S.; Tsang, D.H.-K. Distributed Intelligence in Wireless Networks. *IEEE Open J. Commun. Soc.* **2023**, *4*, 1001–1039. [CrossRef]
16. Shen, X.; Gao, J.; Wu, W.; Li, M.; Zhou, C.; Zhuang, W. Holistic Network Virtualization and Pervasive Network Intelligence for 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1–30. [CrossRef]
17. Baccour, E.; Mhaisen, N.; Abdellatif, A.A.; Erbad, A.; Mohamed, A.; Hamdi, M.; Guizani, M. Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2366–2418. [CrossRef]
18. Saad, W.; Bennis, M.; Chen, M. A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Netw.* **2020**, *34*, 134–142. [CrossRef]
19. Sundarum, M. Distributed Compute and Communications in 5G. 5G Americas 2022. Available online: https://www.5gamericas.org/distributed-compute-and-communication-in-5g/ (accessed on 1 July 2023).
20. Letaief, K.B.; Chen, W.; Shi, Y.; Zhang, J.; Zhang, Y.-J.A. The Roadmap to 6G: AI Empowered Wireless Networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
21. Li, Q.; Ding, Z.; Tong, X.; Wu, G.; Stojanovski, S.; Luetzenkirchen, T.; Kolekar, A.; Bangolae, S.; Palat, S. 6G Cloud-Native System: Vision, Challenges, Architecture Framework and Enabling Technologies. *IEEE Access* **2022**, *10*, 96602–96625. [CrossRef]
22. Wang, C.-X.; Di Renzo, M.; Stanczak, S.; Wang, S.; Larsson, E.G. Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges. *IEEE Wirel. Commun.* **2020**, *27*, 16–23. [CrossRef]
23. Liu, G.; Huang, Y.; Dong, J.; Jin, J.; Wang, Q.; Li, N. Vision, requirements and network architecture of 6G mobile network beyond 2030. *China Commun.* **2020**, *17*, 92–104. [CrossRef]
24. Ahammed, T.B.; Patgiri, R.; Nayak, S. A vision on the artificial intelligence for 6G communication. *ICT Express* **2023**, *9*, 197–210. [CrossRef]
25. Mahmood, M.R.; Matin, M.A.; Sarigiannidis, P.; Goudos, S.K. A Comprehensive Review on Artificial Intelligence/Machine Learning Algorithms for Empowering the Future IoT Toward 6G Era. *IEEE Access* **2022**, *10*, 87535–87562. [CrossRef]
26. Shahraki, A.; Ohlenforst, T.; Kreyß, F. When machine learning meets Network Management and Orchestration in Edge-based networking paradigms. *J. Netw. Comput. Appl.* **2023**, *212*, 103558. [CrossRef]
27. Zappone, A.; Di Renzo, M.; Debbah, M. Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both? *IEEE Trans. Commun.* **2019**, *67*, 7331–7376. [CrossRef]
28. Mao, Q.; Hu, F.; Hao, Q. Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2595–2621. [CrossRef]
29. Zhang, C.; Patras, P.; Haddadi, H. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
30. Lei, L.; Tan, Y.; Zheng, K.; Liu, S.; Zhang, K.; Shen, X. Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1722–1760. [CrossRef]
31. Agarwal, B.; Togou, M.A.; Ruffini, M.; Muntean, G.-M. A Comprehensive Survey on Radio Resource Management in 5G HetNets: Current Solutions, Future Trends and Open Issues. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2495–2534. [CrossRef]
32. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [CrossRef]
33. Ding, H.; Zhao, F.; Tian, J.; Li, D.; Zhang, H. A deep reinforcement learning for user association and power control in heterogeneous networks. *Ad Hoc Netw.* **2020**, *102*, 102069. [CrossRef]
34. Nie, H.; Li, S.; Liu, Y. Multi-Agent Deep Reinforcement Learning for Resource Allocation in the Multi-Objective HetNet. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2021; pp. 116–121.
35. Elsayed, M.; Erol-Kantarci, M.; Kantarci, B.; Wu, L.; Li, J. Low-Latency Communications for Community Resilience Microgrids: A Reinforcement Learning Approach. *IEEE Trans. Smart Grid* **2020**, *11*, 1091–1099. [CrossRef]
36. Naderializadeh, N.; Sydir, J.J.; Simsek, M.; Nikopour, H. Resource Management in Wireless Networks via Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3507–3523. [CrossRef]
37. Zhao, N.; Liang, Y.-C.; Niyato, D.; Pei, Y.; Jiang, Y. Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
38. Giannopoulos, A.; Spantideas, S.; Kapsalis, N.; Gkonis, P.; Sarakis, L.; Capsalis, C.; Vecchio, M.; Trakadas, P. Supporting Intelligence in Disaggregated Open Radio Access Networks: Architectural Principles, AI/ML Workflow, and Use Cases. *IEEE Access* **2022**, *10*, 39580–39595. [CrossRef]
39. Iturria-Rivera, P.E.; Zhang, H.; Zhou, H.; Mollahasani, S.; Erol-Kantarci, M. Multi-Agent Team Learning in Virtualized Open Radio Access Networks (O-RAN). *Sensors* **2022**, *22*, 5375. [CrossRef] [PubMed]
40. Zhang, H.; Zhou, H.; Erol-Kantarci, M. Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 958–963.
41. Nomikos, N.; Zoupanos, S.; Charalambous, T.; Krikidis, I. A Survey on Reinforcement Learning-Aided Caching in Heterogeneous Mobile Edge Networks. *IEEE Access* **2022**, *10*, 4380–4413. [CrossRef]

42. Zhang, T.; Fang, X.; Wang, Z.; Liu, Y.; Nallanathan, A. Stochastic Game Based Cooperative Alternating Q-Learning Caching in Dynamic D2D Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 13255–13269. [CrossRef]
43. Chen, S.; Yao, Z.; Jiang, X.; Yang, J.; Hanzo, L. Multi-Agent Deep Reinforcement Learning-Based Cooperative Edge Caching for Ultra-Dense Next-Generation Networks. *IEEE Trans. Commun.* **2021**, *69*, 2441–2456. [CrossRef]
44. Jiang, W.; Feng, G.; Qin, S.; Liu, Y. Multi-Agent Reinforcement Learning Based Cooperative Content Caching for Mobile Edge Networks. *IEEE Access* **2019**, *7*, 61856–61867. [CrossRef]
45. Yu, Z.; Hu, J.; Min, G.; Lu, H.; Zhao, Z.; Wang, H.; Georgalas, N. Federated Learning Based Proactive Content Caching in Edge Computing. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
46. Zhao, L.; Ran, Y.; Wang, H.; Wang, J.; Luo, J. Towards Cooperative Caching for Vehicular Networks with Multi-level Federated Reinforcement Learning. In Proceedings of the ICC 2021—IEEE International Conference on Communications, Virtual, 14–23 June 2021; pp. 1–6.
47. Haibeh, L.A.; Yagoub, M.C.E.; Jarray, A. A Survey on Mobile Edge Computing Infrastructure: Design, Resource Management, and Optimization Approaches. *IEEE Access* **2022**, *10*, 27591–27610. [CrossRef]
48. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Netw.* **2019**, *33*, 156–165. [CrossRef]
49. Huang, X.; Zhang, K.; Wu, F.; Leng, S. Collaborative Machine Learning for Energy-Efficient Edge Networks in 6G. *IEEE Netw.* **2021**, *35*, 12–19. [CrossRef]
50. Ren, Y.; Sun, Y.; Peng, M. Deep Reinforcement Learning Based Computation Offloading in Fog Enabled Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4978–4987. [CrossRef]
51. Tuong, V.D.; Truong, T.P.; Nguyena, T.-V.; Noh, W.; Cho, S. Partial Computation Offloading in NOMA-Assisted Mobile-Edge Computing Systems Using Deep Reinforcement Learning. *IEEE Internet Things J.* **2021**, *8*, 13196–13208. [CrossRef]
52. Goudarzi, M.; Palaniswami, M.S.; Buyya, R. A Distributed Deep Reinforcement Learning Technique for Application Placement in Edge and Fog Computing Environments. *IEEE Trans. Mob. Comput.* **2023**, *22*, 2491–2505. [CrossRef]
53. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [CrossRef]
54. Ssengonzi, C.; Kogeda, O.P.; Olwal, T.O. A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization. *Array* **2022**, *14*, 100142. [CrossRef]
55. Phyu, H.P.; Naboulsi, D.; Stanica, R. Machine Learning in Network Slicing—A Survey. *IEEE Access* **2023**, *11*, 39123–39153. [CrossRef]
56. Mason, F.; Nencioni, G.; Zanella, A. A Multi-Agent Reinforcement Learning Architecture for Network Slicing Orchestration. In Proceedings of the 2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet), Ibiza, Spain, 15–17 June 2021; pp. 1–8.
57. Chergui, H.; Blanco, L.; Garrido, L.A.; Ramantas, K.; Kuklinski, S.; Ksentini, A.; Verikoukis, C. Zero-Touch AI-Driven Distributed Management for Energy-Efficient 6G Massive Network Slicing. *IEEE Netw.* **2021**, *35*, 43–49. [CrossRef]
58. Liu, Q.; Choi, N.; Han, T. OnSlicing: Online End-to-End Network Slicing with Reinforcement Learning. In Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies, Munich, Germany, 7–10 December 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 141–153.
59. Kim, Y.; Lim, H. Multi-Agent Reinforcement Learning-Based Resource Management for End-to-End Network Slicing. *IEEE Access* **2021**, *9*, 56178–56190. [CrossRef]
60. Mai, T.; Yao, H.; Zhang, N.; He, W.; Guo, D.; Guizani, M. Transfer Reinforcement Learning Aided Distributed Network Slicing Optimization in Industrial IoT. *IEEE Trans. Ind. Inform.* **2022**, *18*, 4308–4316. [CrossRef]
61. Sheraz, M.; Ahmed, M.; Hou, X.; Li, Y.; Jin, D.; Han, Z.; Jiang, T. Artificial Intelligence for Wireless Caching: Schemes, Performance, and Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 631–661. [CrossRef]
62. Wijethilaka, S.; Liyanage, M. Survey on Network Slicing for Internet of Things Realization in 5G Networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 957–994. [CrossRef]
63. Chafii, M.; Bariah, L.; Muhaidat, S.; Debbah, M. Twelve Scientific Challenges for 6G: Rethinking the Foundations of Communications Theory. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 868–904. [CrossRef]
64. Feriani, A.; Hossain, E. Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1226–1252. [CrossRef]
65. Soto, P.; Camelo, M.; De Vleeschauwer, D.; De Bock, Y.; Chang, C.-Y.; Botero, J.F.; Latré, S. Network Intelligence for NFV Scaling in Closed-Loop Architectures. *IEEE Commun. Mag.* **2023**, *61*, 66–72. [CrossRef]
66. Khan, L.U.; Saad, W.; Niyato, D.; Han, Z.; Hong, C.S. Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions. *IEEE Commun. Mag.* **2022**, *60*, 74–80. [CrossRef]
67. Shen, Y.; Shi, Y.; Zhang, J.; Letaief, K.B. Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 101–115. [CrossRef]
68. Rusek, K.; Suarez-Varela, J.; Almasan, P.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2260–2270. [CrossRef]

69. Wang, H.; Wu, Y.; Min, G.; Miao, W. A Graph Neural Network-Based Digital Twin for Network Slicing Management. *IEEE Trans. Ind. Inform.* **2020**, *18*, 1367–1376. [CrossRef]
70. He, S.; Xiong, S.; Ou, Y.; Zhang, J.; Wang, J.; Huang, Y.; Zhang, Y. An Overview on the Application of Graph Neural Networks in Wireless Networks. *IEEE Open J. Commun. Soc.* **2021**, *2*, 2547–2565. [CrossRef]
71. Tam, P.; Song, I.; Kang, S.; Ros, S.; Kim, S. Graph Neural Networks for Intelligent Modelling in Network Management and Orchestration: A Survey on Communications. *Electronics* **2022**, *11*, 3371. [CrossRef]
72. Chen, M.; Gunduz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [CrossRef]
73. Muscinelli, E.; Shinde, S.S.; Tarchi, D. Overview of Distributed Machine Learning Techniques for 6G Networks. *Algorithms* **2022**, *15*, 210. [CrossRef]
74. Hosseinalipour, S.; Brinton, C.G.; Aggarwal, V.; Dai, H.; Chiang, M. From Federated to Fog Learning: Distributed Machine Learning over Heterogeneous Wireless Networks. *IEEE Commun. Mag.* **2020**, *58*, 41–47. [CrossRef]
75. James, G.; Witten, D.; Hastie, T.; Tibshirani, R.; Taylor, J. *An Introduction to Statistical Learning*; Springer Texts in Statistics; Springer International Publishing: Cham, Switzerland, 2023; ISBN 978-3-031-38746-3.
76. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.