ISSN 2770-3541 (Online)
Volume 1 | Issue 1 | October 2021

Intelligence & Robotics

Inaugural issue officially released!

Editorial
Intelligence and robotics
Simon X. Yang

Perspective
Intelligent robotics - misconceptions, current trends, and opportunities
Clarence W. de Silva

Review
Federated reinforcement learning: techniques, applications, and open challenges
Lei Lei, et al.

Bio-inspired intelligence with applications to robotics: a survey
Simon X. Yang, et al.

Research article
Unsupervised monocular depth estimation with aggregating image features and wavelet SSIM (Structural SIMilarity) loss
Hao Zhang, et al.

Open Access

www.intellrobot.com

Editor-in-Chief

Simon X. Yang

Prof. Simon X. Yang is currently the Head of the Advanced Robotics and Intelligent Systems Laboratory at the University of Guelph. His research interests include artificial intelligent, robotics, sensors and multi-sensor fusion, wireless sensor networks, control systems, bio-inspired intelligence, machine learning, neural networks, fuzzy systems, and computational neuroscience.

Our Features

(1) Gold Open Access
(2) Strong Editorial Board
(3) Rigorous Peer-review
(4) Free English Ianguage Editing Service
(5) Online First Once Accepted
(6) Free Publication Before 31 Dec 2024
(7) Wide Promotion (Twitter\LinkedIn\WeChat\Facebook)

Editorial Board

- 1 Editor-in-Chief
- 5 Advisory Editorial Members
- 24 Associate Editors
- 15 Youth Editorial Board Members

Scope

Top-quality unpublished original technical and non-technical application-focused articles are welcome from intelligence and robotics, particularly on the inter-disciplinary areas of intelligence and robotics, including but not limited to the following areas:

- biological, bio-inspired, and artificial intelligence;
- neural networks, fuzzy systems, and evolutionary algorithms;
- sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems;
- robot cooperation, teleoperation, and human-machine interactions.
- development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Journal Home
https://intellrobot.com/

Submission Link
https://oaemesas.com/login?JournalId=ir

EDITORIAL BOARD

GENERAL INFORMATION

About the Journal

Intelligence & Robotics (*IR*), ISSN 2770-3541 (Online), publishes top-quality unpublished original technical and non-technical application-focused articles on intelligence and robotics, particularly on the interdisciplinary areas of intelligence and robotics. The Journal seeks to publish articles that deal with the theory, design, and applications of intelligence and robotics, ranging from software to hardware. The scope of the Journal includes, but is not limited to, biological, bio-inspired, and artificial intelligence; neural networks, fuzzy systems, and evolutionary algorithms; sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems; and robot cooperation, teleoperation and human-machine interactions. The Journal would be interested in distributing development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Information for Authors

Manuscripts should be prepared in accordance with Author Instructions.
Please check https://intellrobot.com/pages/view/author_instructions for details.
All manuscripts should be submitted online at https://oaemesas.com/login?JournalId=ir.

Copyright

Permissions

For information on how to request permissions to reproduce articles/information from this journal, please visit www.intellrobot.com.

Disclaimer

The information and opinions presented in the journal reflect the views of the authors and not of the journal or its Editorial Board or the Publisher. Publication does not constitute endorsement by the journal. Neither the *IR* nor its publishers nor anyone else involved in creating, producing or delivering the *IR* or the materials contained therein, assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information provided in the *IR*, nor shall they be liable for any direct, indirect, incidental, special, consequential or punitive damages arising out of the use of the *IR*. *IR*, nor its publishers, nor any other party involved in the preparation of material contained in the *IR* represents or warrants that the information contained herein is in every respect accurate or complete, and they are not responsible for any errors or omissions or for the results obtained from the use of such material. Readers are encouraged to confirm the information contained herein with other sources.

Published by

OAE Publishing Inc.
245 E Main Street Ste 107, Alhambra CA 91801, USA
Website: www.oaepublish.com

Contacts

E-mail: editorial@intellrobot.com
Website: www.intellrobot.com

Intelligence & Robotics

CONTENTS

Volume 1, Issue 1

Yang. *Intell Robot* 2021;1(1):1-2
DOI: 10.20517/ir.2021.12

Intelligence & Robotics

Editorial

Open Access

Intelligence and robotics

Simon X. Yang

Advanced Robotics & Intelligent Systems Laboratory, School of Engineering, University of Guelph, Guelph ON N1G 2W1, Canada.

Correspondence to: Prof. Simon X. Yang, Advanced Robotics & Intelligent Systems Laboratory, School of Engineering, University of Guelph, 50 Stone Road East, Guelph ON N1G 2W1, Canada. E-mail: syang@uoguelph.ca

How to cite this article: Yang SX. Intelligence and robotics. *Intell Robot* 2021;1(1):1-2. https://dx.doi.org/10.20517/ir.2021.12

Received: 28 Sep 2021 **Accepted:** 28 Sep 2021 **Available online:** 9 Oct 2021

Academic Editor: Hao Zhang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Welcome to the inaugural issue of *Intelligence & Robotics*. It is my great pleasure and honor to be involved as the Editor-in-Chief and founder of the new journal *Intelligence & Robotics* . We are delighted with and highly grateful to the many colleagues and friends with diversified expertise in intelligence and robotics worldwide to support the Journal and serve on our editorial board, including advisory board members, associate editors, and youth editorial board members. We are also immensely grateful for the tremendous support from our publisher OAE Publishing Inc., USA.

Biological inspiration provides the basis for many aspects of robotics. Robot manipulators were first developed to approximate the reaching and grasping abilities of the human arm. Walking machines were attempts to perform some of the locomotion and gait features of living things. Studies of intelligence have made significant progress in understanding the biological intelligence of various species and developing innovative artificial and bionic strategies, mechanisms, algorithms, and technologies, with diversified applications to various fields, particularly robotics. On the other hand, robotics studies have made remarkable progress in theoretical investment and real-world applications to various industries. There is a general movement toward service-oriented intelligent robotic systems that require the ability to adapt to complex dynamic situations and handle various uncertainties, such as self-driving cars. Intelligence will be essential to these robotic systems performing successfully, as living organisms can adapt to changing environments that are only partially known and not predictable. Therefore, it is essential to bring experts from the fields of intelligence and robotics together to accomplish original and innovative achievements.

Intelligence & Robotics publishes top-quality unpublished original technical and non-technical application-focused articles on the general areas of intelligence and robotics, particularly the interdisciplinary area of intelligence and robotics. The Journal seeks to publish articles dealing with the theory, design, analysis, evaluation, and applications of intelligence and robotics, ranging from system modeling, data analysis, algorithms, and software/hardware development of various intelligent and/or robotic systems.

The scope of the Journal includes, but is not limited to, (1) the biological, bio-inspired, and artificial intelligence, such as neural networks, fuzzy systems, evolutionary algorithms, genetic algorithms, ant colony optimization, particle swarm optimization, artificial immune systems, simulated annealing, smart gambler strategy, expert systems, and various other intelligent methodologies; (2) the design, modeling, analysis, evaluation, and implementation of various robotic systems, such as mobile, aerial, surface, and underwater robotic systems; (3) the real-time information acquisition, multi-sensor fusion, data analysis, localization and map building, path planning, tracking, and control for various robotic systems; and (4) the cooperation, coordination, communication, teleoperation, and human-machine interactions of multiple robotic systems. In addition, the Journal would be interested in distributed development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

This journal aims to provide a platform for all experts, professionals, and scholars with creative contributions to get together and share some inspiring ideas and accomplish outstanding achievements in the general fields of intelligence and robotics.

DECLARATIONS

Authors' contributions

The author contributed solely to the article.

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

The author declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

de Silva. *Intell Robot* 2021;1(1):3-17
DOI: 10.20517/ir.2021.01

Intelligence & Robotics

Perspective

Open Access

Check for updates

Intelligent robotics - misconceptions, current trends, and opportunities

Clarence W. de Silva

Department of Mechanical Engineering, University of British Columbia, Vancouver BC V6T 1Z4, Canada.

Correspondence to: Prof. Clarence W. de Silva, Department of Mechanical Engineering, University of British Columbia, 6250 Applied Science Lane, Vancouver BC V6T 1Z4, Canada. E-mail: desilva@mech.ubc.ca

How to cite this article: de Silva CW. Intelligent robotics - misconceptions, current trends, and opportunities. *Intell Robot* 2021;1(1):3-17. https://dx.doi.org/10.20517/ir.2021.01

Received: 25 Jul 2021 **First Decision:** 21 Jul 2021 **Revised:** 24 Jul 2021 **Accepted:** 28 Jul 2021 **Published:** 11 Oct 2021

Academic Editor: Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

The concepts of "Robots" have been of interest to humans from historical times, initially with the desire to create "artificial slaves". Since the technology was not developing to keep up with the "dreams", initially, Robotics was primarily of entertainment value, relegated to plays, movies, stories, *etc*. The practical applications started in the late 1950s and the 1960s with the development of programmable devices for factories and assembly lines as flexible automation. However, since the expectations were not adequately realized, the general enthusiasm and funding for Robotics subsided to some extent. With subsequent research, developments, and curricular enhancement in Engineering and Computer Science and the resurgence of Artificial Intelligence, particularly machine learning, Robotics has found numerous practical applications today, in industry, medicine, household, the service sector, and the general society. Important developments and practical strides are being made, particularly in Soft Robotics, Mobile Robotics (Aerial - drones, Underwater, Ground-based - autonomous vehicles in particular), Swarm Robotics, Homecare, Surgery, Assistive Devices, and Active Prosthesis. This perspective paper starts with a brief history of Robotics while indicating some associated myths and unfair expectations. Next, it will outline key developments in the area. In particular, some important practical applications of Intelligent Robotics, as developed by groups worldwide, including the Industrial Automation Laboratory at the University of British Columbia, headed by the author, are indicated. Finally, some misconceptions and shortcomings concerning Intelligent Robotics are pointed out. The main shortcomings concern the mechanical capabilities and the nature of intelligence. The paper concludes by mentioning future trends and key opportunities available in Intelligent Robotics for both developed and developing counties.

Keywords: Robotics, characteristics of intelligence, machine learning, shortcomings of intelligent robotics, technical needs, opportunities in intelligent robotics

1. INTRODUCTION

Commonly, a robot is considered a machine that can perform work or actions normally performed by humans, automatically or by remote control - Teleoperation. The key features of this definition are the presence of: (1) Mechanical Structure (Machine); (2) Sensors; (3) Actuators (or Effectors); and (4) Controller (or Computer), which is the brain or the decision-maker of the robot. Essentially, a digital computer serves as this "brain" of the robot, and it has to be programmed to carry out its actions. Therefore, the "intelligence" of a robot needs to be incorporated there. As well, there have to be sensors to monitor the operation. The sensory data are processed by the computer to determine the underlying situations and the suitable robotic actions.

The concepts of "Robots" have been of interest to humans from historical times, initially with the desire to create "artificial slaves". The term "Robot" was introduced in the popular media, well before a physical robot became a reality. In 1920, Czech writer Karel Capek first introduced the term in his play "RUR" or "Rossum's Universal Robots". There, it was just a figment of his imagination. Again, in 1942, the Russian-born American science-fiction writer and Boston University Professor Isaac Asimov introduced the term in his fiction. Notably, Asimov was one of the "Big Three" in science fiction. The other two were the Sri Lanka-based late Sir Arthur C. Clarke and Robert A. Heinlein. We know that many predictions of Clarke and Asimov have come true today. A device resembling a humanoid robot was designed and built by the ingenious Leonardo Da Vinci in 1945. It could mechanically move arms, head, and jaw but was not a true robot in today's definition. The first true robot arm, the *Unimate*, was designed by the American inventor George Devol in collaboration with Joseph Engelberger, who is often called the "Father of Robotics". This robot was used in a General Motors (automotive) plant for its manufacturing operations in 1960. It had a primitive digital computer as its brain and used motion sensors and also dc motors as the actuators[1].

Many different types of robots have been developed and put into operation since. Some that we see in the popular media are, however, computer animations rather than "intelligent" robots. Since the technology was not developing to keep up with the "dreams", initially, Robotics was primarily of entertainment value, relegated to plays, movies, stories, *etc*. The practical applications started in the late 1950s and the 1960s with the development of programmable devices for factories and assembly lines as flexible automation. As an example of the application of Robotics in flexible automation, consider the welding robots in an automotive plant[2]. Here, the vehicle model that is being manufactured can be changed very easily and quickly, simply by changing the program. However, the operation itself is not fast, albeit quite complex. However, since the expectations were not adequately realized, the general enthusiasm and funding for Robotics subsided to some extent. With subsequent research, development, and curricular enhancement in Engineering and Computer Science and the resurgence of Artificial Intelligence (AI), particularly machine learning, Intelligent Robotics has found numerous practical applications today in industry, medicine, household, and the general society. Important developments and practical strides are being made, particularly in Soft Robotics, Mobile Robotics (Aerial - drones, Underwater, and Ground-based - autonomous vehicles in particular), Swarm Robotics, Homecare, Surgery, Assistive Devices, and Active Prosthesis.

2. THE STATE-OF-THE-ART

Many different types of robots are available today. A robot that has a human-like body structure is called a "humanoid". An example is the Honda Asimo[3]. However, a robot need not look like a human. An

Unmanned Naval Vehicle that we developed in Figure 1 in our laboratory (Industrial Automation Laboratory or IAL at the University of British Columbia or UBC) is indeed a robot. It has a programmable propelling platform with multiple sensors (to measure the pH value, dissolved oxygen, electrical conductivity, temperature, and the oxidation-reduction potential of water). It is able to autonomously navigate in a water body and map out the quality of the water in a particular region, which can be used to provide warnings to the users, determine the source of pollution or contamination, and also suggest to the relevant authorities suitable corrective actions[4].

Unmanned aerial vehicles (UAVs) or drones are other types of mobile robots. A UAV that is used in our laboratory is shown in Figure 2. A similar UAV is used in our collaborative project in environmental monitoring and field mapping[5]. Due to their capabilities of autonomous navigation in extensive and traditionally unreachable regions and advanced sensing (such as optical and thermal imaging and laser scanning), UAVs are replacing costly, hazardous, and manual ways of inspection and surveillance in various applications. For example, when compared with the manual and ground-based approaches, UAVs are able to capture more useful and detailed information of petrochemical operations, including wellheads, pipelines, boiler systems, tanks, refineries, and furnaces, in a consistent, continuous, accurate, fast, and repeated manner, even in environments where human operators find it time-consuming, difficult and hazardous to function. The data/information collected in this manner can be further processed in a ground-based control platform and used to localize faults and malfunctions in the facility. Based on that information, a UAV is able to deploy the needed equipment, including robotic devices, to that location for detailed diagnosis and possible correction of the problem.

As another example, consider the smart prosthetic limb[6] shown in Figure 3. It is able to adapt to different users and conditions of the terrain intelligently. Also, it uses expectational knowledge during walking, for example, to decide whether to step over an obstacle or to avoid it.

2.1. Common myths of robotics

There have been many myths, misnomers, and misunderstandings surrounding robots. Some are:

1. Robots have capabilities that equate to or exceed those of humans. This is not true. The intelligence of today's robots does not exceed even that of a dog! It is inconceivable that robot intelligence can reach that of human intelligence.
2. Robots will steal our spouses, and fight wars, and defeat us. This is very unrealistic and will never happen!
3. Robots will create mass unemployment: people said this concerning the first Industrial Revolution as well. However, in the end, the industrial revolution (and industrial automation) gave the workers more free time (including the five-day workweek), moved them away from hazardous and difficult work environments, improved the general quality of life, and created higher-paying and more challenging employment (for our friends and relative, if not for us).

While many positive things can be said about robotization, the reasons for the underlying misunderstandings are several. They include fantasy, movies, and popular/social media, unrealistic expectations by the robot enthusiasts, slow developments in the field, inadequate expertise and capabilities for the necessary developments, the lack of necessary technologies (mechanical, electronic, and computer science, *etc.*) to achieve the goals, weak collaboration in the beginning (mainly among the electrical and mechanical engineers, and computer scientists, but this culture has been changed with the establishment of leading robotics institutes); and the lack of adequate "robotic intelligence" for various autonomous operations. See the related video interview of the author[7].

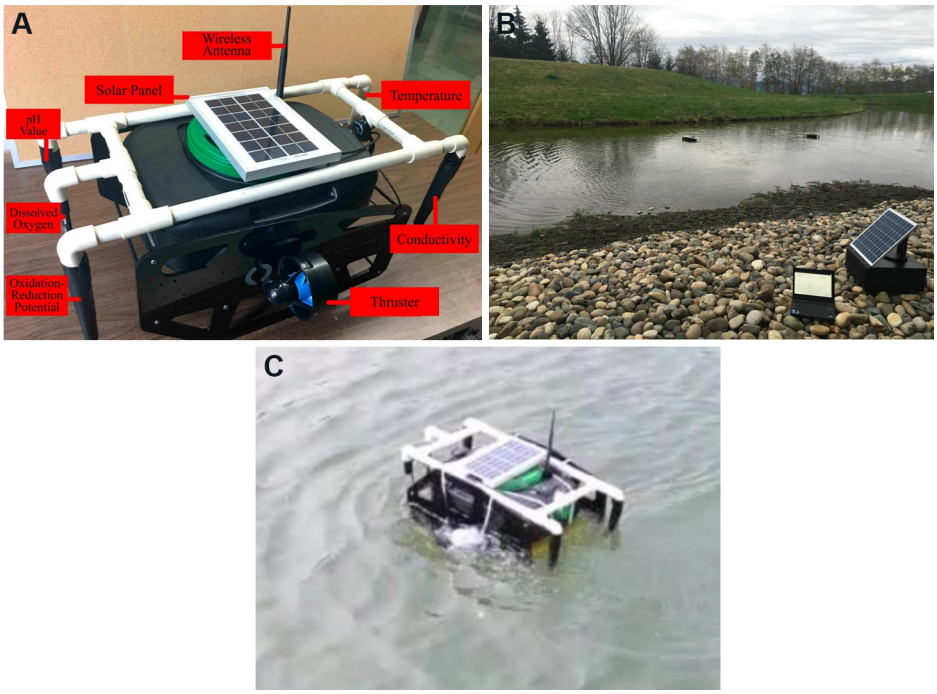


Figure 1. An autonomous naval multi-sensor module that monitors and maps the quality of water. (A) Components of the Unmanned Naval Vehicle; (B) Multi-module deployment in Vancouver; (C) Deployment in a river in India.

Figure 2. An unmanned aerial vehicle that is used in our laboratory.

3. SHORTCOMINGS AND NEEDS

What the engineers and technologists reasonably expect in Robotics has not been realized yet. The common capabilities of the existing robots include navigation with obstacle avoidance (SLAM - Simultaneous Localization and Mapping), visual and verbal communication with humans, operation of some appliances, grasping and carrying of objects (including conformable grasping and tactile sensing), multi-robot cooperation, and haptic teleoperation (with force feedback). However, some obvious shortcomings of today's robots include poor human-like interaction (and poor interaction with humans), slow speed, poor dexterity, and the sequential nature in grasping and handling of objects (*e.g.*, the robotic hand slowly moves

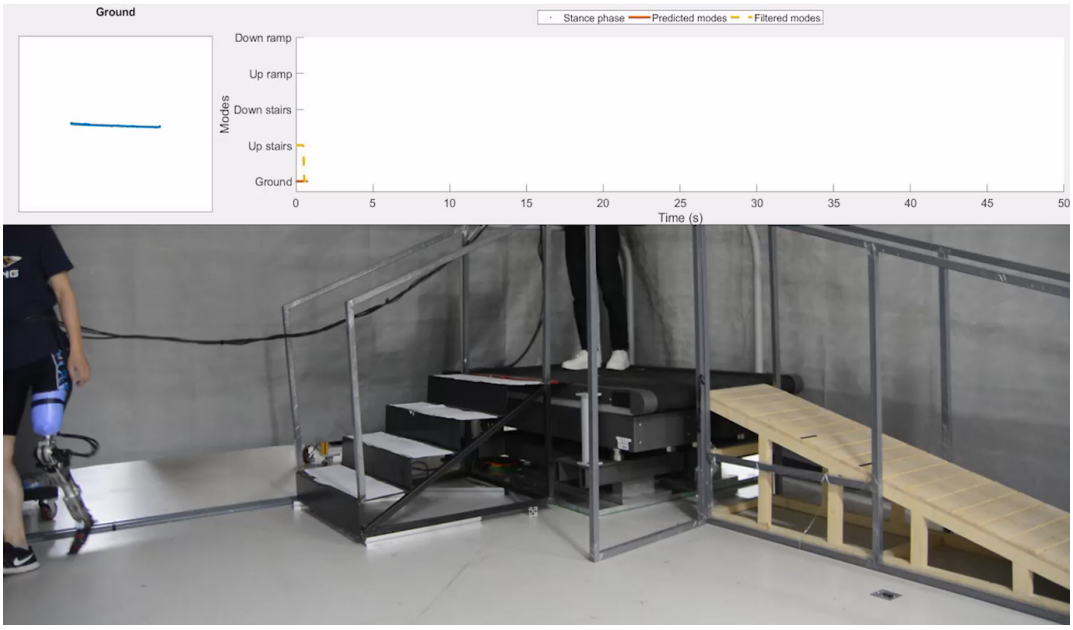


Figure 3. A smart prosthetic limb.

to the object and subsequently the fingers are closed around the object. Instead, in the case of a human, they observe the object to be grasped and simultaneously move the hand and the body in coordination until the object is reached and the grasping is completed. Also, a human can easily sense possible slipping of the object before it occurs), possible safety problems for humans (due to the shortcomings of the current robotic mechanical components and control), sensory limitations (particularly related to smell and taste, or chemical and biological sensing; transparency in teleoperation), limited mechanical capability (in dexterity, flexibility, adaptability, *etc*., unlike humans), and limited robotic intelligence. Concerning the safety in human-robot interaction, robot-inflicted injuries include accidents involving: sharp objects and tools, large forces, fast motions and quick changes of magnitude and direction, and malfunctions in the robotic equipment.

The autonomous operation (*i.e.*, operating on its own, without outside help) of robots is essential in Intelligent Robotics. Some needs for this autonomy are greater robotic intelligence (better learning; operating in dynamic, partially structured, and partially known environments; the capability of enhanced characteristics of intelligence); greater accuracy, speed, dexterity, *etc.*; increased safety (better object handling, accident/obstacle avoidance, *etc.*); more human-friendly and human-like communication and operation; and the redesigning of household appliances for easy operation by robots (and humans).

Consider the required basic caregiver tasks (of humans), for example, verbal and visual communication; assistance for movement/mobility; identifying, grasping, and handling of needed objects properly and safely; safe and quick assistance in the mobility of the care-receiver, in the presence of obstacles (both static and dynamic such as furniture, appliances, other humans and pets); monitoring of objects and the environment for carrying out the caregiver tasks (under normal and emergencies); operating household appliances; and the provision of assistive devices. Similar operational requirements from a robotic caregiver include faster yet safe operation; human-friendly and human-like interaction and communication; autonomous assistance for 24 h, routine and basic care (mobility, bathing, dressing, toileting, meal preparation, providing medicine, *etc.*); effective monitoring and detection of emergencies; and adequate

emergency assistance (possibly incorporating remote monitoring, teleoperation, *etc.*) until regular help arrives. In this context, some needs in haptic teleoperation (teleoperation with feedback from the slave robot to the human master) are improvements to autonomous robotics as in non-teleoperation situations; improved transparency (better/faster tactile/visual/auditory feedback to the remote human operator, for realistic creation of remote presence); stability under (and compensation for) time delays (which are common in teleoperation[8]; human-like manipulation; improved design and control (for accuracy, speed, robustness, reliability, and safety); and 3D virtual reality for the remote operator (for improved transparency).

3.1. Possible directions of advancement

The technology focus may be directed on several aspects to improve the state-of-the-art of Intelligent Robotics. They include autonomy, improved learning and intelligence (for autonomous operation); self-awareness for robots (*i.e.*, knowing the own capabilities of the robot); improved dexterity of handling (*e.g.*, compliant grasping, parallel not sequential, and incoordination; providing the adequate degrees of freedom for manipulation of the handled object); improved robot-human interaction (in particular, working "with" a human rather than working "for" a human); improved speed, stability, robustness, reliability, and safety; improved sensing (particularly, chemical and biological sensing; transparency of remote operations; dynamic sensor networks; intelligent sensor fusion); and significant improvement of the "mechanical" capabilities. Note particularly the capabilities that require "intelligence", software, mechanical capabilities, and instrumentation.

In this context, a question can be posed whether to direct much effort in developing universal robots that have unlimited capabilities and functions, which will, of course, be very costly and complex as well. In other words, should the focus be directed on the development of very complex and costly multi-purpose robots or use existing single-purpose robots cooperatively? In fact, it is not wise to put much effort into the development of complex and costly robots with numerous features and multi-function capabilities. As a scenario, consider the use of existing low-cost robots that have been developed for just one specific task (*e.g.*, security, human assistance, and guidance, street cleaning). If an emergency occurs (*e.g.*, an explosion), they may be called upon to join (if available) in cooperation, for example, put together simple devices and assist in the situation (*e.g.*, evacuation of the injured).

4. ROBOTIC INTELLIGENCE

The importance of intelligence in Robotics is quite clear. In particular, intelligence is essential for the autonomous operation of a robot. In fact, the realization of expectations (including some fantasies?) of intelligent robotics depends on improved robotic intelligence and similarly improved mechanical capability. Today's robots do not have even "primitive" human intelligence. Without significantly improved intelligence, robots cannot achieve human-like capabilities; for example, emotions, common sense, and inventiveness are rather farfetched! Improved intelligence renders the robots acquire some characteristics of human intelligence. A robot may be "trained" for a specific task (through methodologies of machine learning), but this is not the same as developing the robotic brain to reach the nature and capabilities of a human brain, at least at a very basic level. It is simply "fear-mongering" to say that the future robots will be a danger to humankind because of their high level of intelligence.

Primarily, robots improve their intelligence through learning, and the foundation of AI is indeed machine learning. Some claim that since a chess-playing computer has defeated a human champion, it is possible that intelligent robots will defeat humans in many human-centered activities. Here we have to realize that the capabilities of a robot depend on their mechanical capabilities and the control program (or brain), which is

developed by humans. It is true that due to learning, the particular robot intelligence (the decision-making ability related to the learned knowledge) improves. Unlike humans, whose level of intelligence can depreciate for many reasons (physiology, lack of practice, new knowledge, new and complex environments, *etc.*), machine learning will always improve robotic intelligence. This means a chess-playing robot will continuously improve its skills through learning (practice) and can thereby defeat a chess champion. However, we have to realize that such intelligence is provided to robots by humans through control programs. That program will never be able to acquire all the characteristics of a human brain. For instance, we may question whether a chess-playing robot can also perform other tasks (*e.g.*, caring for an elderly, carrying out medical surgery) unless specifically developed and trained for such activities and has the needed mechanical capabilities. Also, can a robot ever acquire such characteristics as common sense or emotions that are possessed by a human, in the same manner as in a human brain?

Humans develop robots, and we program their controllers (brains). We can set limits, checks and balances, regulations, and guidelines as we wish. We should collaborate with social scientists and develop proper guidelines and regulations for the development and the safety and ethical use of robots. Since a proper ethical evaluation and certification are essential for any technology that is used by humans, this should properly adhere for robots as well. In medical surgery, for example, a robot will facilitate the surgical procedures, but they should be performed under the supervision of a human surgeon, who must have the capability to abort the robotic procedure immediately, if necessary.

In fact, those who fear AI simply fear a black box! In order to make a proper determination, we should know what methodologies are used exactly in the AI black box and how those methodologies are implemented and operated. So, we should explore the black box carefully and in detail (with the help of experts who are knowledgeable in the subject) and only then indicate what methodologies in the AI black box might be dangerous. Then other experts will be able to respond intelligently and in an informative manner.

4.1. Characteristics of intelligence

Before exploring AI itself, let us examine intelligence. No precise definition exists for intelligence. They are the external characteristics and capabilities (that we observe from actions) that enable us to claim whether an entity (for example, a robot) is "intelligent". Essentially, the outward characteristics define intelligence.

The characteristics of intelligence include sensory perception; pattern recognition; learning (*i.e.*, knowledge acquisition, which is extremely important for intelligence); inference (*i.e.*, making decisions) from incomplete information; inference from qualitative or approximate information (this is commonly used in "qualitative reasoning" as in fuzzy logic or fuzzy reasoning); ability to deal with unfamiliar situations; adaptability to new, yet related situations (through "expectational knowledge". For example, a human is able to expect the nature of an environment, like a classroom, even when encountering that environment for the first time); inductive reasoning (people must have done this in high school mathematics when proving a mathematical result "by induction"); common sense; display of emotions; inventiveness; and self-awareness (*i.e.*, knowing their own capabilities).

A simplified model for the dynamics of intelligence is shown in Figure 4. The intelligent preprocessors are, in fact, learning modules. They enable one to gain knowledge by "learning" from information and also achieve expertise by further learning through knowledge (including practice). The achieved knowledge and expertise can depreciate for various reasons (including environmental and biological) and also can become outdated. Even though intelligent preprocessing or learning is vital in this model, it is unlikely that machine

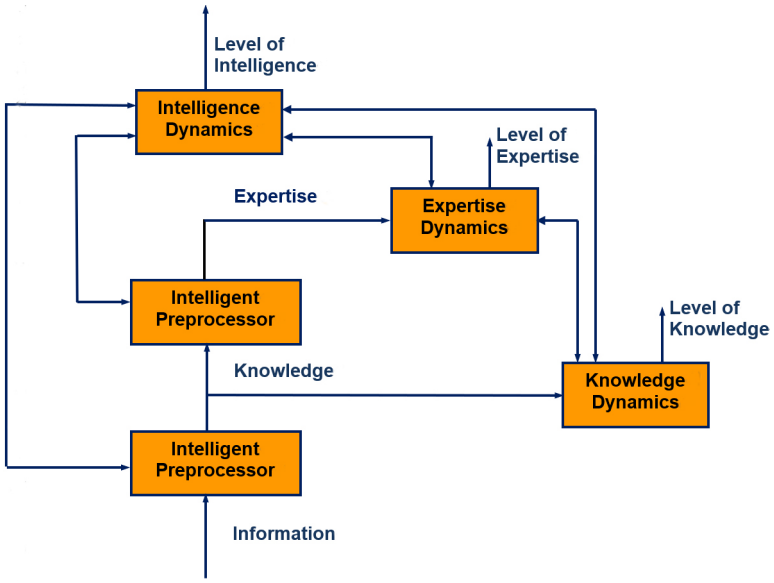


Figure 4. A model for the dynamics of intelligence.

learning alone will be able to achieve all forms of human intelligence in a robotic device.

4.2. Artificial intelligence

AI uses formal techniques to acquire some characteristics of intelligence. Models of AI are used for this purpose based on one or more of the mentioned characteristics. Such approaches (or models) of AI include Machine Learning, a very popular approach to AI. The conventional models of AI include knowledge-based systems, soft computing (consisting of neural networks - NN, fuzzy systems, and evolutionary computing; see[9] for instance), and swarm intelligence. A knowledge-based system typically consists of a knowledge base (or a rule base), a database, and an inference engine (the decision-maker). The decisions are made as follows: some data in the database (including what is acquired recently through sensors) is matched with the (context of the) rules in the knowledge base by the inference engine, and the inferences (or actions) are determined accordingly (*i.e.*, rules are fired). Popular "Expert Systems" are based on this model. Of course, the knowledge base will be improved and enhanced continuously through "learning" and experience (so machine learning is used here as well).

Deep learning is a popular approach to machine learning. It incorporates an intelligent and intensive method of learning and sophisticated computing power that is available with such advancements as graphic processing units and tensor processing units to process massive quantities of data efficiently. Deep learning need not be limited to the use of deep NN but is the current trend. Deep NN includes Convolutional NN or convolutional neural network (CNN)[4,10] (see Figure 5). They have a structure of multiple layers (convolution layers) incorporating the "dynamic" learning ability and ending with a "Softmax" layer, which is the classification layer. First, the NN is trained using "labeled data" (*i.e.*, input data whose proper outcomes are known *a priori*). Then, after the network is trained properly, unlabeled data (or new data) may be used for actual decision-making. Thus, massive amounts of data, including sensed data (a mixture of labeled and unlabeled data), may be effectively used in a deep NN.

Reinforcement learning relies on rewarding the correct decisions and penalizing the wrong decisions to learn the proper decision strategies. AI agents are capable of providing explanations for their decisions

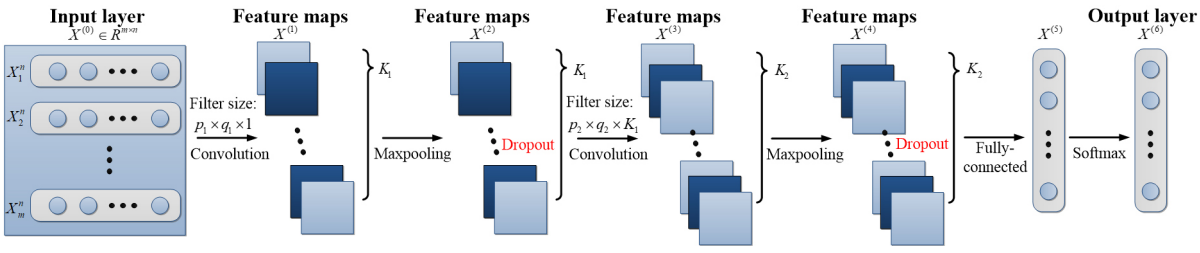


Figure 5. The architecture of a convolutional neural network.

(similar to what Expert Systems provide). In Edge AI, AI algorithms are processed locally on a hardware device. The algorithm uses data created on the device (*e.g.*, data generated by the algorithm) and other data (external data, including those from sensors and through the system interface). Hence, Edge AI functions at the "edge of the system network". Fuzzy logic attempts to be similar to human decision-making by incorporating "fuzzy" or "qualitative" or "approximate" data, such as those that use qualifiers like fast, small, better, and close. Qualitative or fuzzy reasoning is used in the decision-making (inference) process. Swarm Intelligence behaves like a swarm of animals or insects. They are distributed (not hierarchical) and interact with each other to learn and make decisions. The members in a swarm use very simple rules, yet leading to "intelligent" global behavior, even though an individual member is not quite intelligent, which will improve with time. Evolutionary computing relies on genetic algorithms or genetic computing to realize "optimized" behavior through learning. The basis of this methodology is biological evolution (or survival of the fittest).

Within AI, apart from "learning", other characteristics of intelligence need to be investigated and incorporated (*e.g.*, decision making with partial, approximate or qualitative information, use of "expectational knowledge", various approaches of reasoning such as inductive reasoning, ability to deal with unfamiliar situations, common sense, inventiveness, self-awareness, attention representation, and classification). Under machine learning itself, many methods exist, such as CNN, dynamic or recurrent neural networks (RNN), reinforcement learning, support vector machines, and entropy-based approaches. The rationalization of why a particular learning method is chosen (justification) should be a requirement in any application. Comparative evaluations of different methods should be carried out, with a proper reference (basis) for comparison. In this manner, comparative advantages and disadvantages of different methods should be determined, the rationale for choosing a particular approach for the application. When machine learning is applied in a particular situation, domain transformation or domain adaptability needs consideration because the domain of learning is typically not the same as the domain of application[11].

5. ROBOTIC CONTROL

It should be clear that proper control techniques are crucial for the effective operation of a robotic system. Typically, different types of multiple robots are used in practical applications. Then, networked and automated or autonomous operation of multiple robots, in a common, self-adaptive, and intelligent system architecture, implemented on a common platform, with resource sharing, has to be implemented.

The networked operation of multiple robots and other agents (sensors, actuators, controllers, and other devices) is not new. Furthermore, system optimization, intelligent systems, and adaptive control have been extensively investigated and applied by us and others. In this backdrop and the strong foundation of prior work, the networked implementation of multiple robots (and other agents) may focus on the following aspects:

1. Some networked agents (*e.g.*, robots, unmanned aerial vehicles or UAVs, and sensor nodes containing sensors, actuators, effectors, controllers, *etc.*) may be dynamic or mobile.
2. The operating system environment may be dynamic, unstructured, and unknown.
3. The system should be self-adaptive to optimize its performance, particularly by utilizing the dynamic components in addition to parameter adjustment or tuning and structural reorganization.
4. The system may be further optimized by sharing resources among the applications.
5. Dynamic or mobile sensors may receive "feedback" from themselves to improve their sensing effectiveness (*e.g.*, data/information quality, the relevance of their data, speed, and confidence).
6. The networked agents should possess "intelligence" to facilitate autonomous and desired performance.
7. The system should be able to predict, detect, and diagnose malfunctions and faults in it and accommodate or self-repair.

The underlying activities of system development and implementation will pertain to sensor/data fusion and adaptive sensing; multi-agent cooperation; multi-objective and parameter/structure optimization; fault prediction, detection, diagnosis, and resolution; self-organization/adaptation; and distributed/networked intelligent control. Suitable system architecture and an application platform of this type are schematically shown in Figure 6. In this system development process, one may have to determine and quantify the design constraints, performance limits, trade-offs, and development/operation guidelines and benchmarks for the pertinent applications. That will lead to significant improvements in performance, developmental and operational costs, productivity, resource requirements, energy efficiency, safety, fault tolerance, reliability, autonomy, and sustainability of the robotic system.

It is clear that both individual and network control are relevant in the present context, and furthermore, both conventional control and "intelligent" control are also relevant. The present paper has devoted much focus to the aspect of robotic intelligence. Hence, in the present section, particular attention is given to the conventional control of robots.

5.1. Conventional control

The development and application of conventional control have been extended wide effort by many. The relevant techniques include the following.

Feedback control and particularly servo-control of robotic joints had been the main focus in the early developments of robotic control. Here, the robot motions are measured (sensed) and used by the controller in feedback to move the robot in the desired manner. Thus, a robot is "servoed" along a specified motion trajectory through feedback control of the motion error using servo control. Notably, the subject of design and compensation or tuning of proportional-integral-derivative control has received adequate attention.

An image of an object is indeed a valuable source of information about that object. In this context, the imaging device is the sensor, and the image is the sensed data. Depending on the imaging device, an image can be many varieties such as optical, thermal or infrared, X-ray, ultraviolet, acoustic, ultrasound, *etc.* The image processing methods are rather similar among these imaging devices. For example, the digital camera is a very popular optical imaging device used in various engineering applications such as vision-guided robotics. Such operations as object recognition, pattern recognition and classification, abstraction, and knowledge-based decision making can be carried out using the information extracted through image processing. Visual servoing[12,13], in particular, has received much attention and is commonly implemented in robotics. Here, the robot motion, including the actual position of the end effector (gripper, hand, tool, *etc.*) and the relative position of the targeted object, is measured using camera images and compared with

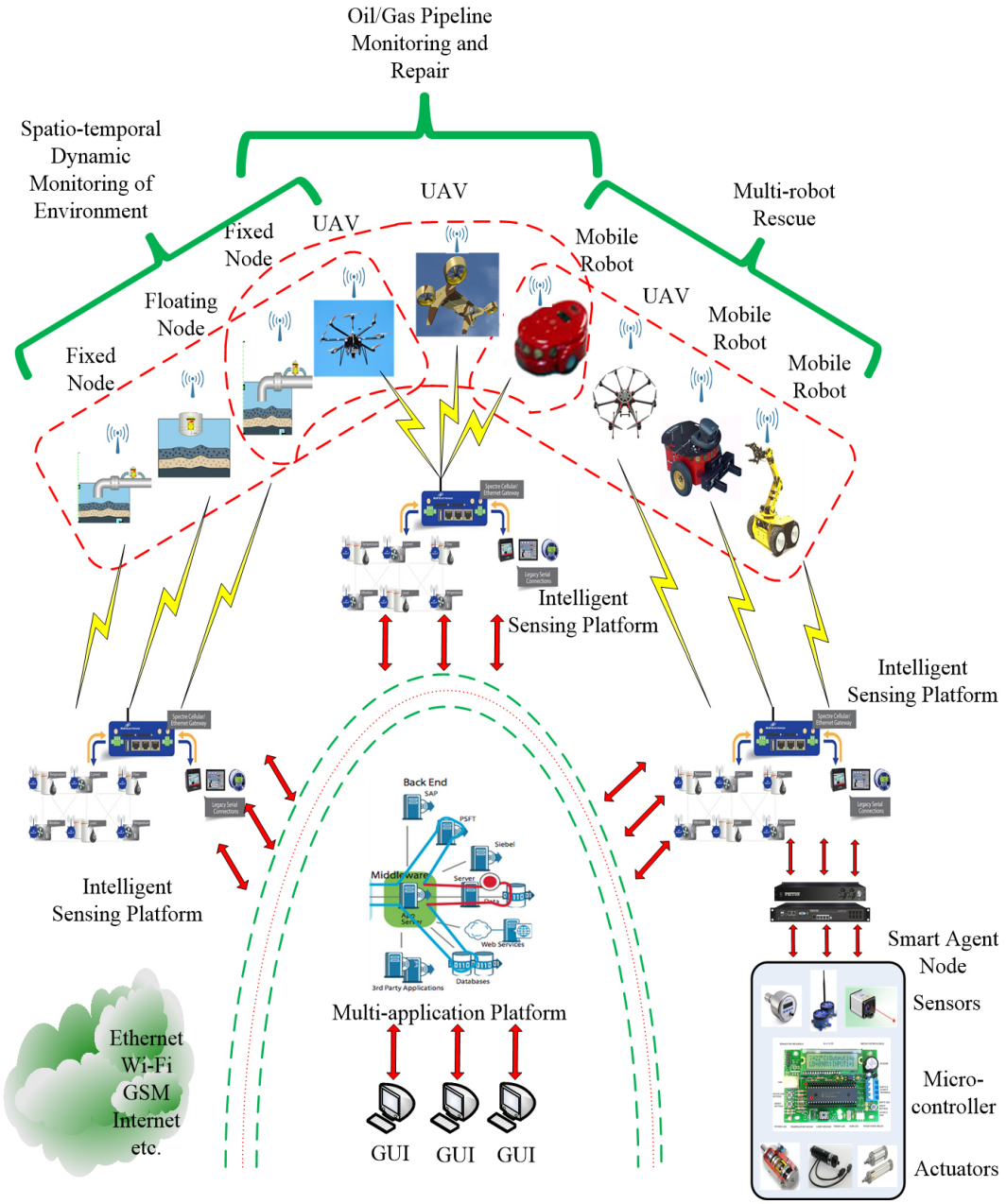


Figure 6. A multi-robot system layout. UAV: Unmanned aerial vehicle; GUI: graphical user interface; GSM: global system for mobile communication.

the desired or reference values. This difference (error) is used to generate a motion command (feedback control) for the robot so that the end effector would follow the desired trajectory and carry out the robotic task.

A robotic system may have inputs that do not participate in feedback control. These inputs are not compared with feedback (measurement) signals to generate control signals. Some of these inputs might be important variables, while others might be undesirable, such as external disturbances and noise. Generally, the performance of a robotic system can be improved by measuring these (unknown) inputs and using the information to generate the control actions. In feedforward control, unknown "inputs" are measured, and

that information, along with desired inputs, is used to generate control signals that can reduce errors due to these unknown inputs or variations in them. The reason for calling this method feedforward control stems from the fact that the associated measurement and control (and compensation) take place in the forward path of the control system. Both feedback and feedforward schemes may be used in the same control system. In some robotic applications, control inputs are computed using the desired outputs and accurate dynamic models for the robots, and the computed inputs are used for control purposes. This is the "inverse model" (or "inverse dynamics") approach because the input is computed using the output and a model (inverse model). In some literature, this method is also known as feedforward control. To avoid confusion, however, it is appropriate to denote this method as computed-input control.

Since the overall response of a plant (*e.g.*, a robot) depends on its individual modes, it should be possible to control a robot by controlling its modes. This is the basis of modal control. A mode is determined by the corresponding eigenvalue and eigenvector. In view of this, a popular approach of modal control is the pole placement or pole assignment. In this method of controller design, the objective is to select a feedback controller that will make the poles of the closed-loop system take up a set of desired values. This approach uses a "linearized" model of the robot.

As we saw, a robot can be controlled using a feedback control law so as to satisfy some performance requirements. In optimal control, the objective is to optimize a suitable objective function (*e.g.*, maximize a performance index or minimize a cost function) by using an appropriate feedback control law[14]. A particularly favorite performance index is the infinite-time *quadratic integral* of the state variables and input variables, and popular control law is linear constant-gain feedback of the system states. The associated controller is known as the linear quadratic regulator (LQR). Linear Quadratic Gaussian (LQG) Control is an optimal control technique that is intended for a linear system with random input disturbances and output (measurement) noise. An LQR controller together with a Kalman filter is used in this approach.

For servo control to be effective, nonlinearities and dynamic coupling of the robot must be compensated faster than the control bandwidth at the servo level. One way of accomplishing this is by implementing a linearizing and decoupling controller inside the servo loops. This technique is termed feedback linearization technique.

An adaptive control system is a feedback control system in which the values of some or all of the controller parameters are modified (adapted) during the system operation (in real-time) on the basis of some performance measure when the response (output) requirements are not satisfied. Many criteria can be employed for modifying the parameter values of a controller. Self-tuning control falls into the same category. Model identification or estimation may be required for adaptive control, which may be considered to be a preliminary step of "learning". A neural network may be used for this purpose. In a learning system, control decisions are made using the cumulative experience and knowledge gained over a period of time. Furthermore, the definition of learning implies that a learning controller will "remember" and improve its performance with time. This is an evolutionary process that is true for intelligent controllers but not generally for adaptive controllers. In model-referenced adaptive control, the same reference input that is applied to the physical system is applied to a reference model as well. The difference between the response of the physical system and the output from the reference model is the error. The ideal objective is to make this error zero at all times. Then the system will perform just like the reference model. The error signal is used by the adaptation mechanism to determine the necessary modifications to the values of the controller parameters in order to achieve this objective.

Sliding mode control[15], variable structure control, and suction control fall within the same class of control techniques and are somewhat synonymous. The control law in this class is generally a switching controller. A variety of switching criteria may be employed. Sliding mode control may be treated as an adaptive control technique. Because the switching surface is not fixed, its variability is somewhat analogous to an adaptation criterion. Specifically, the error of the plant response is zero when the control falls on the sliding surface.

$H\infty$ (H-infinity) control is an optimal control approach, which is different from the LQG method. This frequency-domain technique assumes a linear plant with constant parameters, which may be modeled by a transfer function (matrix in the general case). The underlying design problem is to select a suitable controller that will result in the required performance of the system. In other words, the closed-loop transfer matrix must be properly "shaped" through an appropriate choice of the controller. Specifically, the controller that minimizes the "$H\infty$ norm" of the closed-loop transfer matrix, which is the maximum value of the largest *singular value* of this matrix, is used.

For complex multi-robot systems having various and stringent operating requirements, distributed and networked control is appropriate. It may consist of many programmable logic controllers and a supervisory controller, which will supervise, manage, coordinate and control the overall system. In hierarchical control, the distribution of control is provided both geographically and functionally. The management decisions, supervisory control, and coordination between robots may be provided by the supervisory controller, which is at the highest level of the hierarchy. The next lower level may generate control settings (or reference inputs) for each control region (subsystem). Finally, setpoints and reference signals are inputs to the direct controllers of the robots. In master-slave distributed control, only downloading of information is available.

5.2. Intelligent control

In intelligent control, an "intelligent" method of decision-making is used to make the control decision (*i.e.*, to generate the control action). Soft computing, consisting of neural networks, fuzzy systems, evolutionary computing, and even probabilistic methods, has been popularly used in intelligent control. The topic of soft computing has already been addressed under the general theme of the present paper and is not repeated here. However, it is adequate to mention that, since learning control is used in robotic control, any approach of machine learning such as deep learning and deep neural networks, as discussed earlier in the paper, maybe incorporated into intelligent control of robots.

6. OPPORTUNITIES OF ROBOTICS

The commercial applications of Intelligent Robotics (with AI) include: autonomous agents such as self-driving vehicles (encompassing aerial, ground-based, and underwater vehicles), which are indeed mobile robots; assistive devices (active and adaptive prostheses, wearables, and hand-held smart devices); advisory systems (or, expert systems, which are used in such areas as medical, legal, business, service, and social); monitoring/security systems (they are applicable in such areas as machine fault detection, prediction and diagnosis; and for human health monitoring, in telemedicine, homecare, *etc.*; video analysis; cyber security; human-machine interaction (including natural language processing, facial expression detection, speech recognition, communication, and intelligent connectivity; industrial application (including manufacturing and the assessment of production quality, cost, and efficiency); consumer, service, and entertainment sectors (retail, domestic, social, *etc.*); agriculture (growing, fertilizing, weed removal, and harvesting); smart buildings (heating ventilation, and air conditioning - HVAC; smart metering, safety, smart appliances, automated lighting, and achieving energy efficiency); education ("intelligent" learning management system or LMS, collaboration among students and with teachers - this approach may be quite beneficial in the current epidemic situation of Covid-19); and energy and environment (distribution, exploration,

monitoring, planning, and utilization of energy). Some of these applications have been implemented today. However, some will provide diverse future opportunities.

6.1. Opportunities for developing countries

Here, opportunities exist in all the areas that were mentioned before. However, developing countries should not blindly decide on the considered robotics activities just for the sake of being involved in Robotics or AI. It is important to explore and determine what is in the "black box". Otherwise, one can be dissuaded through fear-mongering or make wrong choices for robotic activities. One must first question whether Robotics is needed for a specific local application. Then they must explore which robotic approaches are relevant for the considered task. Very importantly, they must examine what is in the existing Black Box before implementing it.

Developing countries should primarily concentrate on "robot development", not their application for the automation of local industries. They will be able to market these robots to other countries. Since the developing countries typically have an excessive and smart labor force, using robots for such applications as agriculture and industrial automation is not generally suitable in those countries. Nevertheless, they may consider the development of simple and low-cost robots for local use (*e.g.*, for service and household applications). They may focus on the development of advanced software, in particular, to incorporate other forms of intelligence into robots and efficient software, and the use of advanced platforms like Flexible Cloud, Real-time Internet of Things, and Edge AI. Software development can be carried out without much capital investment, as it is done in India particularly because these countries normally have an educated and vast group of professionals. Also, they should focus on advancing the "mechanical capabilities" of robots, which are essential but may not necessarily be for the local market. As well, they should consider the needs that result from a particular situation (*e.g.*, Covid-19). Very importantly, they should develop their own guidelines and regulations for robotic ethics and safety, which can be done by modifying the existing guidelines and regulations in the highly developed jurisdictions.

7. CONCLUSIONS

Robotics has found numerous practical applications today in industry, medicine, the service sector, household, and the general society. Important developments and practical strides are being made, particularly in Soft Robotics, Mobile Robotics (Aerial - drones, Underwater, Ground-based - autonomous vehicles in particular), Swarm Robotics, Homecare, Surgery, Assistive Devices, and Active Prosthesis. This perspective paper presented a brief history of Robotics while indicating some associated myths and unfair expectations. Next, it presented some important practical applications of Robotics, as developed by groups worldwide, including the Industrial Automation Laboratory at the University of British Columbia, headed by the author. The main shortcomings of Intelligent Robotics included those of the mechanical capabilities and the nature of the available level of intelligence. Concerning robotic intelligence, apart from the current focus of "learning", other characteristics should be further explored and incorporated. They included sensory perception, pattern recognition, decision making from incomplete information, inference from qualitative or approximate information (qualitative reasoning), ability to deal with unfamiliar situations, adaptability to new, yet related situations (through "expectational knowledge"), inductive reasoning, common sense, display of emotions, inventiveness, and self-awareness. Finally, the future trends and key opportunities available in Intelligent Robotics for both developed and developing counties were indicated.

DECLARATIONS

Authors' contributions

The author contributed solely to the article.

Availability of data and materials
Not applicable.

Financial support and sponsorship
This work has been supported through research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Conflicts of interest
The author declared that there are no conflicts of interest.

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

REFERENCES
1. Unimate, the first industrial robot. Available from: https://youtu.be/hxsWeVtb-JQ?t=22. [Last accessed on 9 Aug 2021].
2. Welding robots in an automotive plant. Available from: https://youtu.be/0L7Xk5_s3QQ. [Last accessed on 9 Aug 2021].
3. Humanoid robot, Honda Asimo. Available from: https://asimo.honda.com/. [Last accessed on 9 Aug 2021].
4. Chen J, Shu T, Li T, de Silva CW. Deep reinforced learning tree for spatiotemporal monitoring with mobile robotic wireless sensor networks. *IEEE Trans Syst Man Cybern, Syst* 2020;50:4197-211. DOI
5. Li T, Wang C, Meng MQH, de Silva CW. Coverage sampling planner for UAV-enabled environmental exploration and field mapping. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2019 Nov 3-8; Macau, China. 2019. DOI
6. Zhang K, Luo J, Xiao W, et al. A subvision system for enhancing the environmental adaptability of the powered transfemoral prosthesis. *IEEE Trans Cybern* 2021;51:3285-97. DOI PubMed
7. de Silva CW. Video interview on the award of Killam Research Prize, The University of British Columbia, Vancouver, Canada. Available from: https://youtu.be/rMW32o2QD5E. [Last accessed on 9 Aug 2021].
8. Time delay in teleoperation. Available from: https://youtu.be/sl5ckyY7Zao?t=9. [Last accessed on 9 Aug 2021].
9. Karray FO, de Silva CW. Soft computing and intelligent systems design: theory, tools, and applications. Addison Wesley, New York, NY. 2004.
10. Xia M, Li T, Xu L, Liu L, de Silva CW. Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks. *IEEE/ASME Trans Mechatron* 2018;23:101-10. DOI
11. Wang J, Chen J, Lin J, Sigal L, de Silva CW. Discriminative feature alignment: improving transferability of unsupervised domain adaptation by Gaussian-guided latent alignment. *Pattern Recogn* 2021;116:107943. DOI
12. Zhang Y, Wang Y, Lang H, Wang Y, de Silva CW. Visual avoidance of collision with randomly moving obstacles through approximate reinforcement learning. *Instrumentation* 2019;6:59-66. DOI
13. Wang Y, Zhang G, Lang H, Zuo B, de Silva CW. A modified image-based visual servo controller with hybrid camera configuration for robust robotic grasping. *Robot Auton Syst* 2014;62:1398-407. DOI
14. Ma J, Cheng Z, Wang W, et al. Convex inner approximation for mixed H2/H-infinity control with application to a 2-DoF flexure-based nano-positioning system. *IEEE Trans Ind Electron* 2021. DOI
15. Wang W, Ma J, Cheng Z, Li X, de Silva CW, Lee TH. Global iterative sliding mode control of an industrial biaxial gantry system for contouring motion. Available from: https://arxiv.org/abs/2103.12580 [Last accessed on 9 Aug 2021].

Qi *et al. Intell Robot* 2021;1(1):18-57
DOI: 10.20517/ir.2021.02

Intelligence & Robotics

Research Article

Open Access

Federated reinforcement learning: techniques, applications, and open challenges

Jiaju Qi[1], Qihao Zhou[2], Lei Lei[1], Kan Zheng[2]

[1]School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada.
[2]Intelligent Computing and Communications (IC²) Lab, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Correspondence to: Dr. Lei Lei, School of Engineering, University of Guelph, 50 Stone Road East, Guelph, ON N1G 2W1, Canada. E-mail: leil@uoguelph.ca

How to cite this article: Qi J, Zhou Q, Lei L, Zheng K. Federated reinforcement learning: techniques, applications, and open challenges. *Intell Robot* 2021;1(1):18-57. http://dx.doi.org/10.20517/ir.2021.02

Received: 24 Aug 2021 **First Decision:** 14 Sep 2021 **Revised:** 21 Sep 2021 **Accepted:** 22 Sep 2021 **Published:** 12 Oct 2021

Academic Editor: Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

This paper presents a comprehensive survey of federated reinforcement learning (FRL), an emerging and promising field in reinforcement learning (RL). Starting with a tutorial of federated learning (FL) and RL, we then focus on the introduction of FRL as a new method with great potential by leveraging the basic idea of FL to improve the performance of RL while preserving data-privacy. According to the distribution characteristics of the agents in the framework, FRL algorithms can be divided into two categories, *i.e.*, horizontal federated reinforcement learning and vertical federated reinforcement learning (VFRL). We provide the detailed definitions of each category by formulas, investigate the evolution of FRL from a technical perspective, and highlight its advantages over previous RL algorithms. In addition, the existing works on FRL are summarized by application fields, including edge computing, communication, control optimization, and attack detection. Finally, we describe and discuss several key research directions that are crucial to solving the open problems within FRL.

Keywords: Federated learning, reinforcement learning, federated reinforcement learning

1. INTRODUCTION

As machine learning (ML) develops, it becomes capable of solving increasingly complex problems, such as image recognition, speech recognition, and semantic understanding. Despite the effectiveness of traditional machine learning algorithms in several areas, the researchers found that scenes involving many parties are still

difficult to resolve, especially when privacy is concerned. Federated learning (FL), in these cases, has attracted increasing interest among ML researchers. Technically, the FL is a decentralized collaborative approach that allows multiple partners to train data respectively and build a shared model while maintaining privacy. With its innovative learning architecture and concepts, FL provides safer experience exchange services and enhances capabilities of ML in distributed scenarios.

In ML, reinforcement learning (RL) is one of the branches that focuses on how individuals, *i.e*., agents, interact with their environment and maximize some portion of the cumulative reward. The process allows agents to learn to improve their behavior in a trial and error manner. Through a set of policies, they take actions to explore the environment and expect to be rewarded. Research on RL has been hot in recent years, and it has shown great potential in various applications, including games, robotics, communication, and so on.

However, there are still many problems in the implementation of RL in practical scenarios. For example, considering that in the case of large action space and state space, the performance of agents is vulnerable to collected samples since it is nearly impossible to explore all sampling spaces. In addition, many RL algorithms have the problem of learning efficiency caused by low sample efficiency. Therefore, through information exchange between agents, learning speed can be greatly accelerated. Although distributed RL and parallel RL algorithms[1–3] can be used to address the above problems, they usually need to collect all the data, parameters, or gradients from each agent in a central server for model training. However, one of the important issues is that some tasks need to prevent agent information leakage and protect agent privacy during the application of RL. Agents' distrust of the central server and the risk of eavesdropping on the transmission of raw data has become a major bottleneck for such RL applications. FL can not only complete information exchange while avoiding privacy disclosure, but also adapt various agents to their different environments. Another problem of RL is how to bridge the simulation-reality gap. Many RL algorithms require pre-training in simulated environments as a prerequisite for application deployment, but one problem is that the simulated environments cannot accurately reflect the environments of the real world. FL can aggregate information from both environments and thus bridge the gap between them. Finally, in some cases, only partial features can be observed by each agent in RL. However, these features, no matter observations or rewards, are not enough to obtain sufficient information required to make decisions. At this time, FL makes it possible to integrate this information through aggregation.

Thus, the above challenges give rise to the idea of federated reinforcement learning (FRL). As FRL can be considered as an integration of FL and RL under privacy protection, several elements of RL can be presented in FL frameworks to deals with sequential decision-making tasks. For example, these three dimensions of sample, feature and label in FL can be replaced by environment, state and action respectively in FRL. Since FL can be divided into several categories according to the distribution characteristics of data, including horizontal federated learning (HFL) and vertical federated learning (VFL), we can similarly categorize FRL algorithms into horizontal federated reinforcement learning (HFRL) and vertical federated reinforcement learning (VFRL).

Though a few survey papers on FL[4–6] have been published, to the best of our knowledge, there are currently no relevant survey papers focused on FRL. Due to the fact that FRL is a relatively new technique, most researchers may be unfamiliar with it to some extent. We hope to identify achievements from current studies and serve as a stepping stone to further research. In summary, this paper sheds light on the following aspects.

1. *Systematic tutorial on FRL methodology*. As a review focusing on FRL, this paper tries to explain the knowledge about FRL to researchers systematically and in detail. The definition and categories of FRL are introduced firstly, including system model, algorithm process, *etc*. In order to explain the framework of HFRL and VFRL and the difference between them clearly, two specific cases are introduced, *i.e*., autonomous driving and smart grid. Moreover, we comprehensively introduce the existing research on FRL's algorithm

design.

2. *Comprehensive summary for FRL applications*. This paper collects a large number of references in the field of FRL, and provides a comprehensive and detailed investigation of the FRL applications in various areas, including edge computing, communications, control optimization, attack detection, and some other applications. For each reference, we discuss the authors' research ideas and methods, and summarize how the researchers combine the FRL algorithm with the specific practical problems.
3. *Open issues for future research*. This paper identifies several open issues for FRL as a guide for further research. The scope covers communication, privacy and security, join and exit mechanisms design, learning convergence and some other issues. We hope that they can broaden the thinking of interested researchers and provide help for further research on FRL.

The organization of this paper is as follows. To quickly gain a comprehensive understanding of FRL, the paper starts with FL and RL in Section 2 and Section 3, respectively, and extends the discussion further to FRL in Section 4. The existing applications of FRL are summarized in Section 5. In addition, a few open issues and future research directions for FRL are highlighted in Section 6. Finally, the conclusion is given in Section 7.

2. FEDERATED LEARNING

2.1. Federated learning definition and basics

In general, FL is a ML algorithmic framework that allows multiple parties to perform ML under the requirements of privacy protection, data security, and regulations[7]. In FL architecture, model construction includes two processes: model training and model inference. It is possible to exchange information about the model between parties during training, but not the data itself, so that data privacy will not be compromised in any way. An individual party or multiple parties can possess and maintain the trained model. In the process of model aggregation, more data instances collected from various parties contribute to updating the model. As the last step, a fair value-distribution mechanism should be used to share the profits obtained by the collaborative model[8]. The well-designed mechanism enables the federation sustainability. Aiming to build a joint ML model without sharing local data, FL involves technologies from different research fields such as distributed systems, information communication, ML and cryptography[9]. FL has the following characteristics as a result of these techniques, *i.e*.,

- Distribution. There are two or more parties that hope to jointly build a model to tackle similar tasks. Each party holds independent data and would like to use it for model training.
- Data protection. The data held by each party does not need to be sent to the other during the training of the model. The learned profits or experiences are conveyed through model parameters that do not involve privacy.
- Secure communication. The model is able to be transmitted between parties with the support of an encryption scheme. The original data cannot be inferred even if it is eavesdropped during transmission.
- Generality. It is possible to apply FL to different data structures and institutions without regard to domains or algorithms.
- Guaranteed performance. The performance of the resulting model is very close to that of the ideal model established with all data transferred to one centralized party.
- Status equality. To ensure the fairness of cooperation, all participating parties are on an equal footing. The shared model can be used by each party to improve its local models when needed.

A formal definition of FL is presented as follows. Consider that there are N parties $\{\mathcal{F}_i\}_{i=1}^N$ interested in establishing and training a cooperative ML model. Each party has their respective datasets $\mathcal{D}_i$. Traditional ML approaches consist of collecting all data $\{\mathcal{D}_i\}_{i=1}^N$ together to form a centralized dataset $\mathbb{R}$ at one data server. The expected model $\mathcal{M}_{SUM}$ is trained by using the dataset $\mathbb{R}$. On the other hand, FL is a reform of ML process in which the participants $\mathcal{F}_i$ with data $\mathcal{D}_i$ jointly train a target model $\mathcal{M}_{FED}$ without aggregating their data. Respective data $\mathcal{D}_i$ is stored on the owner $\mathcal{F}_i$ and not exposed to others. In addition, the performance mea-

sure of the federated model $\mathcal{M}_{FED}$ is denoted as $\mathcal{V}_{FED}$, including accuracy, recall, and F1-score, *etc*, which should be a good approximation of the performance of the expected model $\mathcal{M}_{SUM}$, *i.e.*, $\mathcal{V}_{SUM}$. In order to quantify differences in performance, let δ be a non-negative real number and define the federated learning model $\mathcal{M}_{FED}$ has δ performance loss if

$$|\mathcal{V}_{SUM} - \mathcal{V}_{FED}| < \delta.$$

Specifically, the FL model hold by each party is basically the same as the ML model, and it also includes a set of parameters w_i which is learned based on the respective training dataset $\mathcal{D}_i$ [10]. A training sample j typically contains both the input of FL model and the expected output. For example, in the case of image recognition, the input is the pixel of the image, and the expected output is the correct label. The learning process is facilitated by defining a loss function on parameter vector w for every data sample j. The loss function represents the error of the model in relation to the training data. For each dataset $\mathcal{D}_i$ at party $\mathcal{F}_i$, the loss function on the collection of training samples can be defined as follow [11],

$$F_i(w) = \frac{1}{|\mathcal{D}_i|} \sum_{j \in \mathcal{D}_i} f_j(w),$$

where $f_j(w)$ denotes the loss function of the sample j with the given model parameter vector w and $|\cdot|$ represents the size of the set. In FL, it is important to define the global loss function since multiple parties are jointly training a global statistical model without sharing a dataset. The common global loss function on all the distributed datasets is given by,

$$F_g(w) = \sum_{i=1}^{N} \eta_i F_i(w),$$

where η_i indicates the relative impact of each party on the global model. In addition, $\eta_i > 0$ and $\sum_{i=1}^{N} \eta_i = 1$. This term η can be flexibly defined to improve training efficiency. The natural setting is averaging between parties, *i.e.*, $\eta = 1/N$. The goal of the learning problem is to find the optimal parameter that minimizes the global loss function $F_g(w)$. It is presented in formula form,

$$w^* = \arg\min_{w} F_g(w).$$

Considering that FL is designed to adapt to various scenarios, the objective function may be appropriate depending on the application. However, a closed-form solution is almost impossible to find with most FL models due to their inherent complexity. A canonical federated averaging algorithm (FedAvg) based on gradient-descent techniques is presented in the study from McMahan *et al.* [12], which is widely used in FL systems. In general, the coordinator has the initial FL model and is responsible for aggregation. Distributed participants know the optimizer settings and can upload information that does not affect privacy. The specific architecture of FL will be discussed in the next subsection. Each participant uses their local data to perform one step (or multiple steps) of gradient descent on the current model parameter $\bar{w}(t)$ according to the following formula,

$$\forall i, w_i(t+1) = \bar{w}(t) - \gamma \nabla F_i(\bar{w}_i(t)),$$

where γ denotes a fixed learning rate of each gradient descent. After receiving the local parameters from participants, the central coordinator updates the global model using a weighted average, *i.e.*,

$$\bar{w}_g(t+1) = \sum_{i=1}^{N} \frac{n_i}{n} w_i(t+1),$$

where n_i indicates the number of training data samples of the i-th participant has and n denotes the total number of samples contained in all the datasets. Finally, the coordinator sends the aggregated model weights $\bar{w}_g\ (t+1)$ back to the participants. The aggregation process is performed at a predetermined interval or iteration round. Additionally, FL leverages privacy-preserving techniques to prevent the leakage of gradients or model weights. For example, the existing encryption algorithms are added on top of the original FedAvg to provide secure FL[13,14].

2.2. Architecture of federated learning

According to the application characteristics, the architecture of FL can be divided into two types[7], *i.e.*, client-server model and peer-to-peer model.

As shown in Figure 1, there are two major components in the client-server model, *i.e.*, participants and coordinators. The participants are the data owners and can perform local model training and updates. In different scenarios, the participants are made up of different devices, the vehicles in the internet of vehicles (IoV), or the smart devices in the IoT. In addition, participants usually possess at least two characteristics. Firstly, each participant has a certain level of hardware performance, including computation power, communication and storage. The hardware capabilities ensure that the FL algorithm operates normally. Secondly, participants are independent of one another and located in a wide geographic area. In the client-server model, coordinator can be considered as a central aggregation server, which can initialize a model and aggregate model updates from participants[12]. As participants train both based on local data sets concurrently and share their experience through the coordinator with the model aggregation mechanism, it will greatly enhance the efficiency of the training and enhance the performance of the model. However, since participants won't be able to communicate directly, the coordinator must perform well to train the global model and maintain communication with all participants. Therefore, the model has security challenges such as a single point of failure. If the coordinator fails to complete the model aggregation task, the local model of participant has difficulty meeting target performance. The basic workflow of the client-server model can be summarized in the following five steps. The process continues to repeat the steps from 2 to 5 until the model converges, or until the maximum number of iterations is reached.

- Step 1: In the process of setting up a client-server-based learning system, the coordinator creates an initial model and sends it to each participant. Those participants who join later can access the latest global model.
- Step 2: Each participant trains a local model based on their respective dataset.
- Step 3: Updates of model parameters are sent to the central coordinator.
- Step 4: The coordinator combines the model updates using specific aggregation algorithms.
- Step 5: The combined model is sent back to the corresponding participant.

The peer-to-peer based FL architecture does not require a coordinator as illustrated in Figure 2. Participants can directly communicate with each other without relying on a third party. Therefore, each participant in the architecture is equal and can initiate a model exchange request with anyone else. As there is no central server, participants must agree in advance on the order in which model should be sent and received. Common transfer modes are cyclic transfer and random transfer. The peer-to-peer model is suitable and important for specific scenarios. For example, multiple banks jointly develop an ML-based attack detection model. With FL, there is no need to establish a central authority between banks to manage and store all attack patterns. The attack record is only held at the server of the attacked bank, but the detection experience can be shared with other participants through model parameters. The FL procedure of the peer-to-peer model is simpler than that of the client-server model.

- Step 1: Each participant initializes their local model depending on its needs.
- Step 2: Train the local model based on the respective dataset.
- Step 3: Create a model exchange request to other participants and send local model parameters.
- Step 4: Aggregate the model received from other participants into the local model.

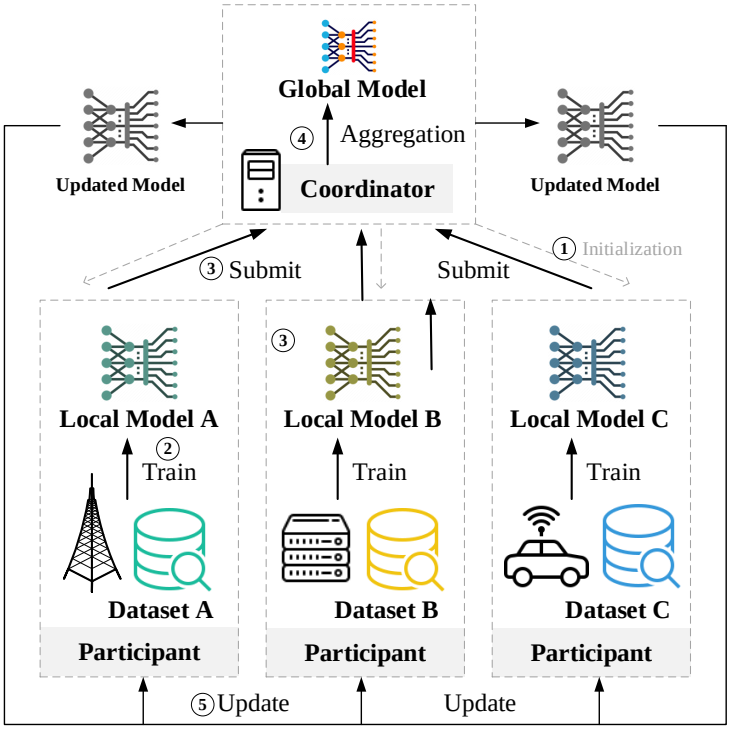


Figure 1. An example of federated learning architecture: Client-Server Model.

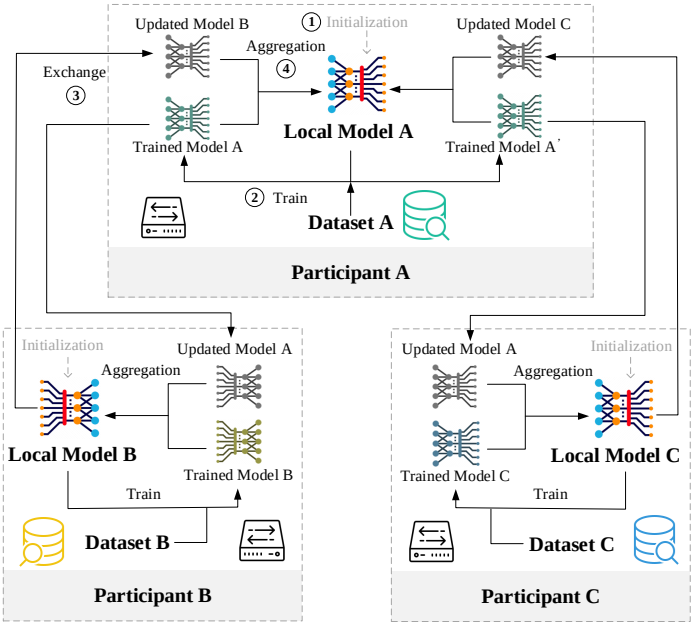


Figure 2. An example of federated learning architecture: Peer-to-Peer Model.

The termination conditions of the process can be designed by participants according to their needs. This architecture further guarantees security since there is no centralized server. However, it requires more communication resources and potentially increased computation for more messages.

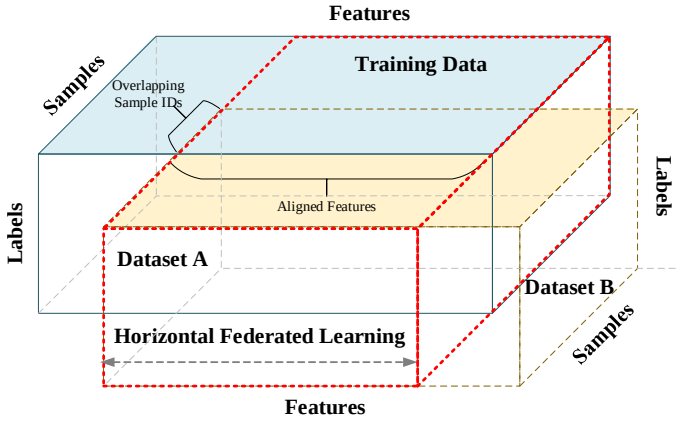


Figure 3. Illustration of horizontal federated learning.

2.3. Categories of federated learning

Based on the way data is partitioned within a feature and sample space, FL may be classified as HFL, VFL, or federated transfer learning (FTL)[8]. In Figure 3, Figure 4, and Figure 5, these three federated learning categories for a two-party scenario are illustrated. In order to define each category more clearly, some parameters are formalized. We suppose that the i-th participant has its own dataset $\mathcal{D}_i$. The dataset includes three types of data, *i.e.*, the feature space $\mathcal{X}_i$, the label space $\mathcal{Y}_i$ and the sample ID space $\mathcal{I}_i$. In particular, the feature space $\mathcal{X}_i$ is a high-dimensional abstraction of the variables within each pattern sample. Various features are used to characterize data held by the participant. All categories of association between input and task target are collected in the label space $\mathcal{Y}_i$. The sample ID space $\mathcal{I}_i$ is added in consideration of actual application requirements. The identification can facilitate the discovery of possible connections among different features of the same individual.

HFL indicates the case in which participants have their dataset with a small sample overlap, while most of the data features are aligned. The word "horizontal" is derived from the term "horizontal partition". This is similar to the situation where data is horizontally partitioned inside the traditional tabular view of a database. As shown in Figure 3, the training data of two participants with the aligned features is horizontally partitioned for HFL. A cuboid with a red border represents the training data required in learning. Especially, a row corresponds to complete data features collected from a sampling ID. Columns correspond to different sampling IDs. The overlapping part means there can be more than one participant sampling the same ID. In addition, HFL is also known as feature-aligned FL, sample-partitioned FL, or example-partitioned FL. Formally, the conditions for HFL can be summarized as

$$\mathcal{X}_i = \mathcal{X}_j, \mathcal{Y}_i = \mathcal{Y}_j, \mathcal{I}_i \neq \mathcal{I}_j, \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j,$$

where $\mathcal{D}_i$ and $\mathcal{D}_j$ denote the datasets of participant i and participant j respectively. In both datasets, the feature space $\mathcal{X}$ and label space $\mathcal{Y}$ are assumed to be the same, but the sampling ID space $\mathcal{I}$ is assumed to be different. The objective of HFL is to increase the amount of data with similar features, while keeping the original data from being transmitted, thus improving the performance of the training model. Participants can still perform feature extraction and classification if new samples appear. HFL can be applied in various fields because it benefits from privacy protection and experience sharing[15]. For example, regional hospitals may receive different patients, and the clinical manifestations of patients with the same disease are similar. It is imperative to protect the patient's privacy, so data about patients cannot be shared. HFL provides a good way to jointly build a ML model for identifying diseases between hospitals.

VFL refers to the case where different participants with various targets usually have datasets that have different

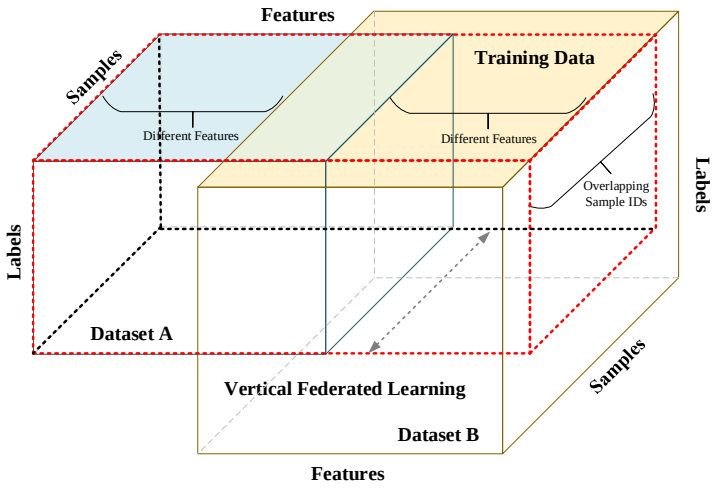


Figure 4. Illustration of vertical federated learning.

feature spaces, but those participants may serve a large number of common users. The heterogeneous feature spaces of distributed datasets can be used to build more general and accurate models without releasing the private data. The word "vertical" derives from the term "vertical partition", which is also widely used in reference to the traditional tabular view. Different from HFL, the training data of each participant are divided vertically. Figure 4 shows an example of VFL in a two-party scenario. The important step in VFL is to align samples, *i.e.*, determine which samples are common to the participants. Although the features of the data are different, the sampled identity can be verified with the same ID. Therefore, VFL is also called sample-aligned FL or feature-partitioned FL. Multiple features are vertically divided into one or more columns. The common samples exposed to different participants can be marked by different labels. The formal definition of VFL's applicable scenario is given.

$$\mathcal{X}_i \neq \mathcal{X}_j, \mathcal{Y}_i \neq \mathcal{Y}_j, \mathcal{I}_i = \mathcal{I}_j, \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j,$$

where $\mathcal{D}_i$ and $\mathcal{D}_i$ represent the dataset held by different participants, and the data feature space pair $(\mathcal{X}_i, \mathcal{X}_j)$ and label space pair $(\mathcal{Y}_i, \mathcal{Y}_j)$ are assumed to be different. The sample ID space $\mathcal{I}_i$ and $\mathcal{I}_j$ are assumed to be the same. It is the objective of VFL to collaborate in building a shared ML model by exploiting all features collected by each participant. The fusion and analysis of existing features can even infer new features. An example of the application of VFL is the evaluation of trust. Banks and e-commerce companies can create a ML model for trust evaluation for users. The credit card record held at the bank and the purchasing history held at the e-commerce company for the set of same users can be used as training data to improve the evaluation model.

FTL applies to a more general case where the datasets of participants are not aligned with each other in terms of samples or features. FTL involves finding the invariant between a resource-rich source domain and a resource-scarce target domain, and exploiting that invariant to transfer knowledge. In comparison with traditional transfer learning[16], FTL focuses on privacy-preserving issues and addresses distributed challenges. An example of FTL is shown in Figure 5. The training data required by FTL may include all data owned by multiply parties for comprehensive information extraction. In order to predict labels for unlabeled new samples, a prediction model is built using additional feature representations for mixed samples from participants A and B. More formally, FTL is applicable for the following scenarios:

$$\mathcal{X}_i \neq \mathcal{X}_j, \mathcal{Y}_i \neq \mathcal{Y}_j, \mathcal{I}_i \neq \mathcal{I}_j, \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j,$$

In datasets $\mathcal{D}_i$ and $\mathcal{D}_j$, there is no duplication or similarity in terms of features, labels and samples. The objective of FTL is to generate as accurate a label prediction as possible for newly incoming samples or unlabeled samples already present. Another benefit of FTL is that it is capable of overcoming the absence of data or labels.

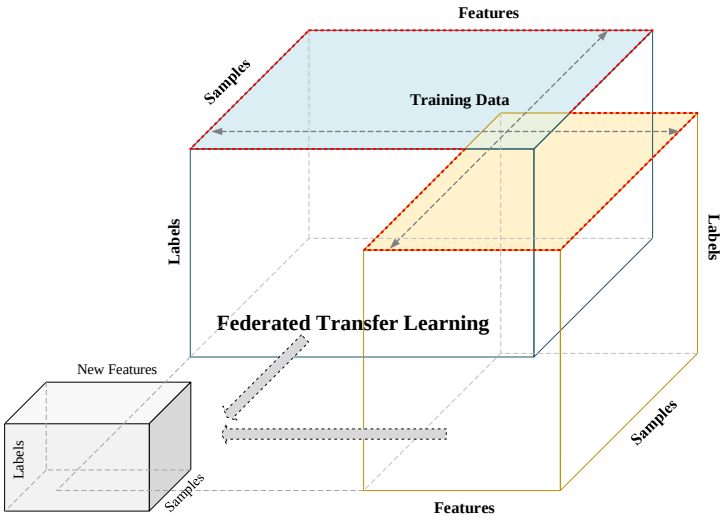


Figure 5. Illustration of federated transfer learning.

For example, a bank and an e-commerce company in two different countries want to build a shared ML model for user risk assessment. In light of geographical restrictions, the user groups of these two organizations have limited overlap. Due to the fact that businesses are different, only a small number of data features are the same. It is important in this case to introduce FTL to solve the problem of small unilateral data and fewer sample labels, and improve the model performance.

3. REINFORCEMENT LEARNING

3.1. Reinforcement learning definition and basics

Generally, the field of ML includes supervised learning, unsupervised learning, RL, *etc*[17]. While supervised and unsupervised learning attempt to make the agent copy the data set, *i.e.*, learning from the pre-provided samples, RL is to make the agent gradually stronger in the interaction with the environment, *i.e.*, generating samples to learn by itself[18]. RL is a very hot research direction in the field of ML in recent years, which has made great progress in many applications, such as IoT[19–22], autonomous driving[23,24], and game design[25]. For example, the AlphaGo program developed by DeepMind is a good example to reflect the thinking of RL[26]. The agent gradually accumulates the intelligent judgment on the sub-environment of each move by playing game by game with different opponents, so as to continuously improve its level.

The RL problem can be defined as a model of the agent-environment interaction, which is represented in Figure 6. The basic model of RL contains several important concepts, *i.e.*,

- **Environment** and **agent**: Agents are a part of a RL model that exists in an external environment, such as the player in the environment of chess. Agents can improve their behavior by interacting with the environment. Specifically, they take a series of actions to the environment through a set of policies and expect to get a high payoff or achieve a certain goal.
- **Time step**: The whole process of RL can be discretized into different time steps. At every time step, the environment and the agent interact accordingly.
- **State**: The state reflects agents' observations of the environment. When agents take action, the state will also change. In other words, the environment will move to the next state.
- **Actions**: Agents can assess the environment, make decisions and finally take certain actions. These actions are imposed on the environment.
- **Reward**: After receiving the action of the agent, the environment will give the agent the state of the current

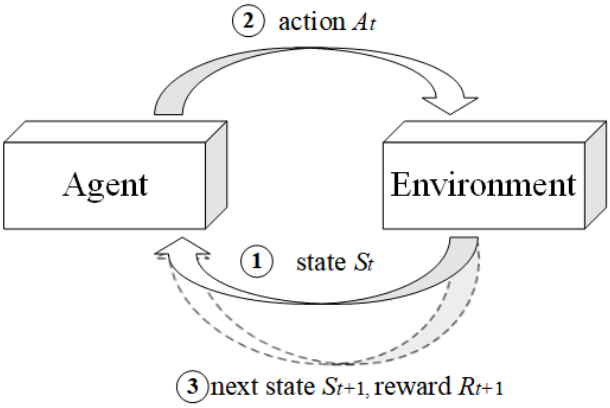


Figure 6. The agent-environment interaction of the basic reinforcement learning model.

environment and the reward due to the previous action. Reward represents an assessment of the action taken by agents.

More formally, we assume that there are a series of time steps $t = 0,1,2,\ldots$ in a basic RL model. At a certain time step t, the agent will receive a state signal S_t of the environment. In each step, the agent will select one of the actions allowed by the state to take an action A_t. After the environment receives the action signal A_t, the environment will feed back to the agent the corresponding status signal S_{t+1} at the next step $t + 1$ and the immediate reward R_{t+1}. The set of all possible states, *i.e.*, the state space, is denoted as $\mathcal{S}$. Similarly, the action space is denoted as $\mathcal{A}$. Since our goal is to maximize the total reward, we can quantify this total reward, usually referred to as return with

$$G_t = R_{t+1} + R_{t+2} + \ldots + R_T,$$

where T is the last step, *i.e.*, S_T as the termination state. An **episode** is completed when the agent completes the termination action.

In addition to this type of episodic task, there is another type of task that does not have a termination state, in other words, it can in principle run forever. This type of task is called a continuing task. For continuous tasks, since there is no termination state, the above definition of return may be divergent. Thus, another way to calculate return is introduced, which is called discounted return, *i.e.*,

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

where the discount factor γ satisfies $0 \leqslant \gamma \leqslant 1$. When $\gamma = 1$, the agent can obtain the full value of all future steps, while when $\gamma = 0$, the agent can only see the current reward. As γ changes from 0 to 1, the agent will gradually become forward-looking, looking not only at current interests, but also for its own future.

The value function is the agent's prediction of future rewards, which is used to evaluate the quality of the state and select actions. The difference between the value function and rewards is that the latter is defined as evaluating an immediate sense for interaction while the former is defined as the average return of actions over a long period of time. In other words, the value function of the current state $S_t = s$ is its long-term expected return. There are two significant value functions in the field of RL, *i.e.*, state value function $V_\pi(s)$ and action value function $Q_\pi(s, a)$. The function $V_\pi(s)$ represents the expected return obtained if the agent continues to follow strategy π all the time after reaching a certain state S_t, while the function $Q_\pi(s, a)$ represents the expected return obtained if action $A_t = a$ is taken after reaching the current state $S_t = s$ and the following actions are taken according to the strategy π. The two functions are specifically defined as follows, *i.e.*,

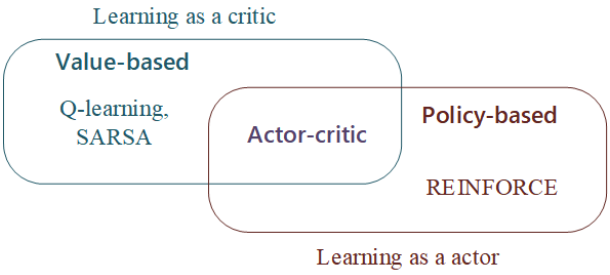


Figure 7. The categories and representative algorithms of reinforcement learning.

$$V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s], \forall s \in \mathcal{S}$$
$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a], \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

The results of RL are action decisions, called as the policy. The policy gives agents the action a that should be taken for each state s. It is noted as $=\pi(A_t = a|S_t = s)$, which represents the probability of taking action $A_t = a$ in state $S_t = s$. The goal of RL is to learn the optimal policy that can maximize the value function by interacting with the environment. Our purpose is not to get the maximum reward after a single action in the short term, but to get more reward in the long term. Therefore, the policy can be figured out as,

$$\pi^* = arg\max_\pi V_\pi(s), \forall s \in \mathcal{S}.$$

3.2. Categories of reinforcement learning

In RL, there are several categories of algorithms. One is value-based and the other is policy-based. In addition, there is also an actor-critic algorithm that can be obtained by combining the two, as shown in Figure 7.

3.2.1. Value-based methods

Recursively expand the formulas of the action value function, the corresponding Bellman equation is obtained, which describes the recursive relationship between the value function of the current state and subsequent state. The recursive expansion formula of the action value function $Q_\pi(s,a)$ is

$$Q_\pi(s,a) = \sum_{s',r} p\left(s', r|s, a\right)\left[r + \gamma \sum_{a'} \pi\left(a'|s'\right) Q_\pi\left(s', a'\right)\right],$$

where the function $p\left(s', r|s, a\right) = Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}$ defines the trajectory probability to characterize the environment's dynamics. $R_t = r$ indicates the reward obtained by the agent taking action $A_{t-1} = a$ in state $S_{t-1} = s$. Besides, $S_t = s'$ and $A_t = a'$ respectively represent the state and the action taken by the agent at the next moment t.

In the value-based algorithms, the above value function $Q_\pi(s,a)$ is calculated iteratively, and the strategy is then improved based on this value function. If the value of every action in a given state is known, the agent can select an action to perform. In this way, if the optimal $Q_\pi(s, a = a^*)$ can be figured out, the best action a^* will be found. There are many classical value-based algorithms, including Q-learning[27], state–action–reward–state–action (SARSA)[28], *etc.*

Q-learning is a typical widely-used value-based RL algorithm. It is also a model-free algorithm, which means that it does not need to know the model of the environment but directly estimates the Q value of each executed

action in each encountered state through interacting with the environment[27]. Then, the optimal strategy is formulated by selecting the action with the highest Q value in each state. This strategy maximizes the expected return for all subsequent actions from the current state. The most important part of Q-learning is the update of Q value. It uses a table, *i.e.*, Q-table, to store all Q value functions. Q-table uses state as row and action as column. Each (s, a) pair corresponds to a Q value, *i.e.*, $Q(s, a)$, in the Q-table, which is updated as follows,

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q\left(s', a'\right) - Q(s,a) \right]$$

where r is the reward given by taking action a under state s at the current time step. s' and a' indicate the state and the action taken by the agent at the next time step respectively. α is the learning rate to determine how much error needs to be learned, and γ is the attenuation of future reward. If the agent continuously accesses all state-action pairs, the Q-learning algorithm will converge to the optimal Q function. Q-learning is suitable for simple problems, *i.e.*, small state space, or a small number of actions. It has high data utilization and stable convergence.

3.2.2. Policy-based methods

The above value-based method is an indirect approach to policy selection, and has trouble handling an infinite number of actions. Therefore, we want to be able to model the policy directly. Different from the value-based method, the policy-based algorithm does not need to estimate the value function, but directly fits the policy function, updates the policy parameters through training, and directly generates the best policy. In policy-based methods, we input a state and output the corresponding action directly, rather than the value $V(s)$ or Q value $Q(s, a)$ of the state. One of the most representative algorithms is strategy gradient, which is also the most basic policy-based algorithm.

Policy gradient chooses to optimize the policy directly and update the parameters of the policy network by calculating the gradient of expected reward[29]. Therefore, its objective function $J(\theta)$ is directly designed as expected cumulative rewards, *i.e.*,

$$J(\theta) = \mathbb{E}_{\tau \_\theta(\tau)} [r(\tau)] = \int_{\tau\ \pi(\tau)} r(\tau)\, \pi_\theta(\tau)\, d\tau \quad .$$

By taking the derivative of $J(\theta)$, we get

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\ \pi_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(A_t | S_t\right) \sum_{t=1}^{T} r\left(S_t, A_t\right) \right].$$

The above formula consists of two parts. One is $\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(A_t | S_t\right)$ which denotes the probability of the gradient in the current trace. The other is $\sum_{t=1}^{T} r\left(S_t, A_t\right)$ which represents the return of the current trace. Since the return is total rewards and can only be obtained after one episode, the policy gradient algorithm can only be updated for each episode, not for each time step.

The expected value can be expressed in a variety of ways, corresponding to different ways of calculating the loss function. The advantage of the strategy gradient algorithm is that it can be applied in the continuous action space. In addition, the change of the action probability is smoother, and the convergence is better guaranteed.

REINFORCE algorithm is a classic policy gradient algorithm[30]. Since the expected value of the cumulative reward cannot be calculated directly, the Monte Carlo method is applied to approximate the average value of multiple samples. REINFORCE updates the unbiased estimate of the gradient by using Monte Carlo sampling.

Each sampling generates a trajectory, which runs iteratively. After obtaining a large number of trajectories, the cumulative reward can be calculated by using certain transformations and approximations as the loss function for gradient update. However, the variance of this algorithm is large since it needs to interact with the environment until the terminate state. The reward for each interaction is a random variable, so each variance will add up when the variance is calculated. In particular, the REINFORCE algorithm has three steps:

- Step 1: sample τ_i from $\pi_\theta\left(A_t|S_t\right)$
- Step 2: $\nabla_\theta J\left(\theta\right) \approx \sum_i \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(A_t^i|S_t^i\right) \sum_{t=1}^{T} r\left(S_t^i, A_t^i\right)\right]$
- Step 3: $theta \leftarrow \theta + \alpha \nabla_\theta J\left(\theta\right)$

The two algorithms, value-based and policy-based methods, both have their own characteristics and disadvantages. Firstly, the disadvantages of the value-based methods are that the output of the action cannot be obtained directly, and it is difficult to extend to the continuous action space. The value-based methods can also lead to the problem of high bias, *i.e.*, it is difficult to eliminate the error between the estimated value function and the actual value function. For the policy-based methods, a large number of trajectories must be sampled, and the difference between each trajectory may be huge. As a result, high variance and large gradient noise are introduced. It leads to the instability of training and the difficulty of policy convergence.

3.2.3. Actor-critic methods

The actor-critic architecture combines the characteristics of the value-based and policy-based algorithms, and to a certain extent solves their respective weaknesses, as well as the contradictions between high variance and high bias. The constructed agent can not only directly output policies, but also evaluate the performance of the current policies through the value function. Specifically, the actor-critic architecture consists of an actor which is responsible for generating the policy and a critic to evaluate this policy. When the actor is performing, the critic should evaluate its performance, both of which are constantly being updated[31]. This complementary training is generally more effective than a policy-based method or value-based method.

In specific, the input of actor is state S_t, and the output is action A_t. The role of actor is to approximate the policy model $\pi_\theta\left(A_t|S_t\right)$. Critic uses the value function Q as the output to evaluate the value of the policy, and this Q value $Q\left(S_t, A_t\right)$ can be directly applied to calculate the loss function of actor. The gradient of the expected revenue function $J\left(\theta\right)$ in the action-critic framework is developed from the basic policy gradient algorithm. The policy gradient algorithm can only implement the update of each episode, and it is difficult to accurately feedback the reward. Therefore, it has poor training efficiency. Instead, the actor-critic algorithm replaces $\sum_{t=1}^{T} r\left(S_t^i, A_t^i\right)$ with $Q\left(S_t, A_t\right)$ to evaluate the expected returns of state-action tuple $\{S_t, A_t\}$ in the current time step t. The gradient of $J\left(\theta\right)$ can be expressed as

$$\nabla_\theta J\left(\theta\right) = \mathbb{E}_{\tau\ \pi_\theta(\tau)}\left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(A_t|S_t\right) Q\left(S_t, A_t\right)\right].$$

3.3. Deep reinforcement learning

With the continuous expansion of the application of deep learning, its wave also swept into the RL field, resulting in deep reinforcement learning (DRL), *i.e.*, using a multi-layer deep neural network to approximate value function or policy function in the RL algorithm[32,33]. DRL mainly solves the curse-of-dimensionality problem in real-world RL applications with large or continuous state and/or action space, where the traditional tabular RL algorithms cannot store and extract a large amount of feature information[17,34].

Q-learning, as a very classical algorithm in RL, is a good example to understand the purpose of DRL. The big issue with Q-learning falls into the tabular method, which means that when state and action spaces are very large, it cannot build a very large Q table to store a large number of Q values[35]. Besides, it counts and iterates

Table 1. Taxonomy of representative algorithms for DRL.

Types		Representative algorithms
Value-based		Deep Q-Network (DQN) [37], Double Deep Q-Network (DDQN) [39], DDQN with proportional prioritization [40]
Policy-based		REINFORCE [30], Q-prop [41]
Actor-critic		Soft Actor-Critic (SAC) [42], Asynchronous Advantage Actor Critic (A3C) [43], Deep Deterministic Policy Gradient (DDPG) [44], Distributed Distributional Deep Deterministic Policy Radients (D4PG) [45], Twin Delayed Deep Deterministic (TD3) [46], Trust Region Policy Optimization (TRPO) [47], Proximal Policy Optimization (PPO) [48]
Advanced	POMDP	Deep Belief Q-Network (DBQN) [49], Deep Recurrent Q-Network (DRQN) [50], Recurrent Deterministic Policy Gradients (RDPG) [51]
	Multi-agents	Multi-Agent Importance Sampling (MAIS) [52], Coordinated Multi-agent DQN [53], Multi-agent Fingerprints (MAF) [52], Counterfactual Multiagent Policy Gradient (COMAPG) [54], Multi-Agent DDPG (MADDPG) [55]

Q values based on past states. Therefore, on the one hand, the applicable state and action space of Q-learning is very small. On the other hand, if a state never appears, Q-learning cannot deal with it [36]. In other words, Q-learning has no prediction ability and generalization ability at this point.

In order to make Q-learning with prediction ability, considering that neural network can extract feature information well, deep Q network (DQN) is proposed by applying deep neural network to simulate Q value function. In specific, DQN is the continuation of Q-learning algorithm in continuous or large state space to approximate Q value function by replacing Q table with neural networks [37].

In addition to the value-based DRL algorithm such as DQN, we summarize a variety of classical DRL algorithms according to algorithm types by referring to some DRL related surveys [38] in Table 1, including not only the policy-based and actor-critic DRL algorithms, but also the advanced DRL algorithms of partially observable markov decision process (POMDP) and multi-agents.

4. FEDERATED REINFORCEMENT LEARNING

In this section, the detailed background and categories of FRL will be discussed.

4.1. Federated reinforcement learning background

Despite the excellent performance that RL and DRL have achieved in many areas, they still face several important technical and non-technical challenges in solving real-world problems. The successful application of FL in supervised learning tasks arouses interest in exploiting similar ideas in RL, *i.e*., FRL. On the other hand, although FL is useful in some specific situations, it fails to deal with cooperative control and optimal decision-making in dynamic environments [10]. FRL not only provides the experience for agents to learn to make good decisions in an unknown environment, but also ensures that the privately collected data during the agent's exploration does not have to be shared with others. A forward-looking and interesting research direction is how to conduct RL under the premise of protecting privacy. Therefore, it is proposed to use FL framework to enhance the security of RL and define FRL as a security-enhanced distributed RL framework to accelerate the learning process, protect agent privacy and handle not independent and identically distributed (Non-IID) data [8]. Apart from improving the security and privacy of RL, we believe that FRL has a wider and larger potential in helping RL to achieve better performance in various aspects, which will be elaborated in the following subsections.

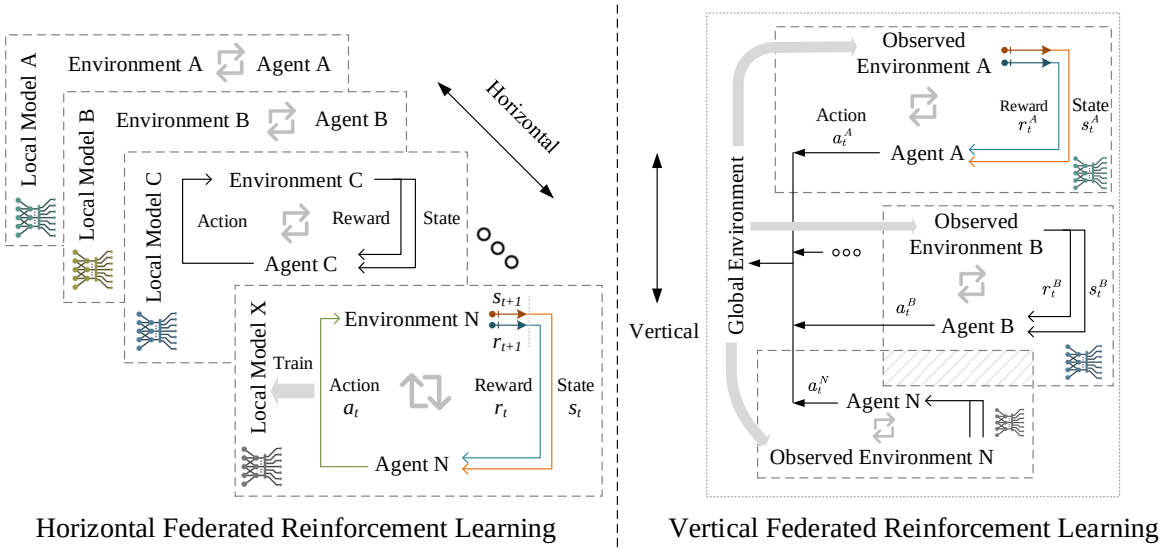


Figure 8. Comparison of horizontal federated reinforcement learning and vertical federated reinforcement learning.

In order to facilitate understanding and maintain consistency with FL, FRL is divided into two categories depending on environment partition [7], *i.e.*, HFRL and VFRL. Figure 8 gives the comparison between HFRL and VFRL. In HFRL, the environment that each agent interacts with is independent of the others, while the state space and action space of different agents are aligned to solve similar problems. The action of each agent only affects its own environment and results in corresponding rewards. As an agent can hardly explore all states of its environment, multiple agents interacting with their own copy of the environment can accelerate training and improve model performance by sharing experience. Therefore, horizontal agents use server-client model or peer-to-peer model to transmit and exchange the gradients or parameters of their policy models (actors) and/or value function models (critics). In VFRL, multiple agents interact with the same global environment, but each can only observe limited state information in the scope of its view. Agents can perform different actions depending on the observed environment and receive local reward or even no reward. Based on the actual scenario, there may be some observation overlap between agents. In addition, all agents' actions affect the global environment dynamics and total rewards. As opposed to the horizontal arrangement of independent environments in HFRL, the vertical arrangement of observations in VFRL poses a more complex problem and is less studied in the existing literature.

4.2. Horizontal federated reinforcement learning

HFRL can be applied in scenarios in which the agents may be distributed geographically, but they face similar decision-making tasks and have very little interaction with each other in the observed environments. Each participating agent independently executes decision-making actions based on the current state of environment and obtains positive or negative rewards for evaluation. Since the environment explored by one agent is limited and each agent is unwilling to share the collected data, multiple agents try to train the policy and/or value model together to improve model performance and increase learning efficiency. The purpose of HFRL is to alleviate the sample-efficiency problem in RL, and help each agent quickly obtain the optimal policy which can maximize the expected cumulative reward for specific tasks, while considering privacy protection.

In the HFRL problem, the environment, state space, and action space can replace the data set, feature space, and label space of basic FL. More formally, we assume that N agents $\{\mathcal{F}_i\}_{i=1}^N$ can observe the environment $\{\mathcal{E}_i\}_{i=1}^N$ within their field of vision. $\mathcal{G}$ denotes the collection of all environments. The environment $\mathcal{E}_i$ where the i-th agent is located has a similar model, *i.e.*, state transition probability and reward function compared to other environments. Note that the environment $\mathcal{E}_i$ is independent of the other environments, in that the state

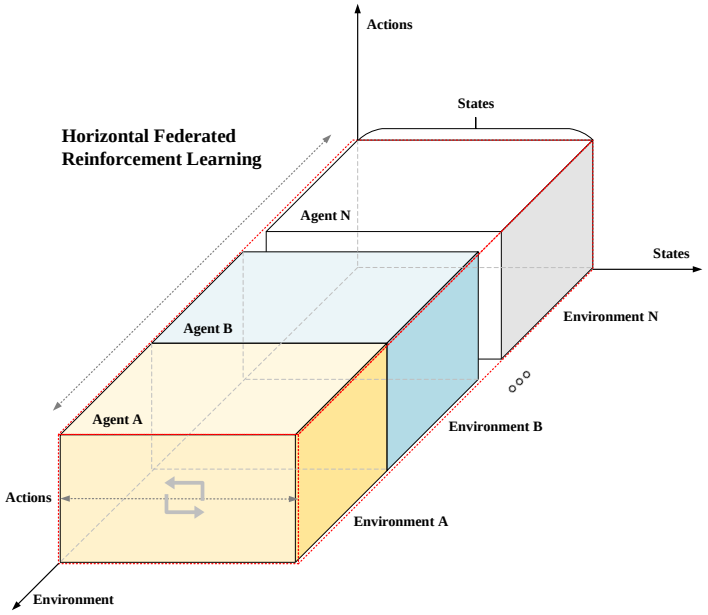


Figure 9. Illustration of horizontal federated reinforcement learning.

transition and reward model of $\mathcal{E}_i$ do not depend on the states and actions of the other environments. Each agent $\mathcal{F}_i$ interacts with its own environment $\mathcal{E}_i$ to learn an optimal policy. Therefore, the conditions for HFRL are presented as follows, *i.e.*,

$$\mathcal{S}_i = \mathcal{S}_j, \mathcal{A}_i = \mathcal{A}_j, \mathcal{E}_i \neq \mathcal{E}_j, \forall i, j \in \{1,2,...,N\}, \mathcal{E}_i, \mathcal{E}_j \in \mathcal{G}, i \neq j,$$

where $\mathcal{S}_i$ and $\mathcal{S}_j$ denote the similar state space encountered by the i-th agent and j-th agent, respectively. $\mathcal{A}_i$ and $\mathcal{A}_j$ denote the similar action space of the i-th agent and j-th agent, respectively The observed environment $\mathcal{E}_i$ and $\mathcal{E}_i$ are two different environments that are assumed to be independent and ideally identically distributed.

Figure 9 shows the HFRL in graphic form. Each agent is represented by a cuboid. The axes of the cuboid denote three dimensions of information, *i.e.*, the environment, state space, and action space. We can intuitively see that all environments are arranged horizontally, and multiple agents have aligned state and action spaces. In other words, each agent explores independently in its respective environment, and needs to obtain optimal strategies for similar tasks. In HFRL, agents share their experiences by exchanging masked models to enhance sample efficiency and accelerate the learning process.

A typical example of HFRL is the autonomous driving system in IoV. As vehicles drive on roads throughout the city and country, they can collect various environmental information and train the autonomous driving models locally. Due to driving regulations, weather conditions, driving routes, and other factors, one vehicle cannot be exposed to every possible situation in the environment. Moreover, the vehicles have basically the same operations, including braking, acceleration, steering, *etc*. Therefore, vehicles driving on different roads, different cities, or even different countries could share their learned experience with each other by FRL without revealing their driving data according to the premise of privacy protection. In this case, even if other vehicles have never encountered a situation, they can still perform the best action by using the shared model. The exploration of multiple vehicles together also creates an increased chance of learning rare cases to ensure the reliability of the model.

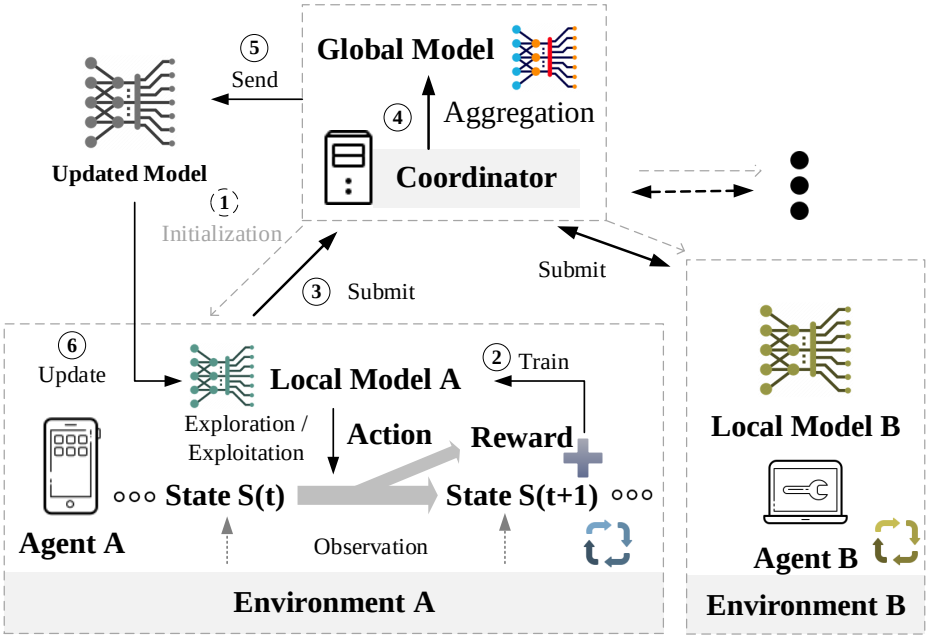


Figure 10. An example of horizontal federated reinforcement learning architecture.

For a better understanding of HFRL, Figure 10 shows an example of HFRL architecture using the server-client model. The coordinator is responsible for establishing encrypted communication with agents and implementing aggregation of shared models. The multiple parallel agents may be composed of heterogeneous equipment (*e.g.*, IoT devices, smart phone and computers, *etc.*) and distributed geographically. It is worth noting that there is no specific requirement for the number of agents, and agents are free to choose to join or leave. The basic procedure for conducting HFRL can be summarized as follows.

- Step 1: The initialization/join process can be divided into two cases, one is when the agent has no model locally, and the other is when the agent has a model locally. For the first case, the agent can directly download the shared global model from a coordinator. For the second case, the agent needs to confirm the model type and parameters with the central coordinator.
- Step 2: Each agent independently observes the state of the environment and determines the private strategy based on the local model. The selected action is evaluated by the next state and received reward. All agents train respective models in state-action-reward-state (SARS) cycles.
- Step 3: Local model parameters are encrypted and transmitted to the coordinator. Agents may submit local models at any time as long as the trigger conditions are met.
- Step 4: The coordinator conducts the specific aggregation algorithm to evolve the global federated model. Actually, there is no need to wait for submissions from all agents, and appropriate aggregation conditions can be formulated depending on communication resources.
- Step 5: The coordinator sends back the aggregated model to the agents.
- Step 6: The agents improve their respective models by fusing the federated model.

Following the above architecture and process, applications suitable for HFRL should meet the following characteristics. First, agents have similar tasks to make decisions under dynamic environments. Different from the FL setting, the goal of the HFRL-based application is to find the optimal strategy to maximize reward in the future. For the agent to accomplish the task requirements, the optimal strategy directs them to perform certain actions, such as control, scheduling, navigation, *etc.* Second, distributed agents maintain independent observations. Each agent can only observe the environment within its field of view, but does not ensure that the collected data follows the same distribution. Third, it is important to protect the data that each agent collects

and explores. Agents are presumed to be honest but curious, *i.e*., they honestly follow the learning mechanism but are curious about private information held by other agents. Due to this, the data used for training is only stored at the owner and is not transferred to the coordinator. HFRL provides an implementation method for sharing experiences under the constraints of privacy protection. Additionally, various reasons limit the agent's ability to explore the environment in a balanced manner. Participating agents may include heterogeneous devices. The amount of data collected by each agent is unbalanced due to mobility, observation, energy and other factors. However, all participants have sufficient computing, storage, and communication capabilities. These capabilities assist the agent in completing model training, merging, and other basic processes. Finally, the environment observed by a agent may change dynamically, causing differences in data distribution. The participating agents need to update the model in time to quickly adapt to environmental changes and construct a personalized local model.

In existing RL studies, some applications that meet the above characteristics can be classified as HFRL. Nadiger *et al*.[56] presents a typical HFRL architecture, which includes the grouping policy, the learning policy, and the federation policy. In this work, RL is used to show the applicability of granular personalization and FL is used to reduce training time. To demonstrate the effectiveness of the proposed architecture, a non-player character in the Atari game Pong is implemented and evaluated. In the study from Liu *et al*.[57], the authors propose the lifelong federated reinforcement learning (LFRL) for navigation in cloud robotic systems. It enables the robot to learn efficiently in a new environment and use prior knowledge to quickly adapt to the changes in the environment. Each robot trains a local model according to its own navigation task, and the centralized cloud server implements a knowledge fusion algorithm for upgrading a shared model. In considering that the local model and the shared model might have different network structures, this paper proposes to apply transfer learning to improve the performance and efficiency of the shared model. Further, researchers also focus on HFRL-based applications in the IoT due to the high demand for privacy protection. Ren *et al*.[58] suggest deploying the FL architecture between edge nodes and IoT devices for computation offloading tasks. IoT devices can download RL model from edge nodes and train the local model using own data, including the remained energy resources and the workload of IoT device, *etc*. The edge node aggregates the updated private model into the shared model. Although this method considers privacy protection issues, it requires further evaluation regarding the cost of communication resources by the model exchange. In addition, the work[59] proposes a federated deep-reinforcement-learning-based framework (FADE) for edge caching. Edge devices, including base stations (BSs), can cooperatively learn a predictive model using the first round of training parameters for local learning, and then upload the local parameters tuned to the next round of global training. By keeping the training on local devices, the FADE can enable fast training and decouple the learning process between the cloud and data owner in a distributed-centralized manner. More HFRL-based applications will be classified and summarized in the next section.

Prior to HFRL, a variety of distributed RL algorithms have been extensively investigated, which are closely related to HFRL. In general, distributed RL algorithms can be divided into two types: synchronized and asynchronous. In synchronous RL algorithms, such as Sync-Opt synchronous stochastic optimization (Sync-Opt)[60] and parallel advantage actor critic (PAAC)[3], the agents explore their own environments separately, and after a number of samples are collected, the global parameters are updated synchronously. On the contrary, the coordinator will update the global model immediately after receiving the gradient from an arbitrary agent in asynchronous RL algorithms, rather than waiting for other agents. Several asynchronous RL algorithms are presented, including A3C[61], Impala[62], Ape-X[63] and general reinforcement learning architecture (Gorila)[1]. From the perspective of technology development, HFRL can also be considered security-enhanced parallel RL. In parallel RL, multiple agents interact with a stochastic environment to seek the optimal policy for the same task[1,2]. By building a closed loop of data and knowledge in parallel systems, parallel RL helps determine the next course of action for each agent. The state and action representations are fed into a designed neural network to approximate the action value function[64]. However, parallel RL typically transfers

the experience of agent without considering privacy protection issues[7]. In the implementation of HFRL, further restrictions accompany privacy protection and communication consumption to adapt to special scenarios, such as IoT applications[59]. In addition, another point to consider is Non-IID data. In order to ensure convergence of the RL model, it is generally assumed in parallel RL that the states transitions in the environment follow the same distribution, *i.e.*, the environments of different agents are IID. But in actual scenarios, the situation faced by agents may differ slightly, so that the models of environments for different agents are not identically distributed. Therefore, HFRL needs to improve the generalization ability of the model compared with parallel RL to meet the challenges posed by Non-IID data.

Based on the potential issues faced by the current RL technology, the advantages of HFRL can be summarized as follows.

- Enhancing training speed. In the case of a similar target task, multiple agents sharing training experiences gained from different environments can expedite the learning process. The local model rapidly evolves through aggregation and update algorithms to assess the unexplored environment. Moreover, the data obtained by different agents are independent, reducing correlations between the observed data. Furthermore, this also helps to solve the issue of unbalanced data caused by various restrictions.
- Improving the reliability of model. When the dimensions of the state of the environment are enormous or even uncountable, it is difficult for a single agent to train an optimal strategy for situations with extremely low occurrence probabilities. Horizontal agents are exploring independently while building a cooperative model to improve the local model's performance on rare states.
- Mitigating the problems of devices heterogeneity. Different devices deploying RL agents in the HFRL architecture may have different computational and communication capabilities. Some devices may not meet the basic requirements for training, but strategies are needed to guide actions. HFRL makes it possible for all agents to obtain the shared model equally for the target task.
- Addressing the issue of non-identical environment. Considering the differences in the environment dynamics for the different agents, the assumption of IID data may be broken. Under the HFRL architecture, agents in not identically-distributed environment models can still cooperate to learn a federated model. In order to address the difference in environment dynamics, a personalized update algorithm of local model could be designed to minimize the impact of this issue.
- Increasing the flexibility of the system. The agent can decide when to participate in the cooperative system at any time, because HFRL allows asynchronous requests and aggregation of shared models. In the existing HFRL-based application, new agents also can apply for membership and benefit from downloading the shared model.

4.3. Vertical federated reinforcement learning

In VFL, samples of multiple data sets have different feature spaces but these samples may belong to the same groups or common users. The training data of each participant are divided vertically according to their features. More general and accurate models can be generated by building heterogeneous feature spaces without releasing private information. VFRL applies the methodology of VFL to RL and is suitable for POMDP scenarios where different RL agents are in the same environment but have different interactions with the environment. Specifically, different agents could have different observations that are only part of the global state. They could take actions from different action spaces and observe different rewards, or some agents even take no actions or cannot observe any rewards. Since the observation range of a single agent to the environment is limited, multiple agents cooperate to collect enough knowledge needed for decision making. The role of FL in VFRL is to aggregate the partial features observed by various agents. Especially for those agents without rewards, the aggregation effect of FL greatly enhances the value of such agents in their interactions with the environment, and ultimately helps with the strategy optimization. It is worth noting that in VFRL the issue of privacy protection needs to be considered, *i.e.*, private data collected by some agents do not have to be shared with others. Instead, agents can transmit encrypted model parameters, gradients, or direct mid-product to each other. In

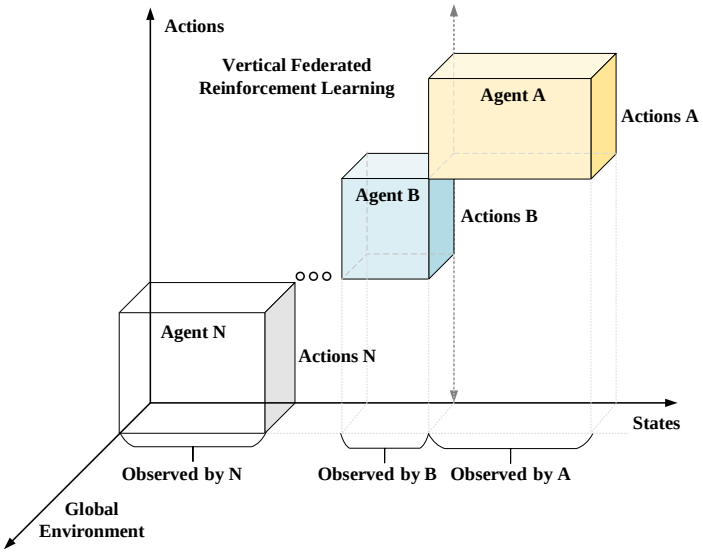


Figure 11. Illustration of vertical federated reinforcement learning.

short, the goal of VFRL is for agents interacting with the same environment to improve the performance of their policies and the effectiveness in learning them by sharing experiences without compromising the privacy.

More formally, we denote $\{\mathcal{F}_i\}_{i=1}^N$ as N agents in VFRL, which interact with a global environment $\mathcal{E}$. The i-th agent $\mathcal{F}_i$ is located in the environment $\mathcal{E}_i = \mathcal{E}$, obtains the local partial observation $\mathcal{O}_i$, and can perform the set of actions $\mathcal{A}_i$. Different from HFRL, the state/observation and action spaces of two agents $\mathcal{F}_i$ and $\mathcal{F}_j$ may be not identical, but the aggregation of the state/observation spaces and action spaces of all the agents constitutes the global state and action spaces of the global environment $\mathcal{E}$. The conditions for VFRL can be defined as *i.e.*,

$$\mathcal{O}_i \neq \mathcal{O}_j, \mathcal{A}_i \neq \mathcal{A}_j, \mathcal{E}_i = \mathcal{E}_j = \mathcal{E}, \bigcup_{i=1}^{N} \mathcal{O}_i = \mathcal{S}, \bigcup_{i=1}^{N} \mathcal{A}_i = \mathcal{A}, \forall i, j \in \{1,2,...,N\}, i \neq j,$$

where $\mathcal{S}$ and $\mathcal{A}$ denote the global state space and action space of all participant agents respectively. It can be seen that all the observations of the N agents together constitute the global state space $\mathcal{S}$ of the environment $\mathcal{E}$. Besides, the environments $\mathcal{E}_i$ and $\mathcal{E}_j$ are the same environment $\mathcal{E}$. In most cases, there is a great difference between the observations of two agents $\mathcal{F}_i$ and $\mathcal{F}_j$.

Figure 11 shows the architecture of VFRL. The dataset and feature space in VFL are converted to the environment and state space respectively. VFL divides the dataset vertically according to the features of samples, and VFRL divides agents based on the state spaces observed from the global environment. Generally speaking, every agent has its local state which can be different from that of the other agents and the aggregation of these local partial states corresponds to the entire environment state [65]. In addition, after interacting with the environment, agents may generate their local actions which correspond to the labels in VFL.

Two types of agents can be defined for VFRL, *i.e.*, decision-oriented agents and support-oriented agents. Decision-oriented agents $\{\mathcal{F}_i\}_{i=1}^K$ can interact with the environment $\mathcal{E}$ based on their local state $\{\mathcal{S}_i\}_{i=1}^K$ and action $\{\mathcal{A}_i\}_{i=1}^K$. Meanwhile, support-oriented agents $\{\mathcal{F}_i\}_{i=K+1}^N$ take no actions and receive no rewards but only the observations of the environment, *i.e.*, their local states $\{\mathcal{S}_i\}_{i=K+1}^N$. In general, the following six steps, as shown in Figure 12, are the basic procedure for VFRL, *i.e.*,

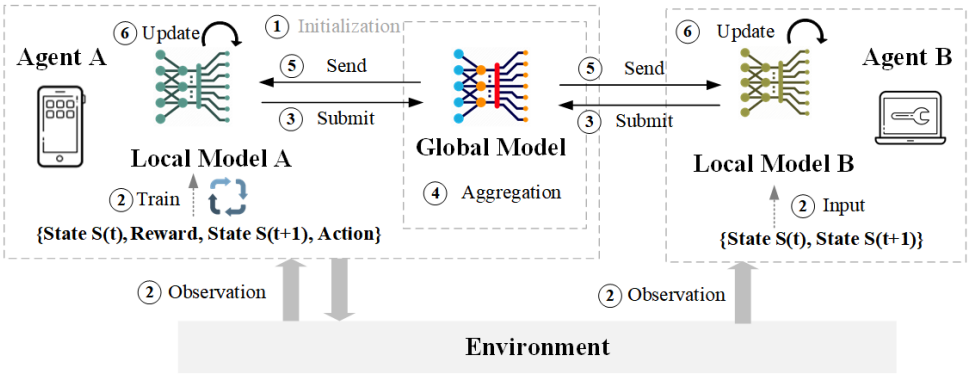


Figure 12. An example of vertical federated reinforcement learning architecture.

- Step 1: Initialization is performed for all agent models.
- Step 2: Agents obtain states from the environment. For decision-oriented agents, actions are obtained based on the local models, and feedbacks are obtained through interactions with the environment, *i.e.*, the states of the next time step and rewards. The data tuple of state-action-reward-state (SARS) is used to train the local models.
- Step 3: All agents calculate the mid-products of the local models and then transmit the encrypted mid-products to the federated model.
- Step 4: The federated model performs the aggregation calculation for mid-products and trains the federated model based on the aggregation results.
- Step 5: Federated model encrypts model parameters such as weight and gradient and passes them back to other agents.
- Step 6: All agents update their local models based on the received encrypted parameters.

As an example of VFRL, consider a microgrid (MG) system including household users, the power company, and the photovoltaic (PV) management company as the agents. All the agents observe the same MG environment while their local state spaces are quite different. The global states of the MG system generally consist of several dimensions/features, *i.e.*, state-of-charge (SOC) of the batteries, load consumption of the household users, power generation from PV, etc. The household agents can obtain the SOC of their own batteries and their own load consumption, the power company can know the load consumption of all the users, and PV management company can know the power generation of PV. As to the action, the power company needs to make decisions on the power dispatch of the diesel generators (DG), and the household users can make decisions to manage their electrical utilities with demand response. Finally, the power company can observe rewards such as the cost of DG power generation, the balance between power generation and consumption, and the household users can observe rewards such as their electricity bill that is related to their power consumption. In order to learn the optimal policies, these agents need to communicate with each other to share their observations. However, PV managers do not want to expose their data to other companies, and household users also want to keep their consumption data private. In this way, VFRL is suitable to achieve this goal and can help improve policy decisions without exposing specific data.

Compared with HFRL, there are currently few works on VFRL. Zhuo *et al.*[65] present the federated deep reinforcement learning (FedRL) framework. The purpose of this paper is to solve the challenge where the feature space of states is small and the training data are limited. Transfer learning approaches in DRL are also solutions for this case. However, when considering the privacy-aware applications, directly transferring data or models should not be used. Hence, FedRL combines the advantage of FL with RL, which is suitable for the case when agents need to consider their privacy. FedRL framework assumes agents cannot share their partial observations of the environment and some agents are unable to receive rewards. It builds a shared value

network, *i.e*., multiLayer perceptron (MLP), and takes its own Q-network output and encryption value as input to calculate a global Q-network output. Based on the output of global Q-network, the shared value network and self Q-network are updated. Two agents are used in the FedRL algorithm, *i.e*., agent α and β, which interact with the same environment. However, agent β cannot build its own policies and rewards. Finally, FedRL is applied in two different games, *i.e*., Grid-World and Text2Action, and achieves better results than the other baselines. Although the VFRL model in this paper only contains two agents, and the structure of the aggregated neural network model is relatively simple, we believe that it is a great attempt to first implement VFRL and verify its effectiveness.

Multi-agent RL (MARL) is very closely related to VFRL. As the name implies, MARL takes into account the existence of multiple agents in the RL system. However, the empirical evaluation shows that applying the simple single-agent RL algorithms directly to scenarios of multiple agents cannot converge to the optimal solution, since the environment is no longer static from the perspective of each agent[66]. In specific, the action of each agent will affect the next state, thus affecting all agents in the future time step[67]. Besides, the actions performed by one certain agent will yield different rewards depending on the actions taken by other agents. This means that agents in MARL correlate with each other, rather than being independent of each other. This challenge, called as the non-stationarity of the environment, is the main problem to be solved in the development of an efficient MARL algorithm[68].

MARL and VFRL both study the problem of multiple agents learning concurrently how to solve a task by interacting with the same environment[69]. Since MARL and VFRL have a large range of similarities, the review of MARL's related works is a very useful guide to help researchers summarize the research focus and better understand VFRL. There is abundant literature related to MARL. However, most MARL research[70–73] is based on a fully observed markov decision process (MDP), where each agent is assumed to have the global observation of the system state[68]. These MARL algorithms are not applicable to the case of POMDP where the observations of individual agents are often only a part of the overall environment[74]. Partial observability is a crucial consideration for the development of algorithms that can be applied to real-world problems[75]. Since VFRL is mainly oriented towards POMDP scenarios, it is more important to analyze the related works of MARL based on POMDP as the guidance of VFRL.

Agents in the above scenarios partially observe the system state and make decisions at each step to maximize the overall rewards for all agents, which can be formalized as a decentralized partially observable markov decision process (Dec-POMDP)[76]. Optimally addressing a Dec-POMDP model is well known to be a very challenging problem. In the early works, Omidshafiei *et al*.[77] proposes a two-phase MT-MARL approach that concludes the methods of cautiously-optimistic learners for action-value approximation and concurrent experience replay trajectories (CERTs) as the experience replay targeting sample-efficient and stable MARL. The authors also apply the recursive neural network (RNN) to estimate the non-observed state and hysteretic Q-learning to address the problem of non-stationarity in Dec-POMDP. Han *et al*.[78] designs a neural network architecture, IPOMDP-net, which extends QMDP-net planning algorithm[79] to MARL settings under POMDP. Besides, Mao *et al*.[80] introduces the concept of information state embedding to compress agents' histories and proposes an RNN model combining the state embedding. Their method, *i.e*., embed-then-learn pipeline, is universal since the embedding can be fed into any existing partially observable MARL algorithm as the black-box. In the study from Mao *et al*.[81], the proposed Attention MADDPG (ATT-MADDPG) has several critic networks for various agents under POMDP. A centralized critic is adopted to collect the observations and actions of the teammate agents. Specifically, the attention mechanism is applied to enhance the centralized critic. The final introduced work is from Lee *et al*.[82]. They present an augmenting MARL algorithm based on pretraining to address the challenge in disaster response. It is interesting that they use behavioral cloning (BC), a supervised learning method where agents learn their policy from demonstration samples, as the approach to pretrain the neural network. BC can generate a feasible Dec-POMDP policy from demonstration samples,

which offers advantages over plain MARL in terms of solution quality and computation time.

Some MARL algorithms also concentrate on the communication issue of POMDP. In the study from Sukhbaatar *et al.*[83], communication between the agents is performed for a number of rounds before their action is selected. The communication protocol is learned concurrently with the optimal policy. Foerster *et al.*[84] proposes a deep recursive network architecture, *i.e.*, deep distributed recurrent Q-network (DDRQN), to address the communication problem in a multi-agent partially-observable setting. This work makes three fundamental modifications to previous algorithms. The first one is last-action inputs, which let each agent access its previous action as an input for the next time-step. Besides, inter-agent weight sharing allows diverse behavior between agents, as the agents receive different observations and thus evolve in different hidden states. The final one is disabling experience replay, which is because the non-stationarity of the environment renders old experiences obsolete or misleading. Foerster *et al.*[84] considers the communication task of fully cooperative, partially observable, sequential multi-agent decision-making problems. In their system model, each agent can receive a private observation and take actions that affect the environment. In addition, the agent can also communicate with its fellow agents via a discrete limited-bandwidth channel. Despite the partial observability and limited channel capacity, authors achieved the task that the two agents could discover a communication protocol that enables them to coordinate their behavior based on the approach of deep recurrent Q-networks.

While there are some similarities between MARL and VFRL, several important differences have to be paid attention to, *i.e.*,

- VFRL and some MARL algorithms are able to address similar problems, *e.g.*, the issues of POMDP. However, there are differences between the solution ideas between two algorithms. Since VFRL is the product of applying VFL to RL, the FL component of VFRL focuses more on the aggregation of partial features, including states and rewards, observed by different agents since VFRL inception. Security is also an essential issue in VFRL. On the contrary, MARL may arise as the most natural way of adding more than one agent in a RL system[85]. In MARL, agents not only interact with the environment, but also have complex interactive relationships with other agents, which creates a great obstacle to the solution of policy optimization. Therefore, the original intentions of two algorithms are different.
- Two algorithms are slightly different in terms of the structure. The agents in MARL must surely have the reward even some of them may not have their own local actions. However, in some cases, the agents in VFRL are not able to generate a corresponding operation policy, so in these cases, some agents have no actions and rewards[65]. Therefore, VFRL can solve more extensive problems that MARL is not capable of solving.
- Both two algorithms involve the communication problem between agents. In MARL, information such as the states of other agents and model parameters can be directly and freely propagated among agents. During communication, some MARL methods such as DDRQN in the work of Foerster *et al.*[84] consider the previous action as an input for the next time-step state. Weight sharing is also allowed between agents. However, VFRL assumes states cannot be shared among agents. Since these agents do not exchange experience and data directly, VFRL focuses more on security and privacy issues of communication between agents, as well as how to process mid-products transferred by other agents and aggregate federated models.

In summary, as a potential and notable algorithm, VFRL has several advantages as follows, *i.e.*,

- Excellent privacy protection. VFRL inherits the FL algorithm's idea of data privacy protection, so for the task of multiple agents cooperation in the same environment, information interaction can be carried out confidently to enhance the learning efficiency of RL model. In this process, each participant does not have to worry about any leakage of raw real-time data.
- Wide application scenarios. With appropriate knowledge extraction methods, including algorithm design and system modeling, VFRL can solve more real-world problems compared with MARL algorithms. This

is because VFRL can consider some agents that cannot generate rewards into the system model, so as to integrate their partial observation information of the environment based on FL while protecting privacy, train a more robust RL agent, and further improve learning efficiency.

4.4. Other types of FRL

The above HFRL or VFRL algorithms borrow ideas from FL for federation between RL agents. Meanwhile, there are also some existing works on FRL that are less affected by FL. Hence, they do not belong to either HFRL or VFRL, but federation between agents is also implemented.

The study from Hu *et al.*[86] is a typical example, which proposes a reward shaping based general FRL algorithm, called federated reward shaping (FRS). It uses reward shaping to share federated information to improve policy quality and training speed. FRS adopts the server-client architecture. The server includes the federated model, while each client completes its own tasks based on the local model. This algorithm can be combined with different kinds of RL algorithms. However, it should be noted that FRS focuses on reward shaping, this algorithm cannot be used when there is no reward in some agents in VFRL. In addition, FRS performs knowledge aggregation by sharing high-level information such as reward shaping value or embedding between client and server instead of sharing experience or policy directly. The convergence of FRS is also guaranteed since only minor changes are made during the learning process, which is the modification of the reward in the replay buffer.

As another example, Anwar *et al.*[87] achieves federation between agents by smoothing the average weight. This work analyzes the Multi-task FRL algorithms (MT-FedRL) with adversaries. Agents only interact and make observations in their environment, which can be featured by different MDPs. Different from HFRL, the state and action spaces do not need to be the same in these environments. The goal of MT-FedRL is to learn a unified policy, which is jointly optimized across all of the environments. MT-FedRL adopts policy gradient methods for RL. In other words, policy parameter is needed to learn the optimal policy. The server-client architecture is also applied and all agents should share their own information with a centralized server. The role of non-negative smoothing average weights is to achieve a consensus among the agents' parameters. As a result, they can help to incorporate the knowledge from other agents as the process of federation.

5. APPLICATIONS OF FRL

In this section, we provide an extensive discussion of the application of FRL in a variety of tasks, such as edge computing, communications, control optimization, attack detection, *etc*. This section is aimed at enabling readers to understand the applicable scenarios and research status of FRL.

5.1. FRL for edge computing

In recent years, edge equipment, such as BSs and road side units (RSUs), has been equipped with increasingly advanced communication, computing and storage capabilities. As a result, edge computing is proposed to delegating more tasks to edge equipment in order to reduce the communication load and reduce the corresponding delay. However, the issue of privacy protection remains challenging since it may be untrustworthy for the data owner to hand off their private information to a third-party edge server[4]. FRL offers a potential solution for achieving privacy-protected intelligent edge computing, especially in decision-making tasks like caching and offloading. Additionally, the multi-layer processing architecture of edge computing is also suitable for implementing FRL through the server-client model. Therefore, many researchers have focused on applying FRL to edge computing.

The distributed data of large-scale edge computing architecture makes it possible for FRL to provide distributed intelligent solutions to achieve resource optimization at the edge. For mobile edge networks, a potential FRL

framework is presented for edge system[88], named as "In-Edge AI", to address optimization of mobile edge computing, caching, and communication problems. The authors also propose some ideas and paradigms for solving these problems by using DRL and Distributed DRL. To carry out dynamic system-level optimization and reduce the unnecessary transmission load, "In-Edge AI" framework takes advantage of the collaboration among edge nodes to exchange learning parameters for better training and inference of models. It has been evaluated that the framework has high performance and relatively low learning overhead, while the mobile communication system is cognitive and adaptive to the environment. The paper provides good prospects for the application of FRL to edge computing, but there are still many challenges to overcome, including the adaptive improvement of the algorithm, and the training time of the model from scratch *etc*.

Edge caching has been considered a promising technique for edge computing to meet the growing demands for next-generation mobile networks and beyond. Addressing the adaptability and collaboration challenges of the dynamic network environment, Wang *et al.*[89] proposes a device-to-device (D2D)-assisted heterogeneous collaborative edge caching framework. User equipment (UE) in a mobile network uses the local DQN model to make node selection and cache replacement decisions based on network status and historical information. In other words, UE decides where to fetch content and which content should be replaced in its cache list. The BS calculates aggregation weights based on the training evaluation indicators from UE. To solve the long-term mixed-integer linear programming problem, the attention-weighted federated deep reinforcement learning (AWFDRL) is presented, which optimizes the aggregation weights to avoid the imbalance of the local model quality and improve the learning efficiency of the DQN. The convergence of the proposed algorithm is verified and simulation results show that the AWFDRL framework can perform well on average delay, hit rate, and offload traffic.

A federated solution for cooperative edge caching management in fog radio access networks (F-RANs) is proposed[90]. Both edge computing and fog computing involve bringing intelligence and processing to the origins of data. The key difference between the two architectures is where the computing node is positioned. A dueling deep Q-network based cooperative edge caching method is proposed to overcome the dimensionality curse of RL problem and improve caching performance. Agents are developed in fog access points (F-APs) and allowed to build a local caching model for optimal caching decisions based on the user content request and the popularity of content. HFRL is applied to aggregate the local models into a global model in the cloud server. The proposed method outperforms three classical content caching methods and two RL algorithms in terms of reducing content request delays and increasing cache hit rates.

For edge-enabled IoT, Majidi *et al.*[91] proposes a dynamic cooperative caching method based on hierarchical federated deep reinforcement learning (HFDRL), which is used to determine which content should be cached or evicted by predicting future user requests. Edge devices that have a strong relationship are grouped into a cluster and one head is selected for this cluster. The BS trains the Q-value based local model by using BS states, content states, and request states. The head has enough processing and caching capabilities to deal with model aggregation in the cluster. By categorizing edge devices hierarchically, HFDRL improves the response time delay to keeps both small and large clusters from experiencing the disadvantages they could encounter. Storage partitioning allows content to be stored in clusters at different levels using the storage space of each device. The simulation results show the proposed method using MovieLens datasets improves the average content access delay and the hit rate.

Considering the low latency requirements and privacy protection issue of IoV, the study of efficient and secure caching methods has attracted many researchers. An FRL-empowered task caching problem with IoV has been analyzed by Zhao *et al.*[92]. The work proposes a novel cooperative caching algorithm (CoCaRL) for vehicular networks with multi-level FRL to dynamically determine which contents should be replaced and where the content requests should be served. This paper develops a two-level aggregation mechanism for

federated learning to speed up the convergence rate and reduces communication overhead, while DRL task is employed to optimize the cooperative caching policy between RSUs of vehicular networks. Simulation results show that the proposed algorithm can achieve a high hit rate, good adaptability and fast convergence in a complex environment.

Apart from caching services, FRL has demonstrated its strong ability to facilitate resource allocation in edge computing. In the study from Zhu *et al.*[93], the authors specifically focus on the data offloading task for mobile edge computing (MEC) systems. To achieve joint collaboration, the heterogeneous multi-agent actor-critic (H-MAAC) framework is proposed, in which edge devices independently learn the interactive strategies through their own observations. The problem is formulated as a multi-agent MDP for modeling edge devices' data allocation strategies, *i.e*., moving the data, locally executing or offloading to a cloud server. The corresponding joint cooperation algorithm that combines the edge federated model with the multi-agent actor-critic RL is also presented. Dual lightweight neural networks are built, comprising original actor/critic networks and target actor/critic networks.

Blockchain technology has also attracted lot attention from researchers in edge computing fields since it is able to provide reliable data management within the massive distributed edge nodes. In the study from Yu *et al.*[94], the intelligent ultra-dense edge computing (I-UDEC) framework is proposed, integrating with blockchain and RL technologies into 5G ultra-dense edge computing networks. In order to achieve low overhead computation offloading decisions and resource allocation strategies, authors design a two-timescale deep reinforcement learning (2Ts-DRL) approach, which consists of a fast-timescale and a slow-timescale learning process. The target model can be trained in a distributed manner via FL architecture, protecting the privacy of edge devices.

Additionally, to deal with the different types of optimization tasks, variants of FRL are being studied. Zhu *et al.*[95] presents a resource allocation method for edge computing systems, called concurrent federated reinforcement learning (CFRL). The edge node continuously receives tasks from serviced IoT devices and stores those tasks in a queue. Depending on its own resource allocation status, the node determines the scheduling strategy so that all tasks are completed as soon as possible. In case the edge host does not have enough available resources for the task, the task can be offloaded to the server. Contrary to the definition of the central server in the basic FRL, the aim of central server in CFRL is to complete the tasks that the edge nodes cannot handle instead of aggregating local models. Therefore, the server needs to train a special resource allocation model based on its own resource status, forwarded tasks and unique rewards. The main idea of CFRL is that edge nodes and the server cooperatively participate in all task processing in order to reduce total computing time and provide a degree of privacy protection.

5.2. FRL for communication networks

In parallel with the continuous evolution of communication technology, a number of heterogeneous communication systems are also being developed to adapt to different scenarios. Many researchers are also working toward intelligent management of communication systems. The traditional ML-based management methods are often inefficient due to their centralized data processing architecture and the risk of privacy leakage[5]. FRL can play an important role in services slicing and access controlling to replace centralized ML methods.

In communication network services, network function virtualization (NFV) is a critical component of achieving scalability and flexibility. Huang *et al.*[96] proposes a novel scalable service function chains orchestration (SSCO) scheme for NFV-enabled networks via FRL. In the work, a federated-learning-based framework for training global learning, along with a time-variant local model exploration, is designed for scalable SFC orchestration. It prevents data sharing among stakeholders and enables quick convergence of the global model. To reduce communication costs, SSCO allows the parameters of local models to be updated just at the beginning and end of each episode through distributed clients and the cloud server. A DRL approach is used to map

virtual network functions (VNFs) into networks with local knowledge of resources and instantiation cost. In addition, the authors also propose a loss-weight-based mechanism for generation and exploitation of reference samples for training in replay buffers, avoiding the strong relevance of each sample. Simulation results demonstrate that SSCO can significantly reduce placement errors and improve resource utilization ratios to place time-variant VNFs, as well as achieving desirable scalability.

Network slicing (NS) is also a form of virtual network architecture to support divergent requirements sustainably. The work from Liu *et al.*[97] proposes a device association scheme (such as access control and handover management) for radio access network (RAN) slicing by exploiting a hybrid federated deep reinforcement learning (HDRL) framework. In view of the large state-action space and variety of services, HDRL is designed with two layers of model aggregations. Horizontal aggregation deployed on BSs is used for the same type of service. Generally, data samples collected by different devices within the same service have similar features. The discrete-action DRL algorithm, *i.e*., DDQN, is employed to train the local model on individual smart devices. BS is able to aggregate model parameters and establish a cooperative global model. Vertical aggregation developed on the third encrypted party is responsible for the services of different types. In order to promote collaboration between devices with different tasks, authors aggregate local access features to form a global access feature, in which the data from different flows is strongly correlated since different data flows are competing for radio resources with each other. Furthermore, the Shapley value[98], which represents the average marginal contribution of a specific feature across all possible feature combinations, is used to reduce communication cost in vertical aggregation based on the global access feature. Simulation results show that HDRL can improve network throughput and communication efficiency.

The open radio access network (O-RAN) has emerged as a paradigm for supporting multi-class wireless services in 5G and beyond networks. To deal with the two critical issues of load balance and handover control, Cao *et al.*[99] proposes a federated DRL-based scheme to train the model for user access control in the O-RAN. Due to the mobility of UEs and the high cost of the handover between BSs, it is necessary for each UE to access the appropriate BS to optimize its throughput performance. As independent agents, UEs make access decisions with assistance from a global model server, which updates global DQN parameters by averaging DQN parameters of selected UEs. Further, the scheme proposes only partially exchanging DQN parameters to reduce communication overheads, and using the dueling structure to allow convergence for independent agents. Simulation results demonstrate that the scheme increases long-term throughput while avoiding frequent handovers of users with limited signaling overheads.

The issue of optimizing user access is important in wireless communication systems. FRL can provide interesting solutions for enabling efficient and privacy-enhanced management of access control. Zhang *et al.*[100] studies the problem of multi-user access in WIFI networks. In order to mitigate collision events on channel access, an enhanced multiple access mechanism based on FRL is proposed for user-dense scenarios. In particular, distributed stations train their local q-learning networks through channel state, access history and feedback from central access point (AP). AP uses the central aggregation algorithm to update the global model every period of time and broadcast it to all stations. In addition, a monte carlo (MC) reward estimation method for the training phase of local model is introduced, which allocates more weight to the reward of that current state by reducing the previous cumulative reward.

FRL is also studied for intelligent cyber-physical systems (ICPS), which aims to meet the requirements of intelligent applications for high-precision, low-latency analysis of big data. In light of the heterogeneity brought by multiple agents, the central RL-based resource allocation scheme has non-stationary issues and does not consider privacy issues. Therefore, the work from Xu *et al.*[101] proposes a multi-agent FRL (MA-FRL) mechanism which synthesizes a good inferential global policy from encrypted local policies of agents without revealing private information. The data resource allocation and secure communication problems are formulated as a

Stackelberg game with multiple participants, including near devices (NDs), far devices (FDs) and relay devices (RDs). Take into account the limited scope of the heterogeneous devices, the authors model this multi-agent system as a POMDP. Furthermore, it is proved that MA-FRL is μ-strongly convex and β-smooth and derives its convergence speed in expectation.

Zhang *et al.*[102] pays attention to the challenges in cellular vehicle-to-everything (V2X) communication for future vehicular applications. A joint optimization problem of selecting the transmission mode and allocating the resources is presented. This paper proposes a decentralized DRL algorithm for maximizing the amount of available vehicle-to-infrastructure capacity while meeting the latency and reliability requirements of vehicle-to-vehicle (V2V) pairs. Considering limited local training data at vehicles, the federated learning algorithm is conducted on a small timescale. On the other hand, the graph theory-based vehicle clustering algorithm is conducted on a large timescale.

The development of communication technologies in extreme environments is important, including deep underwater exploration. The architecture and philosophy of FRL are applied to smart ocean applications in the study of Kwon[103]. To deal with the nonstationary environment and unreliable channels of underwater wireless networks, the authors propose a multi-agent DRL-based algorithm that can realize FL computation with internet-of-underwater-things (IoUT) devices in the ocean environment. The cooperative model is trained by MADDPG for cell association and resource allocation problems. As for downlink throughput, it is found that the proposed MADDPG-based algorithm performed 80% and 41% better than the standard actor-critic and DDPG algorithms, respectively.

5.3. FRL for control optimization

Reinforcement learning based control schemes are considered as one of the most effective ways to learn a nonlinear control strategy in complex scenarios, such as robotics. Individual agent's exploration of the environment is limited by its own field of vision and usually needs a great deal of training to obtain the optimal strategy. The FRL-based approach has emerged as an appealing way to realize control optimization without exposing agent data or compromising privacy.

Automated control of robots is a typical example of control optimization problems. Liu *et al.*[57] discusses robot navigation scenarios and focuses on how to make robots transfer their experience so that they can make use of prior knowledge and quickly adapt to changing environments. As a solution, a cooperative learning architecture, called LFRL, is proposed for navigation in cloud robotic systems. Under the FRL-based architecture, the authors propose a corresponding knowledge fusion algorithm to upgrade the shared model deployed on the cloud. In addition, the paper also discusses the problems and feasibility of applying transfer learning algorithms to different tasks and network structures between the shared model and the local model.

FRL is combined with autonomous driving of robotic vehicles in the study of Liang *et al.*[104]. To reach rapid training from a simulation environment to a real-world environment, Liang *et al.*[104] presents a federated transfer reinforcement learning (FTRL) framework for knowledge extraction where all the vehicles make corresponding actions with the knowledge learned by others. The framework can potentially be used to train more powerful tasks by pooling the resources of multiple entities without revealing raw data information in real-life scenarios. To evaluate the feasibility of the proposed framework, authors perform real-life experiments on steering control tasks for collision avoidance of autonomous driving robotic cars and it is demonstrated that the framework has superior performance to the non-federated local training process. Note that the framework can be considered an extension of HFRL, because the target tasks to be accomplished are highly-relative and all observation data are pre-aligned.

FRL also appears as an attractive approach for enabling intelligent control of IoT devices without revealing

private information. Lim *et al.*[105] proposes a FRL architecture which allows agents working on independent IoT devices to share their learning experiences with each other, and transfer the policy model parameters to other agents. The aim is to effectively control multiple IoT devices of the same type but with slightly different dynamics. Whenever an agent meets the predefined criteria, its mature model will be shared by the server with all other agents in training. The agents continue training based on the shared model until the local model converges in the respective environment. The actor-critical proximal policy optimization (Actor-Critic PPO) algorithm is integrated into the control of multiple rotary inverted pendulum (RIP) devices. The results show that the proposed architecture facilitates the learning process and if more agents participate the learning speed can be improved. In addition, Lim *et al.*[106] uses FRL architecture based on a multi-agent environment to solve the problems and limitations of RL for applications to the real-world problems. The proposed federation policy allows multiple agents to share their learning experiences to get better learning efficacy. The proposed scheme adopts Actor-Critic PPO algorithm for four types of RL simulation environments from OpenAI Gym as well as RIP in real control systems. Compared to a previous real-environment study, the scheme enhances learning performance by approximately 1.2 times.

5.4. FRL for attack detection

With the heterogeneity of services and the sophistication of threats, it is challenging to detect these attacks using traditional methods or centralized ML-based methods, which have a high false alarm rate and do not take privacy into account. FRL offers a powerful alternative to detecting attacks and provides support for network defense in different scenarios.

Because of various constraints, IoT applications have become a primary target for malicious adversaries that can disrupt normal operations or steal confidential information. In order to address the security issues in flying ad-hoc network (FANET), Mowla *et al.*[107] proposes an adaptive FRL-based jamming attack defense strategy for unmanned aerial vehicles (UAVs). A model-free Q-learning mechanism is developed and deployed on distributed UAVs to cooperatively learn detection models for jamming attacks. According to the results, the average accuracy of the federated jamming detection mechanism, employed in the proposed defense strategy, is 39.9% higher than the distributed mechanism when verified with the CRAWDAD standard and the ns-3 simulated FANET jamming attack dataset.

An efficient traffic monitoring framework, known as DeepMonitor, is presented in the study of Nguyen *et al.*[108] to provide fine-grained traffic analysis capability at the edge of software defined network (SDN) based IoT networks. The agents deployed in edge nodes consider the different granularity-level requirements and their maximum flow-table capacity to achieve the optimal flow rule match-field strategy. The control optimization problem is formulated as the MDP and a federated DDQN algorithm is developed to improve the learning performance of agents. The results show that the proposed monitoring framework can produce reliable traffic granularity at all levels of traffic granularity and substantially mitigate the issue of flow-table overflows. In addition, the distributed denial of service (DDoS) attack detection performance of an intrusion detection system can be enhanced by up to 22.83% by using DeepMonitor instead of FlowStat.

In order to reduce manufacturing costs and improve production efficiency, the industrial internet of things (IIoT) is proposed as a potentially promising research direction. It is a challenge to implement anomaly detection mechanisms in IIoT applications with data privacy protection. Wang *et al.*[109] proposes a reliable anomaly detection strategy for IIoT using FRL techniques. In the system framework, there are four entities involved in establishing the detection model, *i.e.*, the Global Anomaly Detection Center (GADC), the Local Anomaly Detection Center (LADC), the Regional Anomaly Detection Center (RADC), and the users. The anomaly detection is suggested to be implemented in two phases, including anomaly detection for RADC and users. Especially, the GADC can build global RADC anomaly detection models based on local models trained by LADCs. Different from RADC anomaly detection based on action deviations, user anomaly detection is

mainly concerned with privacy leakage and is employed by RADC and GADC. Note that the DDPG algorithm is applied for local anomaly detection model training.

5.5. FRL for other applications

Due to the outstanding performance of training efficiency and privacy protection, many researchers are exploring the possible applications of FRL.

FL has been applied to realize distributed energy management in IoT applications. In the revolution of smart home, smart meters are deployed in the advanced metering infrastructure (AMI) to monitor and analyze the energy consumption of users in real-time. As an example[110], the FRL-based approach is proposed for the energy management of multiple smart homes with solar PVs, home appliances, and energy storage. Multiple local home energy management systems (LHEMSs) and a global server (GS) make up FRL architecture of the smart home. DRL agents for LHEMSs construct and upload local models to the GS by using energy consumption data. The GS updates the global model based on local models of LHEMSs using the federated stochastic gradient descent (FedSGD) algorithm. Under heterogeneous home environments, simulation results indicate that the proposed approach outperforms others when it comes to convergence speed, appliance energy consumption, and the number of agents.

Moreover, FRL offers an alternative to share information with low latency and privacy preservation. The collaborative perception of vehicles provided by IoV can greatly enhance the ability to sense things beyond their line of sight, which is important for autonomous driving. Region quadtrees have been proposed as a storage and communication resource-saving solution for sharing perception information[111]. It is challenging to tailor the number and resolution of transmitted quadtree blocks to bandwidth availability. In the framework of FRL, Mohamed *et al.*[112] presents a quadtree-based point cloud compression mechanism to select cooperative perception messages. Specifically, over a period of time, each vehicle covered by an RSU transfers its latest network weights with the RSU, which then averages all of the received model parameters and broadcasts the result back to the vehicles. Optimal sensory information transmission (*i.e.*, quadtree blocks) and appropriate resolution levels for a given vehicle pair are the main objectives of a vehicle. The dueling and branching concepts are also applied to overcome the vast action space inherent in the formulation of the RL problem. Simulation results show that the learned policies achieve higher vehicular satisfaction and the training process is enhanced by FRL.

5.6. Lessons Learned

In the following, we summarize the major lessons learned from this survey in order to provide a comprehensive understanding of current research on FRL applications.

5.6.1. Lessons learned from the aggregation algorithms

The existing FRL literature usually uses classical DRL algorithms, such as DQN and DDPG, at the participants, while the gradients or parameters of the critic and/or actor networks are periodically reported synchronously or asynchronously by the participants to the coordinator. The coordinator then aggregates the parameters or gradients and sends the updated values to the participants. In order to meet the challenges presented by different scenarios, the aggregation algorithms have been designed as a key feature of FRL. In the original FedAvg algorithm[12], the number of samples in a participant's dataset determines its influence on the global model. In accordance with this idea, several papers propose different methods to calculate the weights in the aggregation algorithms according to the requirement of application. In the study from Lim *et al.*[106], the aggregation weight is derived from the average of the cumulative rewards of the last ten episodes. Greater weights are placed on the models of those participants with higher rewards. In contrast to the positive correlation of reward, Huang *et al.*[96] takes the error rate of action as an essential factor to assign weights for participating in the global model training. In D2D -assisted edge caching, Wang *et al.*[89] uses the reward and some

device-related indicators as the measurement to evaluate the local model's contribution to the global model. Moreover, the existing FRL methods based on offline DRL algorithms, such DQN and DDPG, usually use experience replay. Sampling random batch from replay memory can break correlations of continuous transition tuples and accelerate the training process. To arrive at an accurate evaluation of the participants, the paper [102] calculates the aggregation weight based on the size of the training batch in each iteration.

The above aggregation methods can effectively deal with the issue of data imbalance and performance discrepancy between participants, but it is hard for participants to cope with subtle environmental differences. According to the paper [105], as soon as a participant reaches the predefined criteria in its own environment, it should stop learning and send its model parameters as a reference to the remaining individuals. Exchanging mature network models (satisfying terminal conditions) can help other participants complete their training quickly. Participants in other similar environments can continue to use FRL for further updating their parameters to achieve the desired model performance according to their individual environments. Liu *et al.* [57] also suggests that the sharing global model in the cloud is not the final policy model for local participants. An effective transfer learning should be applied to resolve the structural difference between the shared network and private network.

5.6.2. Lessons learned from the relationship between FL and RL

In most of the literature on FRL, FL is used to improve the performance of RL. With FL, the learning experience can be shared among decentralized multiple parties while ensuring privacy and scalability without requiring direct data offloading to servers or third parties. Therefore, FL can expand the scope and enhance the security of RL. Among the applications of FRL, most researchers focus on the communication network system due to its robust security requirements, advanced distributed architecture, and a variety of decision-making tasks. Data offloading [93] and caching [89] solutions powered by distributed AI are available from FRL. In addition, with the ability to detect a wide range of attacks and support defense solutions, FRL has emerged as a strong alternative for performing distributed learning for security-sensitive scenarios. Enabled by the privacy-enhancing and cooperative features, detection and defense solutions can be learned quickly where multiple participants join to build a federated model [107,109]. FRL can also provide viable solutions to realize intelligence for control systems with many applied domains such as robotics [57] and autonomous driving [104] without data exchange and privacy leakage. The data owners (robot or vehicle) may not trust the third-party server and therefore hesitate to upload their private information to potentially insecure learning systems. Each participant of FRL runs a separate RL model for determining its own control policy and gains experience by sharing model parameters, gradients or losses.

Meanwhile, RL may have the potential to optimize FL schemes and improve the efficiency of training. Due to the unstable network connectivity, it is not practical for FL to update and aggregate models simultaneously across all participants. Therefore, Wang *et al.* [113] proposes a RL-based control framework that intelligently chooses the participants to participate in each round of FL with the aim to speed up convergence. Similarly, Zhang *et al.* [114] applies RL to pre-select a set of candidate edge participants, and then determine reliable edge participants through social attribute perception. In IoT or IoV scenarios, due to the heterogeneous nature of participating devices, different computing and communication resources are available to them. RL can speed up training by coordinating the allocation of resources between participants. Zhan *et al.* [115] defines the L4L (Learning for Learning) concept, *i.e.*, use RL to improve FL. Using the heterogeneity of participants and dynamic network connections, this paper investigates a computational resource control problem for FL that simultaneously considers learning time and energy efficiency. An experience-driven resource control approach based on RL is presented to derive the near-optimal strategy with only the participants' bandwidth information in the previous training rounds. In addition, as with any other ML algorithm, FL algorithms are vulnerable to malicious attacks. RL has been studied to defend against attacks in various scenarios, and it can also enhance the security of FL. The paper [116] proposes a reputation-aware RL (RA-RL) based selection

method to ensure that FL is not disrupted. The participating devices' attributes, including computing resources and trust values, *etc*, are used as part of the environment in RL. In the aggregation of the global model, devices with high reputation levels will have a greater chance of being considered to reduce the effects of malicious devices mixed into FL.

5.6.3. Lessons learned from categories of FRL

As discussed above, FRL can be divided into two main categories, *i.e*., HFRL and VFRL. Currently, most of the existing research is focused on HFRL, while little attention is devoted to VFRL. The reason for this is that HFRL has obvious application scenarios, where multiple participants have similar decision-making tasks with individual environments, such as caching allocation[59], offloading optimization[58], and attack monitoring[108]. The participants and coordinator only need to train a similar model with the same state and action spaces. Consequently, the algorithm design can be implemented and the training convergence can be verified relatively easily. On the other hand, even though VFRL has a higher degree of technical difficulty at the algorithm design level, it also has a wide range of possible applications. In a multi-agent scenario, for example, a single agent is limited by its ability to observe only part of the environment, whereas the transition of the environment is determined by the behavior of all the agents. Zhuo *et al*.[65] assumes agents cannot share their partial observations of the environment and some agents are unable to receive rewards. The federated Q-network aggregation algorithm between two agents is proposed for VFRL. The paper[97] specifically applies both HFRL and VFRL for radio access network slicing. For the same type of services, similar data samples are trained locally at participating devices, and BSs perform horizontal aggregation to integrate a cooperative access model by adopting an iterative approach. The terminal device also can optimize the selection of base stations and network slices based on the global model of VFRL, which aggregates access features generated by different types of services on the third encrypted party. The method improves the device's ability to select the appropriate access points when initiating different types of service requests under restrictions regarding privacy protection. The feasible implementation of VFRL also provides guidance for future research.

6. OPEN ISSUES AND FUTURE RESEARCH DIRECTIONS

As we presented in the previous section, FRL serves an increasingly important role as an enabler of various applications. While the FRL-based approach possesses many advantages, there are a number of critical open issues to consider for future implementation. Therefore, this section focuses on several key challenges, including those inherited from FL such as security and communication issues, as well as those unique to FRL. Research on tackling these issues offers interesting directions for the future.

6.1. Learning convergence in HFRL

In realistic HFRL scenarios, while the agents perform similar tasks, the inherent dynamics for the different environments in which the agents reside are usually not exactly identically distributed. The slight difference in the stochastic properties of the transition models for multiple agents could cause the learning convergence issue. One possible method to address this problem is by adjusting the frequency of global aggregation, *i.e.*, after each global aggregation, a period of time is left for each agent to fine-tune its local parameters according to its own environment. Apart from the non-identical environment problem, another interesting and important problem is how to leverage FL to make RL algorithms converge better and faster. It is well-known that DRL algorithms could be unstable and diverge, especially when off-policy training is combined with function approximation and bootstrapping. In FRL, the training curves of some agents could diverge while others converge although the agents are trained in the exact replicas of the same environment. By leveraging FL, it is envisioned that we could expedite the training process as well as increase the stability. For example, we could selectively aggregate the parameters of a subset of agents with a larger potential for convergence, and later transfer the converged parameters to all the agents. To tackle the above problems, several possible solutions proposed for FL algorithm contains certain reference significance. For example, server operators could account for heterogeneity inherent

in partial information by adding a proximal term[117]. The local updates submitted by agents are constrained by the tunable term and have a different effect on the global parameters. In addition, a probabilistic agent selection scheme can be implemented to select the agents whose local FL models have significant effects on the global model to minimize the FL convergence time and the FL training loss[118]. Another problem is theoretical analysis of the convergence bounds. Although some existing studies have been directed at this problem[119], the convergence can be guaranteed since the loss function is convex. How to analyze and evaluate the non-convex loss functions in HFRL is also an important research topic in the future.

6.2. Agents without rewards in VFRL

In most existing works, all the RL agents have the ability to take part in full interaction with the environment and can generate their own actions and rewards. Even though some MARL agents may not participate in the policy decision, they still generate their own reward for evaluation. In some scenarios, special agents in VFRL take the role of providing assistance to other agents. They can only observe the environment and pass on the knowledge of their observation, so as to help other agents make more effective decisions. Therefore, such agents do not have their own actions and rewards. The traditional RL models cannot effectively deal with this thorny problem. Many algorithms either directly use the states of such agents as public knowledge in the system model or design corresponding action and reward for such agents, which may be only for convenience of calculation and have no practical significance. These approaches cannot fundamentally overcome the challenge, especially when privacy protection is also an essential objective to be complied with. Although the FedRL algorithm[65] is proposed to deal with the above problem, which has demonstrated good performance, there are still some limitations. First of all, the number of agents used in experiments and algorithms is limited to two, which means the scalability of this algorithm is not high and VFRL algorithms for a large number of agents need to be designed. Secondly, this algorithm uses Q-network as the federated model, which is a relatively simple algorithm. Therefore, how to design VFRL models based on other more complex and changeable networks remains an open issue.

6.3. Communications

In FRL, the agents need to exchange the model parameters, gradients, intermediate results, etc., between themselves or with a central server. Due to the limited communication resources and battery capacity, the communication cost is an important consideration when implementing these applications. With an increased number of participants, the coordinator has to bear more network workload within the client-server FRL model[120]. This is because each participant needs to upload and download model updates through the coordinator. Although the distributed peer-to-peer model does not require a central coordinator, each agent may have to exchange information with other participants more frequently. In current research for distributed models, there are no effective model exchange protocols to determine when to share experiences with which agents. In addition, DRL involves updating parameters in deep neural networks. Several popular DRL algorithms, such as DQN[121] and DDPG[122], consist of multiple layers or multiple networks. Model updates contain millions of parameters, which isn't feasible for scenarios with limited communication resources. The research directions for the above issues can be divided into three categories. First, it is necessary to design a dynamic update mechanism for participants to optimize the number of model exchanges. A second research direction is to use model compression algorithms to reduce the amount of communication data. Finally, aggregation algorithms that allow participants to only submit the important parts of local model should be studied further.

6.4. Privacy and Security

Although FL provides privacy protection that allows the agents to exchange information in a secure manner during the learning process, it still has several privacy and security vulnerabilities associated with communication and attack[123]. As FRL is implemented based on FL algorithms, these problems also exist in FRL in the same or variant form. It is important to note that the data poisoning attack is a different attack mode between FL and FRL. In the existing classification tasks of FL, each piece of training data in the dataset corresponds to

a respective label. The attacker flips the labels on training examples in one category onto another while the features of the examples are kept unchanged, misguiding the establishment of a target model[124]. However, in the decision-making task of FRL, the training data is continuously generated from the interaction between the agent and the environment. As a result, the data poisoning attack is implemented in another way. For example, the malicious agent tampers with the reward, which causes the evaluative function to shift. An option is to conduct regular safety assessments for all participants. Participants whose evaluation indicator falls below the threshold are punished to reduce the impact on the global model[125]. Apart form the insider attacks which are launched by the agents in the FRL system, there may be various outsider attacks which are launched by intruders or eavesdroppers. Intruders may hide in the environment where the agent is and manipulate the transitions of environment to achieve specific goals. In addition, by listening to the communication between the coordinator and the agent, the eavesdropper may infer sensitive information from exchanging parameters and gradients[126]. Therefore, the development of technology that detects and protects against attacks and privacy threats does have great potential and is urgently needed.

6.5. Join and exit mechanisms design

One overlooked aspect of FRL-based research is the join and exit process of participants. In practice, the management of participants is essential to the normal progression of cooperation. As mentioned earlier in the security issue, the penetration of malicious participants severely impacts the performance of the cooperative model and the speed of training. The joining mechanism provides participants with the legal status to engage in federated cooperation. It is the first line of defense against malicious attackers. In contrast, the exit mechanism signifies the cancellation of the permission for cooperation. Participant-driven or enforced exit mechanisms are both possible. In particular, for synchronous algorithms, ignoring the exit mechanism can negatively impact learning efficiency. This is because the coordinator needs to wait for all participants to submit their information. In the event that any participant is offline or compromised and unable to upload, the time for one round of training will be increased indefinitely. To address the bottleneck, a few studies consider updating the global model using the selected models from a subset of participants[113,127]. Unfortunately, there is no comprehensive consideration of the exit mechanism, and the communication of participants is typically assumed to be reliable. Therefore, research gaps of FRL still exist in joining and exiting mechanisms. It is expected that the coordinator or monitoring system, upon discovering a failure, disconnection, or malicious participant, will use the exit mechanism to reduce its impact on the global model or even eliminate it.

6.6. Incentive mechanisms

For most studies, the agents taking part in the FRL process are assumed to be honest and voluntary. Each agent provides assistance for the establishment of the cooperation model following the rules and freely shares the masked experience through encrypted parameters or gradients. An agent's motivation for participation may come from regulation or incentive mechanisms. The FRL process within an organization is usually governed by regulations. For example, BSs belonging to the same company establish a joint model for offloading and caching. Nevertheless, because participants may be members of different organizations or use disparate equipment, it is difficult for regulation to force all parties to share information learned from their own data in the same manner. If there are no regulatory measures, participants prone to selfish behavior will only benefit from the cooperation model but not submit local updates. Therefore, the cooperation of multiple parties, organizations, or individuals requires a fair and efficient incentive mechanism to encourage their active participation. In this way, agents providing more contributions can benefit more and selfish agents unwilling to share there learning experience will receive less benefit. As an example, Google Keyboard[128] users can choose whether or not to allow Google to use their data, but if they do, they can benefit from more accurate word prediction. Although an incentive mechanism in a context-aware manner among data owners is proposed in the study from Yu *et al*.[129], it is not suitable for the RL problems. There is still no clear plan of action regarding how the FRL-based application can be designed to create a reasonable incentive mechanism for inspiring agents to participate in collaborative learning. To be successful, future research needs to propose a quantitative standard

for evaluating the contribution of agents in FRL.

6.7. Peer-to-peer cooperation

FRL applications have the option of choosing between a central server-client model as well as a distributed peer-to-peer model. A distributed model can not only eliminate the single point of failure, but it can also improve energy efficiency significantly by allowing models to be exchanged directly between two agents. In a typical application, two adjacent cars share experience learned from road condition environment in the form of models with D2D communications to assist autonomous driving. However, the distributed cooperation increases the complexity of the learning system and imposes stricter requirements for application scenarios. This research should include, but not be limited to, the agent selection method for the exchange model, the mechanism for triggering the model exchange, the improvement of algorithm adaptability, and the convergence analysis of the aggregation algorithm.

7. CONCLUSION

As a new and potential branch of RL, FL can make learning safer and more efficient while leveraging the benefits of FL. We have discussed the basic definitions of FL and RL as well as our thoughts on their integration in this paper. In general, FRL algorithms can be classified into two categories, *i.e.*, HFRL and VFRL. Thus, the definition and general framework of these two categories have been given. Specifically, we have highlighted the difference between HFRL and VFRL. Then, a lot of existing FRL schemes have been summarized and analyzed according to different applications. Finally, the potential challenges in the development of FRL algorithms have been explored. Several open issues of FRL have been identified, which will encourage more efforts toward further research in FRL.

DECLARATIONS

Authors' contributions

Made substantial contributions to the research and investigation process, reviewed and summarized the literature, wrote and edited the original draft: Qi J, Zhou Q

Performed oversight and leadership responsibility for the research activity planning and execution, as well as developed ideas and evolution of overarching research aims: Lei L

Performed critical review, commentary and revision, as well as provided administrative, technical, and material support: Zheng K

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Discovery Grant No. 401718) and the CARE-AI Seed Fund at the University of Guelph.

Conflicts of interest

The authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Nair A, Srinivasan P, Blackwell S, et al. Massively parallel methods for deep reinforcement learning. CoRR 2015;abs/1507.04296. Available from: http://arxiv.org/abs/1507.04296.
2. Grounds M, Kudenko D. Parallel reinforcement learning with linear function approximation. In: Tuyls K, Nowe A, Guessoum Z, Kudenko D, editors. Adaptive agents and multi-agent systems III. Adaptation and multi-agent learning. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. pp. 60–74.
3. Clemente AV, Martínez HNC, Chandra A. Efficient parallel methods for deep reinforcement learning. CoRR 2017;abs/1705.04862. Available from: http://arxiv.org/abs/1705.04862.
4. Lim WYB, Luong NC, Hoang DT, et al. Federated larning in mobile edge networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials* 2020;22:2031–63.
5. Nguyen DC, Ding M, Pathirana PN, et al. Federated learning for internet of things: a comprehensive survey. *IEEE Communications Surveys Tutorials* 2021;23:1622–58.
6. Khan LU, Saad W, Han Z, Hossain E, Hong CS. Federated learning for internet of things: recent advances, taxonomy, and open challenges. *IEEE Communications Surveys Tutorials* 2021;23:1759–99.
7. Yang Q, Liu Y, Cheng Y, et al. 1st ed. Morgan & Claypool; 2019.
8. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. *ACM T Intel Syst Tec* 2019;10:1–19.
9. Qinbin L, Zeyi W, Bingsheng H. Federated learning systems: vision, hype and reality for data privacy and protection. CoRR 2019;abs/1907.09693. Available from: http://arxiv.org/abs/1907.09693.
10. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. *IEEE Signal Process Mag* 2020;37:50–60.
11. Wang S, Tuor T, Salonidis T, Leung KK, et al. Adaptive federated learning in resource constrained edge computing systems. *IEEE J Sel Area Comm* 2019;37:1205–21.
12. McMahan HB, Moore E, Ramage D, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. CoRR 2016;abs/1602.05629. Available from: http://arxiv.org/abs/1602.05629.
13. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE T Knowl Date En* 2018;13:1333–45.
14. Zhu H, Jin Y. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems* 2020;31:1310–22.
15. Kairouz P, McMahan HB, Avent B, et al. Advances and open problems in federated learning. CoRR 2019;abs/1912.04977. Available from: http://arxiv.org/abs/1912.04977.
16. Pan SJ, Yang Q. A survey on transfer learning. *IEEE T Knowl Date En* 2010;22:1345–59.
17. Li Y. Deep reinforcement learning: an overview. CoRR 2017;abs/1701.07274. Available from: http://arxiv.org/abs/1701.07274.
18. Xu Z, Tang J, Meng J, et al. Experience-driven networking: a deep reinforcement learning based approach. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE; 2018. pp. 1871–79.
19. Mohammadi M, Al-Fuqaha A, Guizani M, Oh JS. Semisupervised deep reinforcement learning in support of IoT and smart city services. IEEE Internet of Things Journal 2018;5:624–35. [DOI: 10.1109/JIOT.2017.2712560]
20. Bu F, Wang X. A smart agriculture IoT system based on deep reinforcement learning. *Future Generation Computer Systems* 2019;99:500–507. Available from: https://www.sciencedirect.com/science/article/pii/S0167739X19307277.
21. Xiong X, Zheng K, Lei L, Hou L. Resource allocation based on deep reinforcement learning in IoT edge computing. *IEEE J Sel Area Comm* 2020;38:1133–46.
22. Lei L, Qi J, Zheng K. Patent analytics based on feature vector space model: a case of IoT. *IEEE Access* 2019;7:45705–15.
23. Shalev-Shwartz S, Shammah S, Shashua A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. CoRR 2016;abs/1610.03295. Available from: http://arxiv.org/abs/1610.03295.
24. Sallab AE, Abdou M, Perot E, Yogamani S. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*

2017;2017:70–76.
25. Taylor ME. Teaching reinforcement learning with mario: an argument and case study. In: Second AAAI Symposium on Educational Advances in Artificial Intelligence; 2011. Available from: https://www.aaai.org/ocs/index.php/EAAI/EAAI11/paper/viewPaper/3515.
26. Holcomb SD, Porter WK, Ault SV, Mao G, Wang J. Overview on deepmind and its alphago zero ai. In: Proceedings of the 2018 international conference on big data and education; 2018. pp. 67–71.
27. Watkins CJ, Dayan P. Q-learning. *Mach Learn* 1992;8:279–92. Available from: https://link.springer.com/content/pdf/10.1007/BF00992698.pdf.
28. Thorpe TL. Vehicle traffic light control using sarsa. In: Online]. Available: citeseer. ist. psu. edu/thorpe97vehicle. html. Citeseer; 1997. Available from: https://citeseer.ist.psu.edu/thorpe97vehicle.html.
29. Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: Xing EP, Jebara T, editors. Proceedings of the 31st International Conference on Machine Learning. vol. 32 of Proceedings of Machine Learning Research. Bejing, China: PMLR; 2014. pp. 387–95. Available from: https://proceedings.mlr.press/v32/silver14.html.
30. Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 1992;8:229–56.
31. Konda VR, Tsitsiklis JN. Actor-critic algorithms. In: advances in neural information processing systems; 2000. pp. 1008–14. Available from: https://proceedings.neurips.cc/paper/1786-actor-critic-algorithms.pdf.
32. Henderson P, Islam R, Bachman P, et al. Deep reinforcement learning that matters. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32; 2018. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/11694.
33. Lei L, Tan Y, Dahlenburg G, Xiang W, Zheng K. Dynamic energy dispatch based on deep reinforcement learning in IoT-Driven smart isolated microgrids. *IEEE Internet Things* 2021;8:7938–53.
34. Lei L, Xu H, Xiong X, Zheng K, Xiang W, et al. Multiuser resource control with deep reinforcement learning in IoT edge computing. *IEEE Internet Things* 2019;6:10119–33.
35. Ohnishi S, Uchibe E, Yamaguchi Y, et al. Constrained deep q-learning gradually approaching ordinary q-learning. *Front Neurorobotics* 2019;13:103.
36. Peng J, Williams RJ. Incremental multi-step Q-learning. In: machine learning proceedings 1994. Elsevier; 1994. pp. 226–32.
37. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015;518:529–33.
38. Lei L, Tan Y, Zheng K, et al. Deep reinforcement learning for autonomous internet of things: model, applications and challenges. *IEEE Communications Surveys Tutorials* 2020;22:1722–60.
39. Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30; 2016. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/10295.
40. Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. arXiv preprint arXiv:151105952 2015. Available from: https://arxiv.org/abs/1511.05952.
41. Gu S, Lillicrap TP, Ghahramani Z, Turner RE, Levine S. Q-Prop: sample-efficient policy gradient with an off-policy critic. CoRR 2016;abs/1611.02247. Available from: http://arxiv.org/abs/1611.02247.
42. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochas- tic actor. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. pp. 1861–70. Available from: https://proceedings.mlr.press/v80/haarnoja18b.html.
43. Mnih V, Badia AP, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: Balcan MF, Weinberger KQ, editors. Proceedings of The 33rd International Conference on Machine Learning. vol. 48 of Proceedings of Machine Learning Research. New York, New York, USA: PMLR; 2016. pp. 1928–37. Available from: https://proceedings.mlr.press/v48/mniha1 6.html.
44. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv:150902971 2015. Available from: https://arxiv.org/abs/1509.02971.
45. Barth-Maron G, Hoffman MW, Budden D, et al. Distributed distributional deterministic policy gradients. CoRR 2018;abs/1804.08617. Available from: http://arxiv.org/abs/1804.08617.
46. Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. pp. 1587–96. Available from: https://proceedings.mlr.press/v80/fujimoto18a.html.
47. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: Bach F, Blei D, editors. Proceedings of the 32nd International Conference on Machine Learning. vol. 37 of Proceedings of Machine Learning Research. Lille, France: PMLR; 2015. pp. 1889–97. Available from: https://proceedings.mlr.press/v37/schulman15.html.
48. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. CoRR 2017;abs/1707.06347. Available from: http://arxiv.org/abs/1707.06347.
49. Zhu P, Li X, Poupart P. On improving deep reinforcement learning for POMDPs. CoRR 2017;abs/1704.07978. Available from: http://arxiv.org/abs/1704.07978.
50. Hausknecht M, Stone P. Deep recurrent q-learning for partially observable mdps. In: 2015 aaai fall symposium series; 2015. Available from: https://www.aaai.org/ocs/index.php/FSS/FSS15/paper/viewPaper/11673.
51. Heess N, Hunt JJ, Lillicrap TP, Silver D. Memory-based control with recurrent neural networks. CoRR 2015;abs/1512.04455. Available

from: http://arxiv.org/abs/1512.04455.

52. Foerster J, Nardelli N, Farquhar G, et al. Stabilising experience replay for deep multi-agent reinforcement learning. In: Precup D, Teh YW, editors. Proceedings of the 34th International Conference on Machine Learning. vol. 70 of Proceedings of Machine Learning Research. PMLR; 2017. pp. 1146–55. Available from: https://proceedings.mlr.press/v70/foerster17b.html.
53. Van der Pol E, Oliehoek FA. Coordinated deep reinforcement learners for traffic light control. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016) 2016. Available from: https://www.elisevanderpol.nl/papers/vanderpolNIPSMALIC2016.pdf.
54. Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32; 2018. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/11794.
55. Lowe R, Wu Y, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. CoRR 2017;abs/1706.02275. Available from: http://arxiv.org/abs/1706.02275.
56. Nadiger C, Kumar A, Abdelhak S. Federated reinforcement learning for fast personalization. In: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE); 2019. pp. 123–27.
57. Liu B, Wang L, Liu M, Xu C. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. CoRR 2019;abs/1901.06455. Available from: http://arxiv.org/abs/1901.06455.
58. Ren J, Wang H, Hou T, Zheng S, Tang C. Federated learning-based computation offloading optimization in edge computing-supported internet of things. *IEEE Access* 2019;7:69194–201.
59. Wang X, Wang C, Li X, Leung VCM, Taleb T. Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching. *IEEE Internet Things* 2020;7:9441–55.
60. Chen J, Monga R, Bengio S, Józefowicz R. Revisiting Distributed Synchronous SGD. CoRR 2016;abs/1604.00981. Available from: http://arxiv.org/abs/1604.00981.
61. Mnih V, Badia AP, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: Balcan MF, Weinberger KQ, editors. Proceedings of The 33rd International Conference on Machine Learning. vol. 48 of Proceedings of Machine Learning Research. New York, New York, USA: PMLR; 2016. pp. 1928–37. Available from: https://proceedings.mlr.press/v48/mniha16.html.
62. Espeholt L, Soyer H, Munos R, et al. IMPALA: scalable distributed deep-RL with importance weighted actor- learner architectures. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. pp. 1407–16. Available from: http://proceedings.mlr.press/v80/espeholt18a.html.
63. Horgan D, Quan J, Budden D, et al. Distributed prioritized experience replay. CoRR 2018;abs/1803.00933. Available from: http://arxiv.org/abs/1803.00933.
64. Liu T, Tian B, Ai Y, et al. Parallel reinforcement learning: a framework and case study. *IEEE/CAA Journal of Automatica Sinica* 2018;5:827–35.
65. Zhuo HH, Feng W, Xu Q, Yang Q, Lin Y. Federated reinforcement learning. CoRR 2019;abs/1901.08277. Available from: http://arxiv.org/abs/1901.08277.
66. Canese L, Cardarilli GC, Di Nunzio L, et al. Multi-agent reinforcement learning: a review of challenges and applications. *Applied Sciences* 2021;11:4948. Available from: https://doi.org/10.3390/app11114948.
67. Busoniu L, Babuska R, De Schutter B. A comprehensive survey of Multiagent Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 2008;38:156–72.
68. Zhang K, Yang Z, Başar T. Multi-agent reinforcement learning: a selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control* 2021:321–84.
69. Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective. *Auton Robot* 2000;8:345–83.
70. Szepesvári C, Littman ML. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural Comput* 1999;11:2017–60.
71. Littman ML. Value-function reinforcement learning in Markov games. *Cogn Syst Res* 2001;2:55–66.
72. Tan M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the tenth international conference on machine learning; 1993. pp. 330–37.
73. Lauer M, Riedmiller M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In: In Proceedings of the Seventeenth International Conference on Machine Learning. Citeseer; 2000. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary.
74. Monahan GE. State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms. *Manage Sci* 1982;28:1–16.
75. Oroojlooyjadid A, Hajinezhad D. A review of cooperative multi-agent deep reinforcement learning. CoRR 2019;abs/1908.03963. Available from: http://arxiv.org/abs/1908.03963.
76. Bernstein DS, Givan R, Immerman N, Zilberstein S. The complexity of decentralized control of Markov decision processes. *Math Oper Res* 2002;27:819–40.
77. Omidshafiei S, Pazis J, Amato C, How JP, Vian J. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In: Precup D, Teh YW, editors. Proceedings of the 34th International Conference on Machine Learning. vol. 70 of

Proceedings of Machine Learning Research. PMLR; 2017. pp. 2681–90. Available from: https://proceedings.mlr.press/v70/omidshafiei17a.html.

78. Han Y, Gmytrasiewicz P. Ipomdp-net: a deep neural network for partially observable multi-agent planning using interactive pomdps. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp. 6062–69.
79. Karkus P, Hsu D, Lee WS. QMDP-Net: deep learning for planning under partial observability; 2017. Available from: https://arxiv.org/abs/1703.06692.
80. Mao W, Zhang K, Miehling E, Başar T. Information state embedding in partially observable cooperative multi-agent reinforcement learning. In: 2020 59th IEEE Conference on Decision and Control (CDC); 2020. pp. 6124–31.
81. Mao H, Zhang Z, Xiao Z, Gong Z. Modelling the dynamic joint policy of teammates with attention multi-agent DDPG. CoRR 2018;abs/1811.07029. Available from: http://arxiv.org/abs/1811.07029.
82. Lee HR, Lee T. Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response. *Eur J Oper Res* 2021;291:296–308.
83. Sukhbaatar S, szlam a, Fergus R. Learning multiagent communication with backpropagation. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc.; 2016. Available from: https://proceedings.neurips.cc/paper/2016/file/55b1927fdafef39c48e5b73b5d61ea60-Paper.pdf.
84. Foerster JN, Assael YM, de Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. CoRR 2016;abs/1605.06676. Available from: http://arxiv.org/abs/1605.06676.
85. Buşoniu L, Babuška R, De Schutter B. Multi-agent reinforcement learning: an overview. *Innovations in multiagent systems and applications1* 2010:183–221.
86. Hu Y, Hua Y, Liu W, Zhu J. Reward shaping based federated reinforcement learning. *IEEE Access* 2021;9:67259–67.
87. Anwar A, Raychowdhury A. Multi-task federated reinforcement learning with adversaries. CoRR 2021;abs/2103.06473. Available from: https://arxiv.org/abs/2103.06473.
88. Wang X, Han Y, Wang C, et al. In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network* 2019;33:156–65.
89. Wang X, Li R, Wang C, et al. Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. *IEEE J Sel Area Comm* 2021;39:154–69.
90. Zhang M, Jiang Y, Zheng FC, Bennis M, You X. Cooperative edge caching via federated deep reinforcement learning in fog-RANs. In: 2021 IEEE International Conference on Communications Workshops (ICC Workshops); 2021. pp. 1–6.
91. Majidi F, Khayyambashi MR, Barekatain B. HFDRL: an intelligent dynamic cooperate cashing method based on hierarchical federated deep reinforcement learning in edge-enabled IoT. *IEEE Internet Things* 2021:1-1.
92. Zhao L, Ran Y, Wang H, Wang J, Luo J. Towards cooperative caching for vehicular networks with multi-level federated reinforcement learning. In: ICC 2021 - IEEE International Conference on Communications; 2021. pp. 1–6.
93. Zhu Z, Wan S, Fan P, Letaief KB. Federated multi-agent actor-critic learning for age sensitive mobile edge computing. *IEEE Internet Things* 2021:1-1.
94. Yu S, Chen X, Zhou Z, Gong X, Wu D. When deep reinforcement learning meets federated learning: intelligent multi-timescale resource management for multi-access edge computing in 5G ultra dense network. arXiv:200910601 [cs] 2020 Sep. ArXiv: 2009.10601. Available from: http://arxiv.org/abs/2009.10601.
95. Tianqing Z, Zhou W, Ye D, Cheng Z, Li J. Resource allocation in IoT edge computing via concurrent federated reinforcement learning. *IEEE Internet Things* 2021:1–1.
96. Huang H, Zeng C, Zhao Y, et al. Scalable orchestration of service function chains in NFV-Enabled networks: a federated reinforcement learning approach. *IEEE J Sel Area Comm* 2021;39:2558–71.
97. Liu YJ, Feng G, Sun Y, Qin S, Liang YC. Device association for RAN slicing based on hybrid federated deep reinforcement learning. *IEEE T Veh Technol* 2020;69:15731–45.
98. Wang G, Dang CX, Zhou Z. Measure contribution of participants in federated learning. In: 2019 IEEE International Conference on Big Data (Big Data); 2019. pp. 2597–604.
99. Cao Y, Lien SY, Liang YC, Chen KC. Federated deep reinforcement learning for user access control in open radio access networks. In: ICC 2021 - IEEE International Conference on Communications; 2021. pp. 1–6.
100. Zhang L, Yin H, Zhou Z, Roy S, Sun Y. Enhancing WiFi multiple access performance with federated deep reinforcement learning. In: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall); 2020. pp. 1–6.
101. Xu M, Peng J, Gupta BB, et al. Multi-agent federated reinforcement learning for secure incentive mechanism in intelligent cyber-physical systems. *IEEE Internet Things* 2021:1–1.
102. Zhang X, Peng M, Yan S, Sun Y. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. *IEEE Internet Things* 2020;7:6380–91.
103. Kwon D, Jeon J, Park S, Kim J, Cho S. Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks. *IEEE Internet Things* 2020;7:9895–903.

104. Liang X, Liu Y, Chen T, Liu M, Yang Q. Federated transfer reinforcement learning for autonomous driving. arXiv:191006001 [cs] 2019 Oct. ArXiv: 1910.06001. Available from: http://arxiv.org/abs/1910.06001.
105. Lim HK, Kim JB, Heo JS, Han YH. Federated reinforcement learning for training control policies on multiple IoT devices. Sensors 2020 Mar;20:1359. Available from: https://www.mdpi.com/1424-8220/20/5/1359.
106. Lim HK, Kim JB, Ullah I, Heo JS, Han YH. Federated reinforcement learning acceleration method for precise control of multiple devices. *IEEE Access* 2021;9:76296–306.
107. Mowla NI, Tran NH, Doh I, Chae K. AFRL: adaptive federated reinforcement learning for intelligent jamming defense in FANET. *Journal of Communications and Networks* 2020;22:244–58.
108. Nguyen TG, Phan TV, Hoang DT, Nguyen TN, So-In C. Federated deep reinforcement learning for traffic monitoring in SDN-Based IoT networks. *IEEE T Cogn Commun* 2021:1–1.
109. Wang X, Garg S, Lin H, et al. Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning. *IEEE Internet Things* 2021:1–1.
110. Lee S, Choi DH. Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources. *IEEE T Ind Inform* 2020:1–1.
111. Samet H. The quadtree and related hierarchical data structures. *ACM Comput Surv* 1984;16:187–260. Available from: https: // doi.org/10.1145/356924.356930.
112. Abdel-Aziz MK, Samarakoon S, Perfecto C, Bennis M. Cooperative perception in vehicular networks using multi-agent reinforcement learning. In: 2020 54th Asilomar Conference on Signals, Systems, and Computers; 2020. pp. 408–12.
113. Wang H, Kaplan Z, Niu D, Li B. Optimizing federated learning on non-IID data with reinforcement learning. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. Toronto, ON, Canada: IEEE; 2020. pp. 1698–707. Available from: https: //ieeexplore.ieee.org/document/9155494/.
114. Zhang P, Gan P, Aujla GS, Batth RS. Reinforcement learning for edge device selection using social attribute perception in industry 4.0. *IEEE Internet Things* 2021:1–1.
115. Zhan Y, Li P, Leijie W, Guo S. L4L: Experience-driven computational resource control in federated learning. *IEEE T Comput* 2021:1–1.
116. Dong Y, Gan P, Aujla GS, Zhang P. RA-RL: reputation-aware edge device selection method based on reinforcement learning. In: 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM); 2021. pp. 348–53.
117. Sahu AK, Li T, Sanjabi M, et al. On the convergence of federated optimization in heterogeneous networks. CoRR 2018;abs/ 1812.06127. Available from: http://arxiv.org/abs/1812.06127.
118. Chen M, Poor HV, Saad W, Cui S. Convergence time optimization for federated learning over wireless networks. *IEEE T Wirel Commun* 2021;20:2457–71.
119. Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of FedAvg on non-IID data; 2020. Available from: https://arxiv.org/abs/1907.02189?context=stat.ML.
120. Bonawitz KA, Eichner H, Grieskamp W, et al. Towards federated learning at scale: system design. CoRR 2019;abs/1902.01046. Available from: http://arxiv.org/abs/1902.01046.
121. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015;518:529–33. Available from: https://doi.org/10.1038/nature14236.
122. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning; 2019. Available from: https://arxiv.org/abs/1509.02971.
123. Lyu L, Yu H, Yang Q. Threats to federated learning: a survey. CoRR 2020;abs/2003.02133. Available from: https://arxiv.org/abs/2003.02133.
124. Fung C, Yoon CJM, Beschastnikh I. Mitigating sybils in federated learning poisoning. CoRR 2018;abs/1808.04866. Available from: http://arxiv.org/abs/1808.04866.
125. Anwar A, Raychowdhury A. Multi-task federated reinforcement learning with adversaries; 2021.
126. Zhu L, Liu Z, Han S. Deep leakage from gradients. CoRR 2019;abs/1906.08935. Available from: http://arxiv.org/abs/1906.08935.
127. Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC); 2019. pp. 1–7.
128. Yang T, Andrew G, Eichner H, et al. Applied federated learning: improving Google Keyboard query suggestions. CoRR 2018;abs/ 1812.02903. Available from: http://arxiv.org/abs/1812.02903.
129. Yu H, Liu Z, Liu Y, et al. A fairness-aware incentive scheme for federated learning. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. AIES '20. New York, NY, USA: Association for Computing Machinery; 2020. p. 393–399. Available from: https://doi.org/10.1145/3375627.3375840.

Li *et al. Intell Robot* 2021;1(1):58-83
DOI: 10.20517/ir.2021.08

Intelligence & Robotics

Review

Open Access

Bio-inspired intelligence with applications to robotics: a survey

Junfei Li, Zhe Xu, Danjie Zhu, Kevin Dong, Tao Yan, Zhu Zeng, Simon X. Yang

School of Engineering, University of Guelph, 50 Stone Road East, Guelph, ON N1G 2W1, Canada.

Correspondence to: Prof. Simon X. Yang, Advanced Robotics & Intelligent Systems (ARIS) Laboratory, School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada. E-mail: syang@uoguelph.ca

How to cite this article: Li J, Xu Z, Zhu D, Dong K, Yan T, Zeng Z, Yang SX. Bio-inspired intelligence with applications to robotics: a survey. *Intell Robot* 2021;1(1):58-83. http://dx.doi.org/10.20517/ir.2021.08

Received: 9 Sep 2021 **First Decision:** 20 Sep 2021 **Revised:** 26 Sep 2021 **Accepted:** 28 Sep 2021 **Published:** 12 Oct 2021

Academic Editor: Anmin Zhu **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

In the past decades, considerable attention has been paid to bio-inspired intelligence and its applications to robotics. This paper provides a comprehensive survey of bio-inspired intelligence, with a focus on neurodynamics approaches, to various robotic applications, particularly to path planning and control of autonomous robotic systems. Firstly, the bio-inspired shunting model and its variants (additive model and gated dipole model) are introduced, and their main characteristics are given in detail. Then, two main neurodynamics applications to real-time path planning and control of various robotic systems are reviewed. A bio-inspired neural network framework, in which neurons are characterized by the neurodynamics models, is discussed for mobile robots, cleaning robots, and underwater robots. The bio-inspired neural network has been widely used in real-time collision-free navigation and cooperation without any learning procedures, global cost functions, and prior knowledge of the dynamic environment. In addition, bio-inspired backstepping controllers for various robotic systems, which are able to eliminate the speed jump when a large initial tracking error occurs, are further discussed. Finally, the current challenges and future research directions are discussed in this paper.

Keywords: Biologically inspired algorithms, neurodynamics, path planning, mobile robots, cleaning robots, underwater robots, tracking control, formation control

1. INTRODUCTION

From the first stirrings of life, nature has been providing a suitable breeding ground for the intelligence of organisms. Biological intelligence enables organisms to adapt the extreme or changing environments. For instance, a group of birds and fishes can efficiently sense the surrounding dynamic environments and take effective actions based on those inputs often with very simple mechanisms and with limited availability of information. Some species exhibit collective behaviors and can cooperatively accomplish tasks that are beyond the capabilities of a single individual under limited implicit communication. Organisms with such beneficial traits can pass on these traits to offspring, exhibiting high adaptability to environments. The nervous system in the brain gives human abilities of feeling, thinking, and learning abilities.

Recently, there has been a general movement towards service-oriented robots that require the ability to adapt to complex dynamic situations and to handle various uncertainties. Due to the desirable properties of biological organisms, such as adaptability, robustness, versatility, and agility, the researchers have been trying to infuse robots with biological intelligence that will enable safe navigation and efficient cooperation among the autonomous robots in changing environments[1]. The approaches inspired by biological intelligence are known as biologically inspired intelligence, which has been explored and studied for many years in robotics research[2]. The fundamental idea of biologically inspired intelligence is to incorporate useful biological strategies, mechanisms, and structures into the development of new methodologies and technologies to solve existing problems in a more efficient way than existing methodologies and technologies. For instance, swarm intelligence and collective behaviors of living organisms have inspired the design of many robotics algorithms based on their biological strategies[3,4]. The process of natural selection has inspired many computational models to optimize robot performances, such as genetic algorithm[5,6] and differential evolution[7]. The neural network algorithm, derived from neural science, has gained rising popularity among researchers around the world[8,9]. Biologically inspired intelligence algorithms were also integrated with various conventional algorithms to develop more efficient algorithms. For example, a knowledge based genetic algorithm, which incorporated the domain knowledge into its specialized operators, was proposed to efficiently generate collision-free path of robots[10]. A neural network was used to convert the improved central pattern generator output to the foot trajectories of quadruped robots[11]. However, most bio-inspired studies are limited to conceptual or laboratory investigations or do not have much biological inspiration. Thus, the development of new intelligent strategies, algorithms and technologies are still highly needed, such as real-time collision-free navigation algorithms of individual robots or communication, coordination, and cooperation algorithms for multiple robotic systems, to accomplish multi-objective tasks in dynamic environments.

Bio-inspired neurodynamics models have been studied for real-time path planning and control of various robotic systems during the past decades[2]. The shunting neurodynamics model was derived from Hodgkin and Huxley's membrane models for dynamic ion exchanges[12]. Based on the shunting neurodynamics model and its model variants, several new algorithms have been successfully developed for real-time path planning and control of various autonomous robots[13,14]. The definition of real-time is in the sense that the robot path planner and controller respond immediately to the dynamic environment, including the robots, targets, obstacles, sensor noise and disturbances. Many other model variants have been also developed for robot path planning and control. The additive model is computationally simpler and can generate real-time collision-free paths under most conditions[13,15]. The gated dipole model shows excellent performance in multi-robotic path planning and tracking control[16]. Beyond the application of autonomous robots, bio-inspired neurodynamics models have been also widely applied to many other research fields, such as odor dispersion with electronic nose[17] and dynamic ginseng drying[18]. These researches on agriculture have also been extended to biomedical and other industrial applications.

This paper focuses a comprehensive survey of the state-of-the-art research on bio-inspired neurodynamics models with their applications to path planning and control of autonomous robots. A detailed introduction

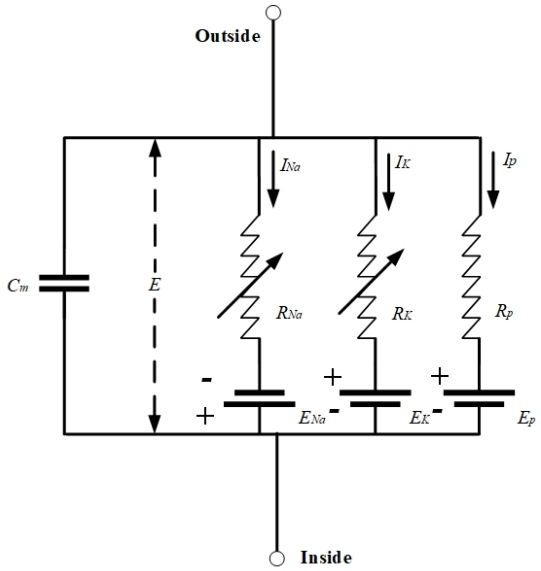


Figure 1. Electrical circuit representing membrane.

of the shunting model and its variants are provided in this paper. Two main applications to robotic systems based on bio-inspired neurodynamics models are focused. The bio-inspired neural networks, in which each neuron is characterized by a neurodynamics model, is discussed for various robotic systems. The bio-inspired backstepping controllers that resolve the speed jump problem in tracking control is further discussed. The bio-inspired controllers have been successfully employed in tracking control and formation control. The pros and cons of different neurodynamics-based algorithms are also discussed in this paper. The overall studies give an insight into neurodynamics models on autonomous robot applications, which could inspire potential ideas for future developments of novel intelligent bio-inspired path planning and control for diversified autonomous robotic systems.

This paper is organized as follows: Section 2 introduces the background of bio-inspired neurodynamics models. Section 3 gives a survey on the path planning of various robots based on bio-inspired neurodynamics models. The applications of bio-inspired neurodynamics models to tracking control and formation control are presented in Section 4. Section 5 discusses the current challenges and future works. Some concluding remarks are finally summarized in Section 6.

2. BIO-INSPIRED NEURODYNAMICS MODELS

In this section, the originality of the shunting model is briefly described. Then, two model variants, the additive model and gated dipole model are also introduced.

2.1. Originality

In 1952, an electrical circuit model was proposed by Hodgkin and Huxley to describe the action potential process in the membrane of neurons, based on experimental findings[19]. The electrical behavior of the membrane can be represented by the circuit shown in Figure 1. The dynamics of voltage across the membrane, V_m, is described using the state equation technique as

$$C_m \frac{dV_m}{dt} = -\left(E_p + V_m\right) g_p + \left(E_{Na} - V_m\right) g_{Na} - \left(E_K + V_m\right) g_K \tag{1}$$

where C_m is the membrane capacitance; E_K, E_{Na}, and E_p are the Nernst potentials (saturation potentials) for potassium ions, sodium ions, and passive leak current in the membrane, respectively; and g_K, g_{Na} and

g_p represent the conductances of the potassium, sodium, and passive channels, respectively. Inspired from this membrane model for dynamic ion exchanges, Grossberg proposed a shunting model[12,20,21]. By setting $C_m = 1$ and substituting $u_k = E_p + V_m$, $A = g_p$, $B = E_{Na} + E_p$, $D = E_k - E_P$, $S_k^e = g_{Na}$, and $S_k^i = g_K$ in Equation (1), a shunting equation is obtained as[22,23]

$$\frac{dx_k}{dt} = -Ax_k + (B - x_k)\, S_k^e - (D + x_k)\, S_k^i \tag{2}$$

where x_k is the neural activity (membrane potential) of the k-th neuron; A, B, and D are nonnegative constants representing the passive decay rate, the upper and lower bounds of the neural activity, respectively; and S_k^e and S_k^i are the excitatory and inhibitory inputs to the neuron, respectively. In the shunting model, B and D are not essential factors because the neural activity is the relative values between the boundary lines. Only parameter A determines the model dynamics. However, A can be chosen in a very wide range. Thus, the shunting model is not very sensitive to the model parameters[13].

Equation (2) shows that the increase of activity x_k depends on the positive term $(B - x_k)S_k^e$ that relies on both the excitatory input S_k^e and the difference of neural activity to its upper bound $(B - x_k)$. Therefore, the increases of x_k become slower as the value of x_k is closing to the upper bound B. If the value of x_k equals to B, the $(B - x_k)$ term becomes zero, and positive term has no effect no matter how big the excitatory input S_k^e is. In the case that the value of x_k is greater than B, the $(B - x_k)$ term becomes negative, then the positive term becomes negative, the excitatory input will decrease the activity x_k until it is not higher than B. Therefore, B is the upper bound of the neural activity x_k. The same for the negative term $(D+x_k)S_k^i$, which guarantees that the neural activity x_k is always greater than the lower bound $-D$. Thus, the neural activity x_k is bounded between the $[-D, B]$ region under various inputs conditions. The shunting model has been studied to understand the adaptive behaviors of individuals in dynamic and complex environments[12]. Many achievements have been accomplished in the past decades, such as, machine vision, sensory motor control, and many other areas[21,22]. In the field of robotics, the shunting model has been wildly used in path planning, tracking control, hunting, cooperation of various autonomous robots[13,24–26].

2.2. Model variants

If the excitatory and inhibitory inputs in Equation (2) are lumped together and the auto-gain control terms are removed, then Equation (2) can be written into a simpler form

$$\frac{dx_k}{dt} = -Ax_k + S_k \tag{3}$$

where S_k is the total inputs of the k-th neuron. Then, Equation (3) is rewritten as:

$$\frac{dx_k}{dt} = -Ax_k + I_k + \sum_{l=1}^{N} w_{kl} f(x_l) \tag{4}$$

where w_{kl} is the connection weight from the l-th neuron to the k-th neuron; $f()$ is an activation function; I_k represents the external input to the k-th neuron; and N is the total number of neurons in the neural network. In most situations, the additive model is computationally simpler and can also generate the real-time collision-free path for robots. However, the shunting model has two important advantages. Firstly, the shunting model in Equation (2) has excitatory and inhibitory auto-gain control terms, $(B-x_k)$ and $(D+x_k)$, respectively, which give the shunting model the dynamic responsive ability to input signals. The shunting model is more sensitive to the changes of inputs[13]. Nevertheless, the dynamics of the additive model may saturate in some situations. Secondly, the shunting model is bounded between the upper bound B and lower bound $-D$, whereas the additive model is bounded only by limiting the input signals. The additive models have been widely applied to artificial vision, learning-based algorithms, and other research fields[21]. Owning to the simple computation process, even the limitations of the additive model exist, the additive model has been also applied to replace the shunting model in many situations[13,15].

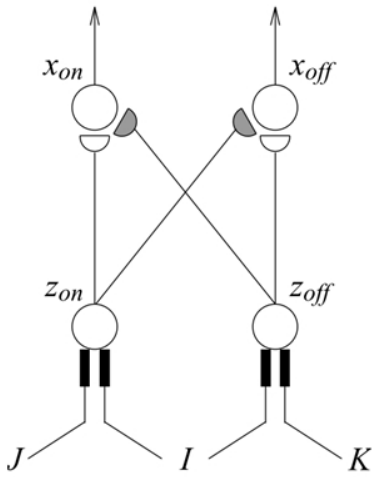


Figure 2. A gated dipole model.

Another essential neurodynamics model is the gated dipole model, which is shown in Figure 2. A basic gated dipole model is consisted of the opponent on-channel and off-channel. An arousal signal I can stimulate both on- and off- channels. The extra inputs J and K stimulate the on-channel and off-channel, respectively. The dynamics of the available transmitters are characterized by

$$\frac{dz_{on}}{dt} = \alpha\,(\beta - z_{on}) - \gamma(I + J)z_{on} \tag{5}$$

$$\frac{dz_{off}}{dt} = \alpha\left(\beta - z_{off}\right) - \gamma(I + K)z_{off} \tag{6}$$

where z_{on} and z_{off} are the number of available transmitters in the on- and off-channels, respectively; α and γ are the transmitter production and depletion rates, respectively; and β represents the total amount of transmitter. The on-cells receive excitatory inputs from the on-channel, while receive inhibitory inputs from its opponent channel (off-channel). Similar to the off-channel, the off-cells receive excitatory inputs from the off-channel, while receive inhibitory inputs from its opponent channel (on-channel). Thus, the dynamics of the on- and off-channels are characterized by the following shunting equations

$$\frac{dx_{on}}{dt} = -Ax_{on} + (B - x_{on})\,(I + J)z_{on} - (D + x_{on})\,(I + K)z_{off} \tag{7}$$

$$\frac{dx_{off}}{dt} = -Ax_{off} + \left(B - x_{off}\right)(I + K)z_{off} - \left(D + x_{off}\right)(I + J)z_{on} \tag{8}$$

where x_{on} and x_{off} are the activities of the on-channel and the off-channels, respectively. In the on-channel, the available transmitters decrease exponentially to a plateau when the extra light J is on, and goes back to its initial resting level in the same manner after the offset of the light. The available transmitter in the off-channel stays constant since there is no change of light. In the on-channel, when the extra light J turns on, there is more available transmitter depleted, and the response of the on-cell initially overshoot. However, after the onset of light, the available transmitter decreases exponentially due to temporal adaptation, the activity of on-cell decays exponentially to a plateau. At the offset of the extra input, the on- and off-channels have the same input I, while the available transmitter in the on-channel is used up by the depletion during the on-light, the activity of the on-channel appears a rebound which is called antagonistic rebound. After the extra light-off, due to temporal adaption, the available transmitter rises exponentially to its resting level, the activity of the on-channel climbs exponentially back also. The gated dipole model was successfully used to explain many biological phenomena that involve agonist and antagonist interaction[27], and had applications for robotic research of path planning and tracking control[16,28].

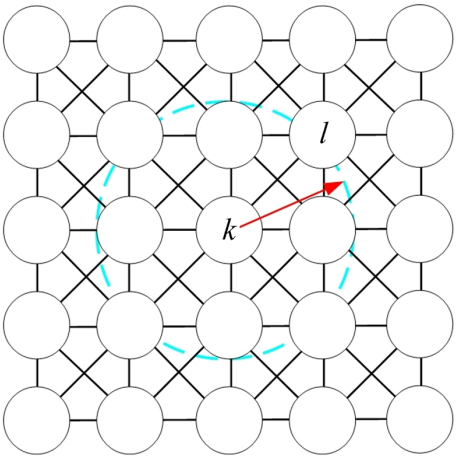


Figure 3. An example of the bio-inspired neural network.

3. PATH PLANNING

A basic path planning problem can be defined as: given a work environment with obstacles, the target, and the initial robot position, a collision-free path should be generated from the initial position to the target. The bio-inspired neurodynamics model has been wildly used in real-time path planning without any learning procedures and any prior knowledge of the dynamic environment. The key point of the neurodynamics-based path planning approach is to represent the work environment as one-to-one corresponding to the neurons in the neural network. The dynamics of each neuron in the neural network are characterized by Equation (2). An example of a neural network is shown in Figure 3. Thus, the neural activity for the k-th neuron is obtained by

$$\frac{\mathrm{d}x_k}{\mathrm{d}t} = -Ax_k + (B - x_k)\left([I_k]^+ + \sum_{l=1}^{n} w_{kl}[x_l]^+\right) - (D + x_k)[I_k]^- \tag{9}$$

where x_l represents the activity value of those neighboring neurons; n is the number of neighboring positions of the k-th neuron; $[a]^+$ is a linear-above-threshold function defined as $[a]^+ = \max\{a, 0\}$; and $[a]^-$ is defined as $[a]^- = \max\{-a, 0\}$. In the bio-inspired neural network, the excitatory input S_k^e is consisted of two parts, $[I_k]^+$ and $\sum_{j=1}^{n} w_{kl}[x_l]^+$, where $[I_k]^+$ is the external input from targets, and $\sum_{j=1}^{n} w_{kl}[x_l]^+$ is the internal input through the propagation of the positive activity from its neighborhoods. The inhibitory input S_k^i has only external input $[I_k]^-$, which is from the obstacle, and only has local effects (no negative activity propagation). Thus, the target has maximum and positive neural activity, which could globally propagate through the neural network to attract the robot, while the obstacles have only local effects without propagating. The path selection rule of the individual robot can be defined as: the next move position of the robot is the maximum neural activity of its current neuron's neighbors. After robots move to the next position, the next position becomes a new current position until the current position is the location of the target. The robot would never choose the position of an obstacle to be the next movement due to the negative neural activity of obstacles. Thus, the robot is able to avoid collisions and move to the target. It is important to note that the path planning process is without any learning procedures and any prior knowledge of the dynamic environment. Due to the real-time performance and computational efficiency, bio-inspired neural network path planning approaches have been developed for various robot systems. In this section, based on the different types of robots, three categories are divided: mobile robots, cleaning robots, and underwater robots.

3.1. Mobile robots

Path planning of mobile robots has received a lot of interest because mobile robots have been participating in human life. In this section, two main challenges of mobile robot path planning are focused: real-time collision-

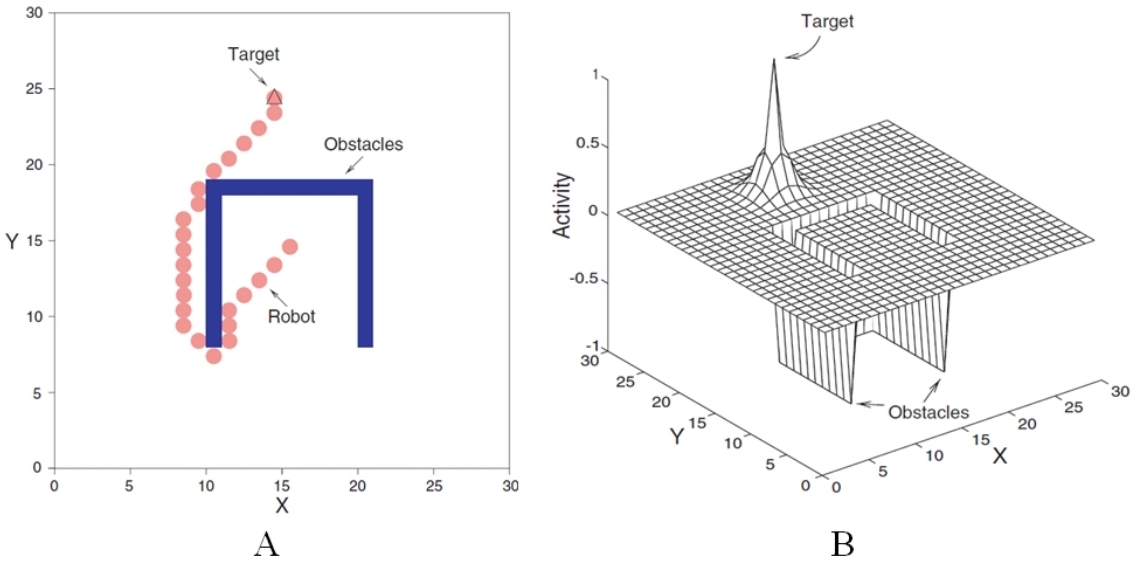


Figure 4. Path planning of a mobile robot to avoid local minima with concave obstacles. A: the robot path; B: the landscape of neural activity[13].

free navigation and the cooperation of the multi-robot systems. In addition, many developed model variants are also discussed for robot path planning.

3.1.1. Navigation

The first bio-inspired neural network framework was proposed by Yang and Meng for the mobile robot path planning[29]. Many remarkable achievements in mobile robot path planning have been achieved[13,30,31]. Due to the global effects of positive neural activity from the target, the robot is not trapped in the undesired local minima. Figure 4 shows an example of path generation of a mobile robot to avoid local minima. The robot is not trapped in a set of concave obstacles and move to the target position.

Some researchers consider the different types of robots in the application to navigation. A nonholonomic car-like robot was studied by Yang *et al.*[15,32,33] for real-time collision-free path planning. The simulation results showed the car-like robots performed well in parallel parking, navigation in several deadlock situations, and sudden environmental changes conditions. In a house-like environment as shown in Figure 5A, the robot moved to the target along the shortest path in case that the door is opened. When the door is closed, the robot travels a much longer path to reach the target without any learning procedures. The robot is capable of reaching the target along the shortest path without any collisions, without violating the kinematic constraint, and without being trapped in deadlock situations.

In addition, Yang and Meng developed the bio-inspired neural network for robot manipulators[13]. The joint space of the robot manipulators was corresponded to the bio-inspired neural network, in which neurons were characterized by the shunting model or the additive model. Figure 5B shows the trajectory of robot manipulators avoiding obstacles. In addition, a virtual assembly system was proposed by Yuan and Yang for assisting product engineers to simulate the assembly-related manufacturing process[34].

An improved bio-inspired neural network based on scaling terrain was proposed by Luo *et al.*[35] for reducing the calculation complexity. This multi-scale method mentioned better performance in terms of time complexity. However, the simulation experiments do not give the criteria for choosing the parameter of coarse-scale and fine-scale maps. Ni *et al.*[36] used a bio-inspired neurodynamics model as the reward function for the

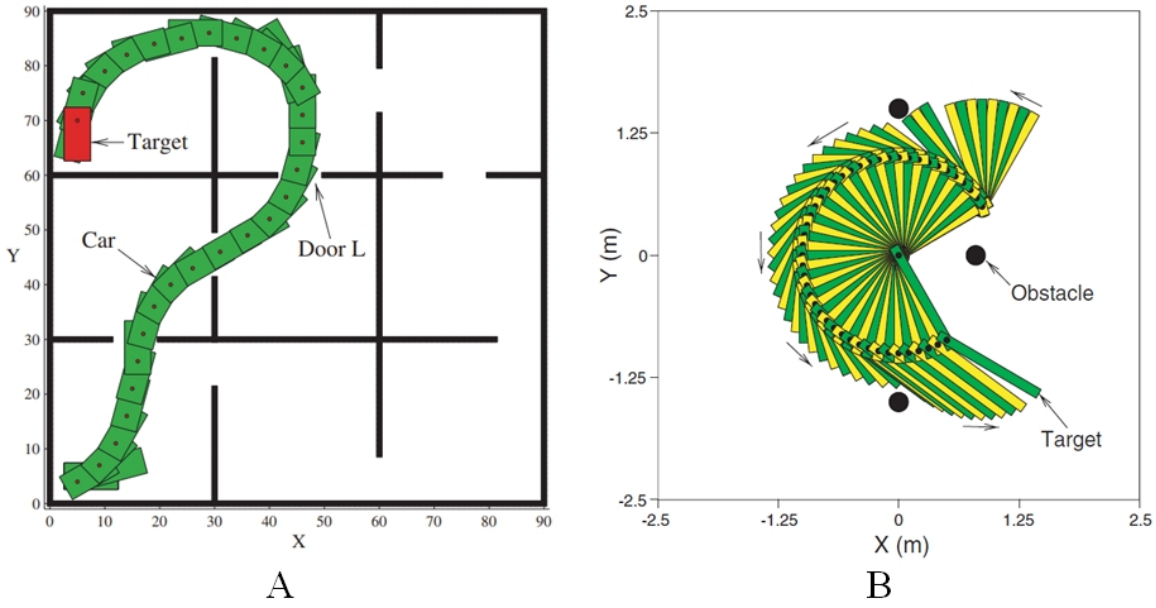


Figure 5. Examples of a nonholonomic car-like robot and a manipulator robot. A: robot motion when the door is opened [15]; B: simple planar robot avoiding obstacles [13].

Q-learning algorithm, which can reduce the effect of the reward function on the convergence speed.

Some researchers pointed out that if the planned path is too close to the obstacles, it is dangerous for robot navigation. A dynamic risk level was incorporated to the shunting neurodynamics model to reduce the probability of collision in the dynamic obstacle avoidance task [37]. In addition, a novel 3-D neural dynamic model was proposed and expected to obtain the safety-enhanced trajectory in the work space considering of minimum sweeping area [38]. A safety consideration path planning can be implemented by setting a constant value σ to inhibitory inputs in Equation (2). The safety consideration shunting equation is obtained by [39,40]

$$\frac{\mathrm{d}x_k}{\mathrm{d}t} = -Ax_k + (B - x_k)\left([I_k]^+ + \sum_{l=1}^{n} w_{kl}\,[x_l]^+\right) - (D + x_k)\left([I_k]^- + \sum_{l=1}^{n} v_{kl}\,[x_l - \sigma]^-\right) \tag{10}$$

where parameter σ is the threshold of the inhibitory lateral neural connections. In Equation (2), the inhibitory input S_k^i is only from the obstacles. However, in the safety consideration model, the inhibitory input S_k^i is consisted of two parts: $[I_k]^-$ and $\sum_{l=1}^{n} v_{kl}\,[x_l - \sigma]^-$.The $\sum_{l=1}^{n} v_{kl}\,[x_l - \sigma]^-$ term guarantees that the negative activity propagates to a small region due to the threshold σ of the inhibitory lateral neural connections. Thus, there is a small negative neural activity region surrounding the obstacles, and the robot is able to keep a safe distance from obstacles to avoid possible collisions.

Many variants of the bio-inspired neurodynamics models have been developed to deal with different situations. The additive model generates the real-time collision-free robot paths under most conditions [13]. Even the computation of the additive model is simpler, the real-time performance of the additive model could be saturated in many situations. A similar neural network model was proposed by Glasius *et al.* [41] for real-time trajectory generation. Even Glasius's model had limitations with fast dynamic systems, Glasius bio-inspired neural network models have been used in underwater robots [42–44]. Inspired by the bio-inspired neural network model, a distance-propagating dynamic system was proposed that can efficiently propagate the distance instead of the neural activity from the target to the entire robot work space [45]. After that, Willms and Yang designed the safety margins around obstacles. The robots not only avoid obstacles but also keep a safe distance between the obstacles [46]. Based on Willms and Yang's previous work, a shortest path neural networks model

was proposed by Li *et al.*[47] for generating the globally shortest path. A modified pulse-coupled neural network was proposed by Qu *et al.*[48,49] for real-time collision-free path planning. The computational complexity of the algorithm was only related to the length of the shortest path. In addition, an improved Hopfield-type neural network model was proposed by Zhong *et al.*[50] for easily responding the real-time changes in dynamic environments. A padding mean neurodynamics model was proposed by Chen *et al.*[51] for the reasonable path generation in both static and dynamic varying environments.

3.1.2. Cooperation of Multi-robotic Systems

A team of robots would work together to accomplish an assigned task rapidly and efficiently. In many challenging applications such as search and rescue operations, security surveillance and safety monitoring, a multi-robotic system has obvious advantages than a single robotic system. The key challenge of multi-robotic systems in dynamic environments is to infuse these robots with biologically inspired intelligence that will enable efficient cooperation among the autonomous robots, and successful completion of designated tasks in changing environments.

For the multiple targets path planning, an online solution based on the bio-inspired neural network was proposed by Bueckert *et al.*[52] in static and dynamic environments. However, the task assignment approach was very simple, as the robot visited the target, this target was removed from the visit list. Thus, the robot was hard to find the optimal visit sequence of targets. A novel hybrid agent framework was proposed by Li *et al.*[53] for real-time path planning to multi-robotic systems considering many moving obstacles. In this work, an improved shunting equation was proposed by setting safety margins for the robots and the moving obstacles. The robots are able to predict the movement of obstacles and avoid any collision. Nanoassembly planning creates enormous potential in a vast range of new applications. An integrated method based on the shunting model was proposed to generate collision-free paths of multi-robotic nanoassembly[54]. The tasks of the multi-robotic nanoassembly planning were a continuous process considering the environmental uncertainty.

If the robotic systems need to track the moving targets, an important influence of the algorithms is the relative moving speed between the target and robot[55]. If the speed of robotic systems is much lower than the target, the robotic systems need to corporately track the target, otherwise the robot will never catch the target. A real-time cooperative hunting algorithm was proposed by Ni and Yang base on shunting neurodynamics models[56]. In this hunting task study, the robots had no previous knowledge about the environment and locations of evaders. It is important to note that the difference between tracking a moving target and the hunting algorithm is that the evader in hunting problems has some intelligence to escape from the hunt of pursuer robots. Figure 6A shows the hunting process considering many evader robots. Compared with other hunting algorithms, the hunting algorithm based on bio-inspired neurodynamics still works efficiently when some hunting robots are broken. Figure 6B shows the hunting process that some robots are broken.

3.2. Cleaning robots

The cleaning tasks require the robot to pass through every area in the work environment. The task requirement is the same as the complete coverage path planning (CCPP), which is a special type of path planning in 2-D environments. The CCPP can be also applied to many other robotic applications, such as painter robots, demining robots, lawnmowers, automated harvesters, agricultural crop harvesting equipment, windows cleaners, and autonomous underwater covering vehicles[57,58]. In the bio-inspired neural network, the unclean areas are set as targets, which globally attract the robot. The obstacles have only local effects, which avoids robot collisions[59–61]. As the cleaning robot works, the unclean areas become clean and the excitatory input of the clean area becomes zero. Thus, the landscape of neural activity dynamically changes with the change of the unclean areas, obstacles, and other robot position. For any current position of the robot, the next robot position p_n is obtained by

$$p_n \Leftarrow x_{p_n} = \max\{x_l + cy_l, l = 1, 2, \cdots, n\} \tag{11}$$

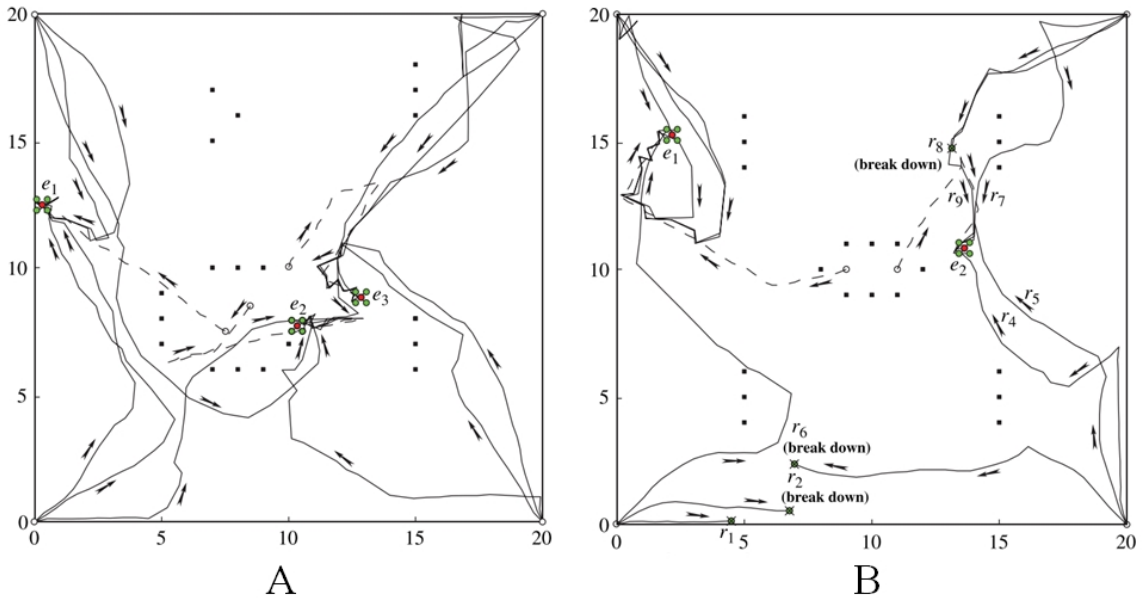


Figure 6. Examples of hunting tasks. A: multiple evaders need to be hunted; B: some robots break down[56].

where c is a positive constant and y_l is a monotonically increasing function of the difference between the current to next robot moving directions. Compared with path planning and CCPP problems, the main difference is that many target positions might attract the cleaning robot because all unclean areas are set as targets. Thus, the turning numbers of clean robots might increase significantly. Function y_l is designed to reduce the turning numbers. If the robot goes straight, y_l =1; if goes backward, y_l = 0. Thus, the cleaning robot tends to go straight.

In this section, based on the previous knowledge of the environment, the research fields of cleaning robots using the neurodynamics model are categorized as: completed known environment and unknown environment.

3.2.1. Completed known environment

Figure 7 shows the neurodynamics-based CCPP in a completely known environment. The neurodynamics model can work efficiently in the dynamic environment, so even considering sudden change environment and moving obstacles in the environment, the cleaning robots can still work efficiently[57,59,60]. In order to improve the computational complexity, a discrete bio-inspired neural network was proposed to convert to the shunting equation a difference equation[62].

One CCPP challenging problem is the deadlock situation. The deadlock area is a specific situation that the cleaning robot is trapped in a position where all of the neighborhood areas have been covered, but the work environment is still unclear. If the cleaning robot moves to deadlock areas, the cleaning robot is unable to escape from the deadlock areas without any interventions. A dynamic neural neighborhood analysis for deadlock avoidance was proposed based on the characteristics of deadlock areas[59]. The robot can recognize whether the current position is the deadlock point. If the current position is a deadlock point, the connection weights of the neural network were changed to generate a path to escape this deadlock point.

3.2.2. Unknown environment

In order to deal with CCPP in the unknown environment, the cleaning robots are typically required to build a surrounding map with a very limited time range[63]. The onboard sensors have been widely used for robot

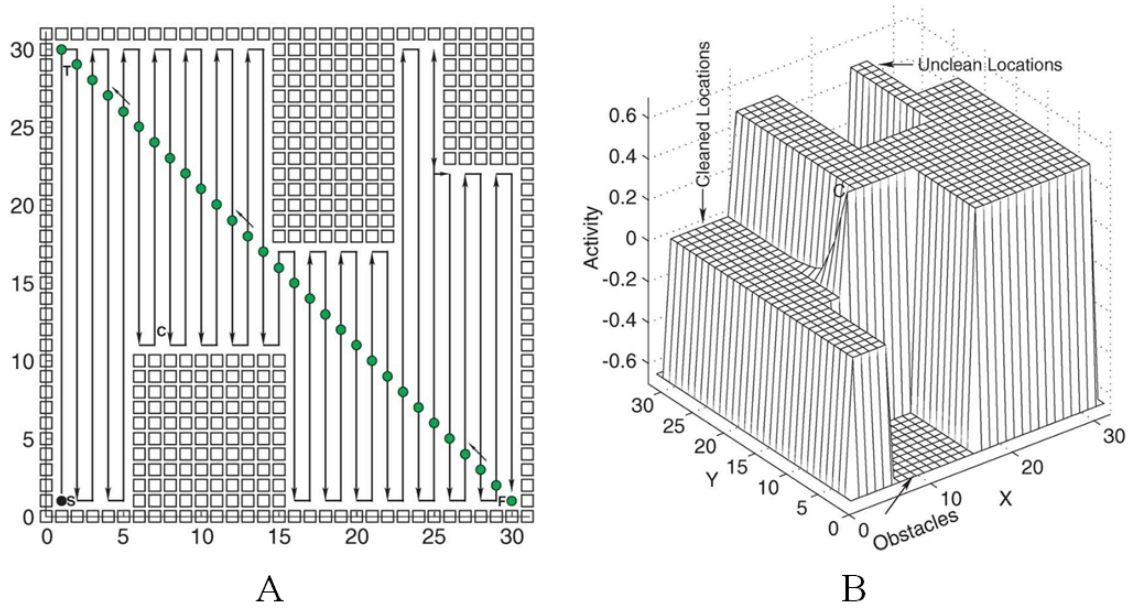


Figure 7. CCPP in a completely known environment. A: the generated robot path; B: the neural activity landscape when the robot reaches point C [63].

navigation with a limited reading range. Thus, the key challenge of CCPP in unknown environments is to design the map-building algorithm and combine it with previous coverage algorithms studies. Combing with the sensor detection, an improved CCPP algorithm based on the neurodynamics model was developed in unknown environments [64,65]. The robots move to the nearest unclean areas and detect the environment until the cleaning task is finished. A real-robot platform iRobot Create 2 was used to test the proposed algorithm in unknown environments [66]. The actual cleaning robots testing showed that the effectiveness of the proposed algorithm, in which the robotic systems could cooperatively work together in a large and complex environment.

3.3. Underwater robots

The autonomous underwater vehicle (AUV) or unmanned underwater vehicle (UUV) have been studied in a variety of tasks such as underwater rescue, data collection, and ocean exploration. In addition, some bionic robots are also studied, such as robotic fish [67,68]. Unlike the work environment of mobile robots or cleaning robots, the underwater environment is more complex and uncertain. Firstly, based on the 2-D neural network structure, a 3-D grid-based neural network is typically required to represent the underwater environment. Secondly, the effect of the ocean or river currents is necessary to consider. Finally, the robots work in underwater environments, facing many uncertainties, such as some robots broken down. Based on different task requirements, three major research fields of underwater robots using the neurodynamics model are studied in this section.

3.3.1. Navigation

For the underwater environment, the neural network architecture needs to be extended to the 3-D environment, where more complex topography of randomly distributed obstacles is involved. Figure 8 shows a typical AUV path planning in 3-D underwater environments. In 2-D neural network architecture, each neuron connects with 8 neighborhood neurons, whereas, in the 3-D neural network, each neuron connects with 26 neighborhood neurons [69]. Thus, the computation complexity dramatically increased. In order to improve the efficiency in the 3-D underwater environment, a dynamic bio-inspired neural network was proposed to guide the movement of AUV in large unknown underwater environments [25]. A virtual target selection approach was applied to search the path and avoid dead loop situations. Since the large unknown environment is par-

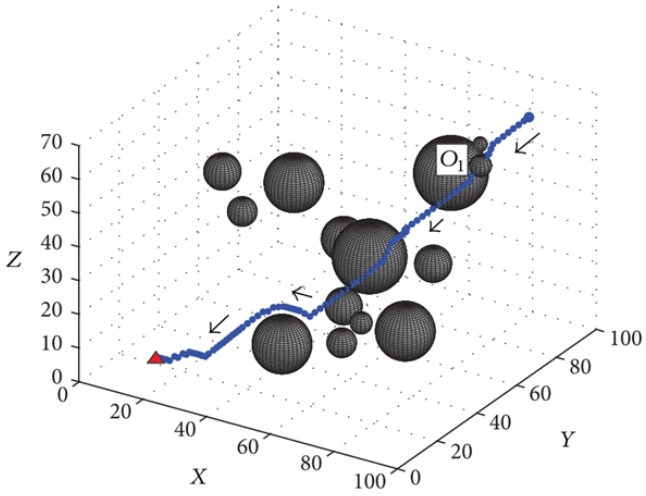


Figure 8. An example of path planning in 3-D underwater environments[25].

titioned into small portions centered with moving AUV, and the bio-inspired neural network only deals with this small range, the path can be calculated relatively fast. However, the dynamic bio-inspired neural network misses the best route in certain circumstances, which could waste the power of the AUV.

The environmental disturbances of the underwater area, such as currents, create inevitable influences on the AUV path planning. A current effect-eliminated bio-inspired neural network was proposed to guide the AUV navigation considering the effect of currents[70]. A current correcting component was incorporated with the bio-inspired neural network to generate the paths. Each neuron in the network, the velocity and direction of robots are corrected for eliminating the current effect. Thus, the generated UUV path is robust and efficient.

The real-world ocean environment is complex and unknown. The onboard robot sensors were used for robot navigation with a limited detection range. The ultrasonic sensor was used to interpret the sonar data and update the map based on the dempster's inference rule[71]. A potential field bio-inspired neural network (PBNN) was proposed to generate a safe path in underwater environments[72,73]. The planned path keeps a safe distance to the obstacles, which could avoid the collisions for the underwater robot navigation.

Multi-AUV systems cooperation has received lots of interest due to the fact that groups of AUVs can work more efficiently and effectively compared with a single AUV. The main task of AUVs cooperation is to assign several targets to a team of AUVs and avoid obstacles autonomously in underwater environments. Due to the similarity of multi-tasks assignment and self-organizing map (SOM) neural networks, many researchers have been applied the SOM approach to solve task assignment problems of multi-robotic systems[74–76] and multi-AUV systems[77,78]. However, the SOM-based methods require an ideal 2-D work environment without obstacles. An integrated biologically inspired SOM (BISOM) method was proposed to deal with collision-free and multi-AUV task assignment problems[79]. After integrated the bio-inspired neural network method, the AUV is able to avoid obstacles and speed jumps. The ocean currents could influence the AUV navigation in the underwater environment. A velocity synthesis algorithm was integrated with the BISOM approach for optimizing the individual robot path in a dynamic environment considering the ocean current[80].

The BISOM method is able to generate the shortest path for the multi-robotic systems in most situations. However, the update rule of the BISOM method ignores the effect of obstacles. Therefore, although the winner AUV is the shortest distance from the target, the obstacles could increase the movement of the winner robot. A novel biologically inspired map algorithm was proposed by Zhu *et al.*[81] for changing the update

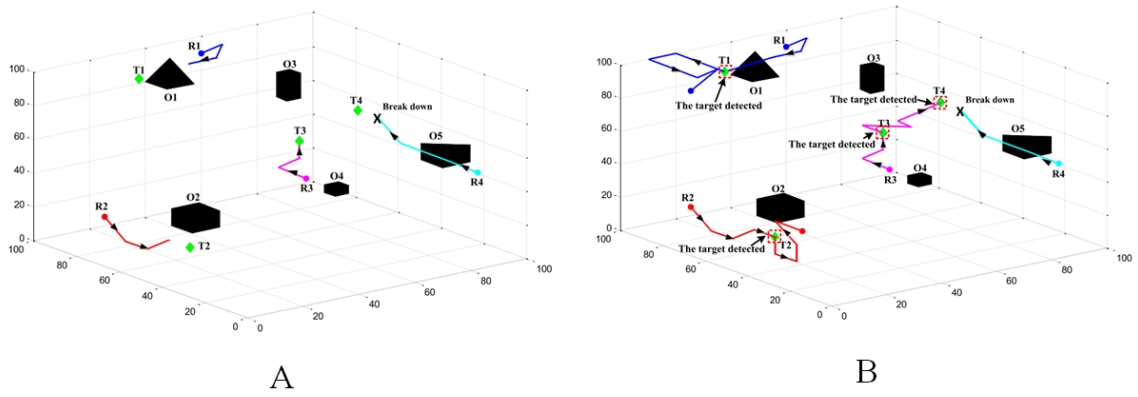


Figure 9. Examples of target search tasks. A: the AUV R4 breaks down; B: final trajectories of the search process [83].

rule. The winner rule is not the shortest distance between target and AUV, whereas the winner rule becomes the maximum neural dynamic value in the neural activity values.

3.3.2. Target search

The fundamental problem of target search for multi-AUV search systems is how to control all the vehicles to search to their target along the optimized paths cooperatively. The initial work on search was carried out by simplifying the search problem as an area coverage problem. As same as in cleaning robot application, the landscape of neural activity can guide the robot to search every unknown areas until the target was searched [82]. However, the coverage algorithm is not an efficient search algorithm as the robot power is wasted by unnecessary visiting positions. In order to improve the efficiency of the search algorithm, a sonar system was applied to extract the information of the environment to build the map and localize the target location [83]. Figure 9 shows that the proposed algorithm not only enabled the multi-AUV team to achieve search but also ensures a successful search if one or several AUVs fail. However, factors in real environments, such as ocean currents, were excluded in this simulation and there might be a waste of search capacity because of the overlapping search spaces. Same as the navigation application, with the consideration of ocean current, an integrated method based on the neurodynamics model and velocity synthesis algorithm was proposed for the cooperative search of the multi-AUV system [84].

3.3.3. Hunting

Based on the previous study of neurodynamics model hunting for mobile robots in 2-D environments, a 3-D underwater environment hunting algorithm was proposed [85,86]. Compared with Ni and Yang's hunting algorithm [56], the catching stage was very different in applying underwater robots. The final hunting state can be divided into four situations. Figure 10 shows one of the hunt situations that four AUVs surrounded the target.

The path conflict situation happened when multiple AUVs chose the same position to be the next movement. A collision-free rule was established that location information is recorded between each AUV and selects the next step grid in each vehicle in anticipation before the movement [87]. If any other AUV has occupied the grid, then choose another grid to move.

4. CONTROL

Robot control is ongoing research that tracks much attention. The control in robotics is to develop controllers that drive robot kinematics or dynamics to reach desired states. Intelligent control of the robot is to develop a

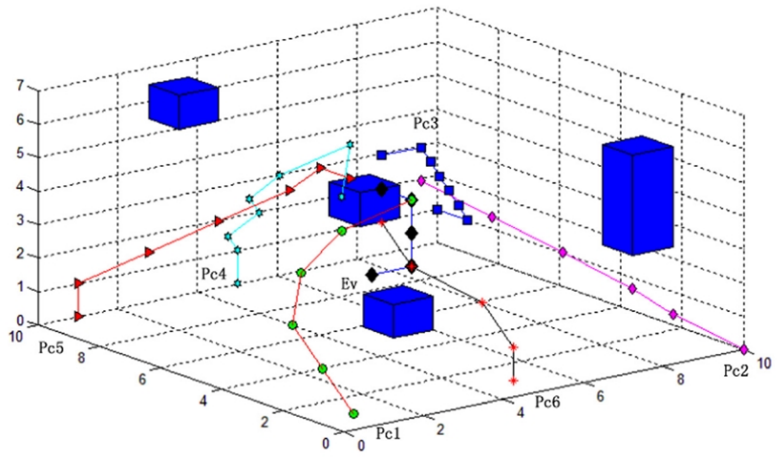


Figure 10. The hunting process with one target and six AUVs. E_v: the target; P_c1-P_c6: the AUVs [86].

controller by taking advantage of vital characteristics of human intelligence, such as fuzzy logic, neural network, etc. bio-inspired intelligent control mechanism is based on biological systems. Using this biologically inspired system is targeting to improve the control performance by the implementation of natural biological systems in the control design.

4.1. Tracking control

The tracking control of robots or motors has been studied for many years. Sliding mode control is robust to variable changes, however this method suffers chattering issues, which is a critical factor that needs to be considered when designing the control strategy. The linearization control method is easy to implement, however, it suffers from a large velocity jump when a large tracking error occurs at the initial stage. Backstepping control is easy to design, however, when a large tracking error occurs, this method becomes impractical as the speed jump will result in a large velocity surge, which can damage the hardware of the system. Neural network and fuzzy logic control are capable of resolving the large velocity jump at the initial stage, however, both neural network and fuzzy logic control are hard to practice. The neural network-based control methods require online learning, which is expensive and computationally complicate, the fuzzy logic control requires human experience to make the robot perform well, both of these control methods are rather expensive to practice.

The bio-inspired backstepping control, which is based on the backstepping technique, aims to eliminate the speed jump in conventional design when a large initial tracking error occurs. The general control design for the unmanned robot with the implementation of bio-inspired neural dynamics can be described in Figure 11. The motion planner plans the desired posture P_d, then the desired trajectory along with the feedback of the current posture P_c propagates through a transformation matrix to convert the tracking error from the inertial frame into body fixed frame. Then, the path tracker, which contains the bio-inspired backstepping controller uses the tracking error and desired velocity to generate a velocity command, which then along with the observed velocity υ_c propagate through torque controller to generate torque command, which drives the robot to generate a velocity and reach its desired posture by propagating the velocity that is generated from robot dynamics to robot kinematics.

The applications of bio-inspired backstepping control are mainly divided into three different platforms: mobile robots, surface robots, and underwater robots. Therefore, this section illustrates the efficiency, effectiveness, and applications of the bio-inspired backstepping control into these three different platforms.

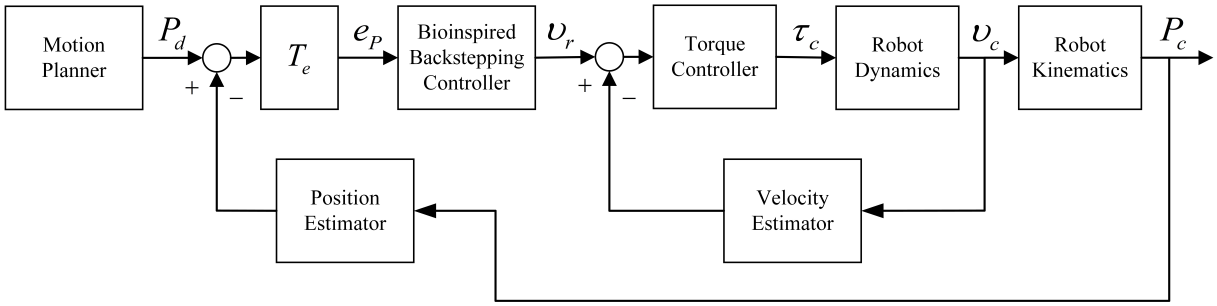


Figure 11. The block diagram of the bio-inspired tracking control for robots

4.1.1. Mobile robots

Real-time tracking control of a mobile robot is a challenging issue in mobile robotics. The main purpose of the tracking control is to eliminate or reduce the effects of errors. However, the disturbance, noise, and sensor errors will interfere with the output of the robotic system and produce errors. Many control algorithms of the mobile robot have been studied for precisely tracking a desired trajectory. The conventional backstepping control for mobile robots suffers from velocity jump issues, this problem is embedded in the design of the controller. The linear velocity error term causes the velocity jump if the initial tracking error does not equal zero. As seen this problem, the bio-inspired neural dynamics was brought into the design of the backstepping control. For a nonholonomic mobile robot operates in a 2-D Cartesian work space, the main control variables for its kinematic model are the linear velocity and angular velocity. Focusing on the design of solving the velocity jump issue, the bio-inspired backstepping kinematic control for a mobile robot is defined as

$$\upsilon_c = \upsilon_s + \upsilon_d \cos e_\theta \tag{12}$$

$$\omega_c = \omega_d + C_1 \upsilon_d e_L + C_2 \upsilon_d \sin e_\theta \tag{13}$$

$$\frac{d\upsilon_s}{dt} = -A\upsilon_s + (B - \upsilon_s)\,[e_D]^+ - (D + \upsilon_s)\,[e_D]^- \tag{14}$$

where υ_s is derived from neural dynamics equation regards to the error in driving direction for the mobile robot, C_1 and C_2 are the designed parameters, υ_d and ω_d are respectively the desired linear and angular velocity that are given at path planning stage, υ_c and ω_c are respectively the linear and angular velocity commands that generated from the controller, and e_D and e_L are respectively the tracking error in driving and lateral directions[14]. Compared to conventional design, the bio-inspired backstepping control takes the advantage of the shunting model that provides bounded smooth output.

The bio-inspired backstepping controller resolved the problem of sharp speed jumps at the initial stage[24,88,89]. The total design of the proposed control and path planning method were able to provide both real-time collision-free path and provide smooth velocity tracking commands for a nonholonomic mobile robot. However, the generated angular velocity seemed to suffer from sharp changes, therefore, the validation of the proposed control strategy is needed. In addition, the simulation environment is assumed as a simple environment with no obstacles. Zheng *et al.*[90] proposed an adaptive robust finite-time bio-inspired neurodynamics control with unmeasurable angular velocity and multiple time-varying bounded disturbances. The outputs were smooth and the sharp jumps of initial values were decreased.

In real-world applications, the model input of the mobile robot may have errors, therefore, to overcome the problem of this abrupt change in the generated velocities caused by the model input errors, a fuzzy neurodynamics-based tracking controller, which incorporated fuzzy control to generate smooth velocities, was proposed[91]. The proposed control considered the model input error that consequently have impacts on the tracking error, which was further reduced using fuzzy logic to incorporate with the bio-inspired backstepping control.

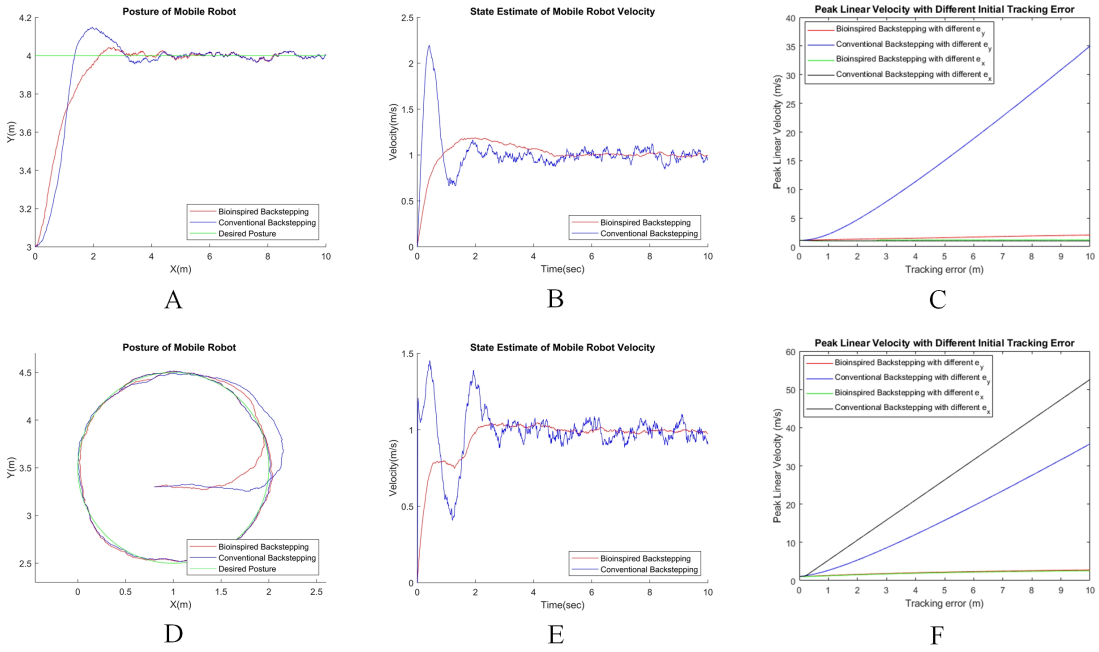


Figure 12. The comparisons of the traditional backstepping control and bio-inspired backstepping control. A: tracking a straight line; B: linear velocity estimates of tracking a straight line; C: peak linear velocity comparison of tracking straight line; D: tracking a circular line; E: linear velocity estimates of tracking a circular line; F: peak linear velocity comparison of tracking a circular line[95].

In addition, the shunting model also incorporated with PID controller to modify the error term, this control strategy provided a smooth velocity curve and more importantly, avoided impulse acceleration and torque, which could potentially damage the mechanical system[92].

In order to improve the efficiency and effectiveness of the bio-inspired backstepping control, the parameters of the control were determined using a genetic algorithm[93]. Tuning control parameters with the genetic algorithm provided better results than the implementation of bio-inspired backstepping control alone. Although the parameters tuned with the genetic algorithm provided satisfactory results, many other optimization methods could be used to choose the parameters, a comparison study could be tested to demonstrate the efficiency of the genetic algorithm. A biologically inspired full-state tracking control technique was proposed to generate smooth velocity commands[94]. The proposed control considered both position error and orientation error as the control input and used the shunting model to constrain its output to reach its goal of providing a smooth velocity curve. There are still some improvements can be made as the path itself is not smooth but has sharp turns before it tracks its desired trajectory in a straight line tracking simulation. In addition to the simulation studies, successful implementation on a real mobile robot system demonstrates the effectiveness of the bio-inspired backstepping controller[14]. The experiment results showed that the robot tracked both the straight path and the circular path, and simulation results provided smooth velocity curves.

The mobile robot usually works in a complicated environment, which system and measurement noises can affect its tracking accurate. Therefore, an enhanced a bio-inspired backstepping control was proposed to generate the smooth, accurate velocity and torque command for mobile robots, respectively[95]. The total control Incorporated bio-inspired backstepping controller with unscented Kalman and Kalman filters that were suitable in real-world applications. The proposed control considered noises in real-world applications, and the proposed control considered such noises effect and successfully eliminated it. However, the proposed control is considered a fixed noise, which is not true in real-world applications.

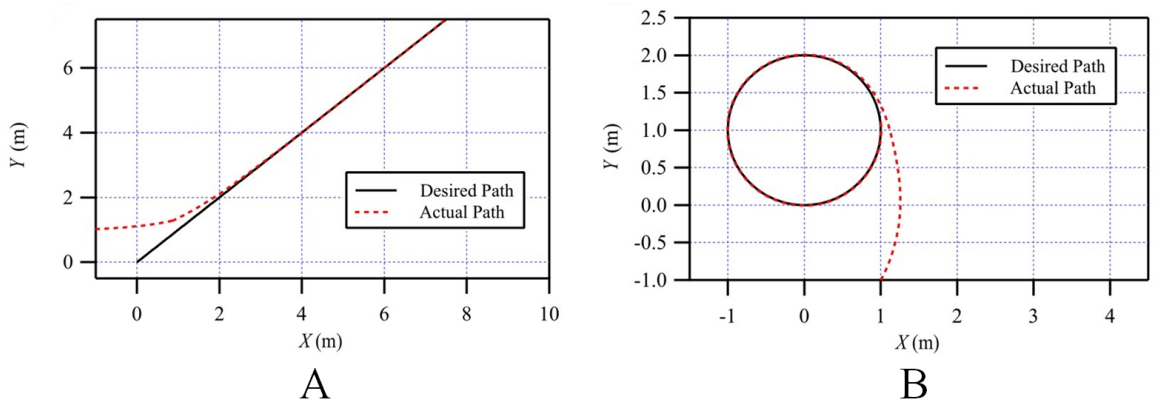


Figure 13. Tracking trajectory comparison of the bio-inspired method and conventional backstepping method for the underactuated surface vessel. A: line tracking; B: circle tracking[96].

To illustrate the efficiency and effectiveness of the bio-inspired backstepping control for a mobile robot, Figure 12 is chosen to show the superiority of the bio-inspired backstepping control over the conventional method. As seen from Figure 12A and Figure 12F, the larger the initial error occurs, the larger the initial velocity jump from the conventional method occurs, however, the bio-inspired backstepping control still makes the robot maintain a low initial velocity change. It is obvious that the bio-inspired backstepping control has practically solved a speed jump issue in backstepping control for a mobile robot, which is more practical in real-world applications.

4.1.2. Surface robots

The tracking problem of the unmanned surface vehicle (USV) usually refers to the design of a tracking controller that forces robots to reach and follow a desired curve, where 2-D and three DOF (surge, sway and yaw) are considered[96,97].

The bio-inspired backstepping controller was used to USV for dealing with the velocity-jump problem[98]. In the case that considering the impact of ocean current, a current ocean observer is fused with the control design to reduce the impact of ocean current in the tracking performance[99]. The bio-inspired backstepping controller was integrated with a single-layer neural network for underactuated surface vessels in unknown and dynamics environments[96]. The proposed tracking controller reduced the calculation process, therefore, the tracking controller avoided the complexity problem existed in conventional backstepping controllers. The stability of the tracking control system is guaranteed by a Lyapunov theory, and the tracking errors are proved to converge to a small neighborhood of the origin such that a satisfactory tracking result is presented in Figure 13.

4.1.3. Underwater robots

Bio-inspired neurodynamics models have been applied to the tracking control of underwater robots for many years[100]. The tracking control of the underwater robots is generally addressed by designing a control law that realizes asymptotically exact tracking of a reference trajectory based on the given underwater robots plant model[101]. However, different from common robots such as the land vehicle or the USV, the underwater robotics system contains more states, whose DOF can be extended to six. Among the six DOFs of the underwater robots, surge, sway, heave, roll, pitch, and yaw, roll and pitch can be neglected since these two DOFs barely have an influence on the underwater vehicle during practical navigation. Therefore, when establishing the trajectory tracking model to keep a controllable operation of the underwater robots, usually only four DOFs: surge, sway, heave, and yaw are involved. As same as the mobile robot, the speed jumps largely affect the robustness of the underwater robots path tracking. Due to the complex underwater work environment

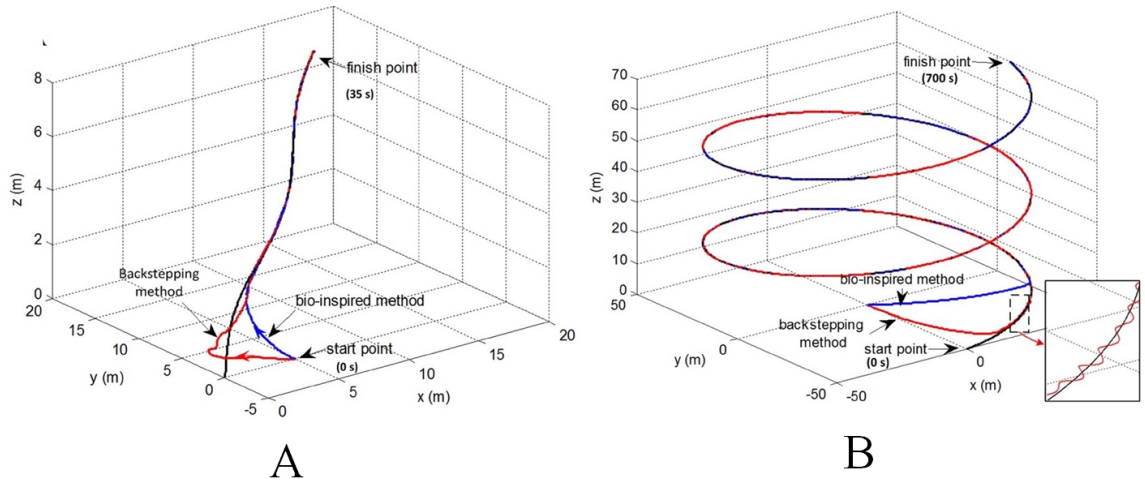


Figure 14. Tracking trajectory comparison of the bio-inspired model based method and conventional backstepping method for the underwater robots. A: curve tracking; B: helix tracking[105].

and limited electric power of underwater robots, the speed jumps as well as the driving saturation problem have to be considered. The bio-inspired backstepping controller was introduced in the control design to give the resolution respectively[102]. Due to the characteristics of the shunting model, the outputs of the control are bounded in a limited range with a smooth variation[103].

The bio-inspired backstepping controller has been applied on different underwater robots under various conditions by combining with a sliding mode control that controls the dynamic component of the vehicle, where an adaptive term is used in the sliding mode control to estimate the non-linear uncertainties part and the disturbance of the underwater vehicle dynamics[104]. For example, the driving saturation problem of a 7000m manned submarine was resolved through this bio-inspired backstepping with the sliding mode control cascade control[105]. The control contains a kinematic controller that used bio-inspired backstepping control to eliminate the speed jump when the tracking error occurred at the initial state. Then, a sliding mode dynamic controller was proposed to reduce the lumped uncertainty in the dynamics of the underwater robots, thus realizing the robust trajectory tracking control without speed jumps for the underwater robots Figure 14. Jiang *et al.*[106] accomplished the trajectory tracking of the autonomous underwater robots in marine environments with a similar bio-inspired backstepping controller and the adaptive integral sliding mode controller. In the sliding mode controller, the chattering problem was alleviated, which increased the practical feasibility of the vehicle. However, more studies are needed to compare to prove the effectiveness of the proposed control strategy, such as the tracking control based on the filtered backstepping method.

4.2. Formation control

The bio-inspired neurodynamics trajectory tracking control for a single nonholonomic mobile robot can be extended to the formation control for multiple nonholonomic mobile robots, in which the follower can track its real-time leader by the proposed kinematic controller. This section introduces leader-follower formation control based on the bio-inspired neurodynamics tracking controller into three different robot platforms.

4.2.1. Mobile robots

The leader-follower formation control based on the bio-inspired neurodynamics tracking controller was studied by Peng *et al.*[107]. The asymptotic stability of the closed-loop system was guaranteed. The issue of impractical velocity jump arising from the use of the backstepping approach was handled by means of the bio-inspired neurodynamics model. However, the control design was based on the level of the kinematics model so that the

performance of formation control relies highly on the low-level servo controller. The robustness property of the system is not analyzed, which is important when facing uncertainties or disturbances. Therefore, further improvements can be made based on these aforementioned points. A leader–follower formation control using a bioinspired neurodynamics-based approach was proposed by Yi *et al.*[108] for resolving the impractical velocity jump problem of nonholonomic vehicles. Simulation results demonstrate the effectiveness of the proposed control law.

In order to further improve the tracking performance, a non-time based controller was also proposed[6]. The path planner not only generated a desired path for the mobile robot, but also became part of the control to adjust the actual path and desired path. Along with the bio-inspired backstepping tracking control, the proposed method provided an overall better performance than a single backstepping control alone for multi-robotic systems.

4.2.2. Surface robots

To fulfill the requirement of accomplishing complex tasks in the unpredictable marine environment, where the ocean currents and the marine organism may affect the efficiency of the vehicle operation, formation control on the system of multiple USVs has become a hot topic in recent decades[109]. Studies of combining the bio-inspired model with the marine vehicle formation control have been proposed and the model is often used to achieve the intelligent planning results of the multi-vehicle system[110].

Regarding the bio-inspired model application on the USV formation control, a novel adaptive formation control scheme based on bio-inspired neurodynamics for waterjet USVs with the input and output constraints was proposed[111]. However, the learning process of the adaptive neural network can reduce the real-time performance, which is the superiority of the bio-inspired neural network. In addition, the robustness property of the resulting closed-loop system is not analyzed when the undesired perturbation is injected into the system, which is considered a critical problem in practical engineering.

For multi-robotic system operates in large and unknown environments, Ni *et al.*[26] used a dynamic bio-inspired neural network for real-time formation control of multi-robotic systems in large and unknown environments. The proposed approach considered many uncertain situations. Figure 15 shows that the multi-robotic systems still finish the formation task, when the leader USV was broken. However, the mathematical analysis for the proposed algorithm is not provided, such as convergence analysis and robustness analysis. Comparison with traditional approaches is not provided, thus it is not sufficient to demonstrate the efficiency of the proposed method.

Intelligent formation control for a group of waterjet USVs considering formation tracking errors constraints was proposed[112]. To guarantee line-of-sight range and angle tracking errors constraints, a time-varying tan-type barrier Lyapunov function is employed. Besides, the bio-inspired neurodynamics was integrated to address the traditional differential explosion problem, *i.e.*, avoiding the differential operation of the virtual control. However, the simulation example is much limited, thus the effectiveness and efficiency of the proposed control scheme are not sufficiently verified, *i.e.*, the lack of the comparison with another type of control method.

4.2.3. Underwater robots

For underwater robots, the definition of formation control is similar to the surface vehicle but with additional dimensions[113]. Formation control of the multi-UUV system considers both 2-D and 3-D, where the former focuses on the lateral movement of the vehicle groups[114].

A formation control on the multi-UUV system to realize the tracking of desired trajectory and obstacle avoid-

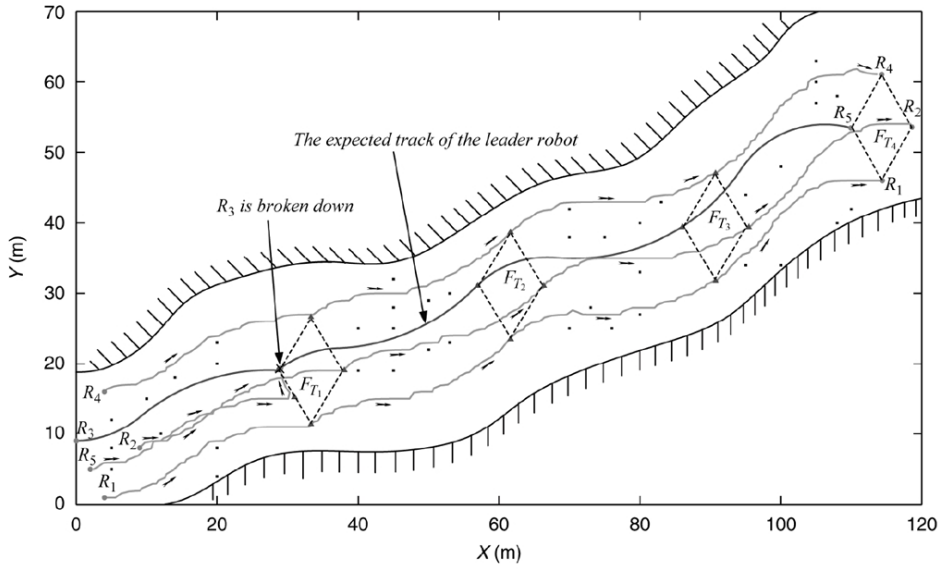


Figure 15. The simulation experiment in the case that the leader robot R_3 is broken down[26].

ance in the 3-D underwater environment was proposed by Ding *et al.*[115]. The bio-inspired neural network helps the leader UUV decide the transform of the formation when encountering obstacle fields to avoid the obstacles for all UUVs and meanwhile sustain on the desired trajectory. However, complex environmental disturbances such as multi-obstacles are not thoroughly considered in this paper.

5. CHALLENGES AND FUTURE WORKS

Although there have been many studies of bio-inspired intelligence with applications to robotics and remarkable achievements have been accomplished, there are still several challenges that would be further investigated as future works.

- Many existing approaches assumed the environment is static and without any uncertainties (*e.g.*, some robot breakdown), disturbance (*e.g.*, wind for surface robots, ocean/river current for underwater robots), and noise (system and measurement noise). However, it is a big challenge for collision-free robot navigation in complex changing environments with many moving robots/targets and subject to uncertainties, disturbance, and noise.
- Communication issue has always been an essential research area in robotics. It is important to build a stable communication network in multi-robotic systems to ensure the updating of neural activity in the bio-inspired neural network. However, most studies on the cooperation of multi-robotic systems did not consider the communication issue, where the communication is normally noisy and with time delay. Many approaches did not consider the optimal performance with multi-objectives (*e.g.*, short total distance, completion time, energy, smoothness of the robot paths). Communication and multi-objectives optimization could be a potential research direction in the future.
- Most conventional aerial robot navigation cannot act properly due to the limitations of communication and perception ability of sensors in complex environments. The complexity of the aerial robot makes the controllers are hard to design to achieve overall good performance. Though real-time collision-free navigation and control of mobile robots, surface robots, and underwater robots have been studied for many years, there is a lack of research for aerial robots based on bio-inspired neurodynamics models. The future research is to incorporate bio-inspired neurodynamics models with other useful algorithms for aerial robot navigation.

- Most studies on the navigation and control of robots fail not to consider the teleoperation and telepresence issues. It is assumed that the robot works without human interactions. New approaches to telerobotic operations and human-robot interactions would be developed based on biologically inspired intelligence to outperform existing technologies. The future developed algorithms will not directly mimic any biological systems. The infusion of "human-like" and biological intelligence into robotic systems is the crux of future research.

6. CONCLUSION

Biologically inspired intelligence has been explored and studied for decades in the field of robotics. The researchers have been trying to replicate or transfer the biological intelligence to robotic systems for empowering the robots stability, adaptability, and cooperativeness. This paper provides a comprehensive survey of the research on bio-inspired neurodynamics models and their applications to path planning and control of autonomous robots. Among all bio-inspired neurodynamics models, shunting models, additive models, and gated dipole models were further elaborated. As for path planning, a bio-inspired neural network was elaborated for the dynamic collision-free path generation for many robotic systems. There are several key points are worth to highlight about bio-inspired neurodynamics models to real-time collision-free path planning.

- The fundamental concept of the neurodynamics-based path planning approach is to develop a one-to-one correspondence neural network, which is called the bio-inspired neural network, to represent the work environment. The neural activity is a continuous signal with both upper and lower bounds.
- The bio-inspired neural network is able to guide the robot to avoid the local minima points and the deadlock situations. The target globally influences the whole work space through neural activity propagation to all directions in the same manners.
- The bio-inspired neural network is able to generate the path without explicitly searching over the free work space or the collision paths, without explicitly optimizing any global cost functions, without any prior knowledge of the dynamic environment, and without any learning procedures.
- The bio-inspired neural network is able to perform properly in an arbitrarily dynamic environment, even with sudden environmental changes, such as suddenly adding or removing obstacles or targets. The obstacles have only local effects to push the robot to avoid collisions.

As for the bio-inspired robot control, several key points are worth to note:

- The neural activity is bounded between the $[-D, B]$ region with different inputs, which is the fundamental concept of the bio-inspired backstepping control.
- The bio-inspired backstepping control provides a smooth velocity curve, which is crucial to ensure the control effectiveness and efficiency
- The speed jump in conventional backstepping control design is eliminated by replacing the tracking error term with shunting model, this modification allows a wider application of the bio-inspired backstepping control in robotics.
- The excellent feasibility of the bio-inspired backstepping control allows it compatible with many other control strategies to form new hybrid control strategies for robots working in various working environments.

The current challenges and future works are the development of original and innovative new intelligent navigation, cooperation, and communication strategies, algorithms and technologies with consideration of uncertainties, disturbance and noise issues, communication issues, and human-robot interaction issues for robots in changing complex situations.

DECLARATIONS

Authors' contributions

Made substantial contributions to the research and investigation process, reviewed and summarized the literature, wrote and edited the original draft: Li J, Xu Z, Zhu D

Made substantial contributions to review and summarize the literature: Dong K, Yan T, Zeng Z

Performed oversight and leadership responsibility for the research activity planning and execution as well as developed ideas and provided critical review, commentary and revision: Yang SX

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Bekey GA. Autonomous robots: from biological inspiration to implementation and control. Boston: MIT press; 2005.
2. Li J, Yang SX, Xu Z. A survey on robot path planning using bio-inspired algorithms. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO); 2019 Dec 6-8; Dali, China. IEEE; 2019. pp. 2111–16.
3. Pradhan B, Nandi A, Hui NB, Roy DS, Rodrigues JJPC. A novel hybrid neural network-based multirobot path planning with motion coordination. *IEEE Trans Veh Technol* 2020;69:1319–27.
4. Huang HC. SoPC-based parallel ACO algorithm and its application to optimal motion controller design for intelligent omnidirectional mobile robots. *IEEE Trans Industr Inform* 2013;9:1828–35.
5. Roberge V, Tarbouchi M, Labonte G. Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans Aerosp Electron Syst* 2018;54:2105–17.
6. Hu E, Yang SX, Chiu DKY. A non-time based tracking controller for multiple nonholonomic mobile robots. In: Proceedings 2002 IEEE International Conference on Robotics and Automation; 2002 May 11-15 ; Washington, USA. IEEE; 2002. pp. 3954–59.
7. Huan TT, Kien CV, Anh HPH, Nam NT. Adaptive gait generation for humanoid robot using evolutionary neural model optimized with modified differential evolution technique. *Neurocomputing* 2018;320:112–20.
8. Guo K, Pan Y, Yu H. Composite learning robot control with friction compensation: a neural network-based Approach. *IEEE Trans Ind Electron* 2019;66:7841–51.
9. Zhang Z, Yan Z. A varying parameter recurrent neural network for solving nonrepetitive motion problems of redundant robot manipulators. *IEEE Trans Control Syst Technol* 2019;27:2680–87.
10. Hu Y, Yang SX. A knowledge based genetic algorithm for path planning of a mobile robot. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04; 2004 Apr 26- May 1; New Orleans, USA. vol. 5. IEEE; 2004. pp. 4350–55.
11. Zeng Y, Li J, Yang S, Ren E. A bio-inspired control strategy for locomotion of a quadruped robot. *Applied Sciences* 2018;8:56.
12. Grossberg S. Contour enhancement, short term memory, and constancies in reverberating neural networks. *Stud Appl Math* 1973;52:213–57.
13. Yang SX, Meng M. Neural network approaches to dynamic collision-free trajectory generation. *IEEE Trans Syst Man Cybern B Cybern* 2001;31:302–18.
14. Yang SX, Zhu A, Yuan G, Meng MQ. A bioinspired neurodynamics-based approach to tracking control of mobile robots. *IEEE Trans*

Consum Electron 2012;59:3211–20.
15. Yang SX, Meng MQH. Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach. *IEEE Trans Neural Netw* 2003;14:1541–52.
16. Zhu A, Yang SX. Path planning of multi-robot systems with cooperation. In: Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium; 2003 Jul 16-20; Kobe, Japan. vol. 2. IEEE; 2003. pp. 1028–33.
17. Pan L, Yang SX. An electronic nose network system for online monitoring of livestock farm odors. *IEEE ASME Trans Mechatron* 2009;14:371–76.
18. Martynenko AI, Yang SX. Biologically inspired neural computation for ginseng drying rate. *Biosyst Eng* 2006;95:385–96.
19. Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* 1952;117:500–544.
20. Cohen MA, Grossberg S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans Syst Man Cybern B Cybern* 1983;SMC-13:815–26.
21. Grossberg S. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks* 1988;1:17–61.
22. Öğmen H, Gagné S. Neural models for sustained and ON-OFF units of insect lamina. *Biol Cybern* 1990;63:51–60.
23. Öğmen H, Gagné S. Neural network architectures for motion perception and elementary motion detection in the fly visual system. *Neural Networks* 1990;3:487–505.
24. Yang SX, Hu E. Real-time path planning and tracking control using a neural dynamics based approach. *IFAC Proceedings Volumes* 2002;35:103–8.
25. Ni J, Wu L, Shi P, Yang SX. A dynamic bioinspired neural network based real-time path planning method for autonomous underwater Vehicles. *Comput Intel Neurosc* 2017;2017:1–16.
26. Ni J, Yang X, Chen J, Yang SX. Dynamic bioinspired neural network for multi-robot formation control in unknown environments. *Int J Rob Autom* 2015;30.
27. Oh H, Shirazi AR, Sun C, Jin Y. Bio-inspired self-organising multi-robot pattern formation: a review. *Robot Auton Syst* 2017;91:83–100.
28. Yang SX, Zhu A, Meng MQH. Biologically inspired tracking control of mobile robots with bounded accelerations. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04; 2004 Apr 26 -May 1; New Orleans,USA. IEEE; 2004. pp. 1610–15.
29. Yang SX, Meng M. An efficient neural network approach to dynamic robot motion planning. *Neural Networks* 2000;13:143–48.
30. Yang SX, Luo C. Neural dynamics and computation for navigation of multiple robots. In: IEEE International Conference on Systems, Man and Cybernetics; 2002 Oct 6-9 ; Yasmine Hammamet, Tunisia. IEEE; 2002. pp. 515–20.
31. Yang SX, Meng M, Li H. A neural computation model for real-time collision-free robot navigation. *IFAC Proceedings Volumes* 2002;35:323–28.
32. Yang X, Meng M. An efficient neural network model for path planning of car-like robots in dynamic environment. In: Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411);1999 May 9-12; Edmonton,Canada. IEEE; 1999. pp. 1374–79.
33. Yang SX, Meng M, Yuan X. A biological inspired neural network approach to real-time collision-free motion planning of a nonholonomic car-like robot. In: Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113); 2000 Oct 31-Nov 5; Takamatsu, Japan. IEEE; 2000. pp. 239–44.
34. Yuan X, Yang SX. Virtual assembly with biologically inspired intelligence. *IEEE Trans Syst Man Cybern, Part C(Appl rev)* 2003;33:159–67.
35. Luo M, Hou X, Yang SX. A multi-scale map method based on bioinspired neural network algorithm for robot path planning. *IEEE Access* 2019;7:142682–91.
36. Ni J, Li X, Hua M, Yang SX. Bioinspired neural network-based Q-learning approach for robot path planning in unknown environments. *Int J Rob Autom* 2016;31:464–74.
37. Ni J, Li X, Fan X, Shen J. A dynamic risk level based bioinspired neural network approach for robot path planning. In: 2014 World Automation Congress (WAC); 2014 Aug 3-7; Waikoloa, USA. IEEE; 2014. pp. 829–33.
38. Chen Y, Xu W, Li Z, et al. Safety-enhanced motion planning for flexible surgical manipulator using neural dynamics. *IEEE Trans Control Syst Technol* 2017;25:1711–23.
39. Yang X, Meng M. A neural network approach to real-time path planning with safety consideration. In: SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics; 1998 Oct 14-14; San Diego, USA. IEEE; 1998. pp. 3412–17.
40. Yang SX, Meng M. An efficient neural network method for real-time motion planning with safety consideration. *Robot Auton Syst* 2000;32:115–28.
41. Glasius R, Komoda A, Gielen SCAM. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks* 1995;8:125–33. [DOI: 10.1016/0893-6080(94)e0045-m]

42. Sun B, Zhu D, Tian C, Luo C. Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming. *IEEE Trans Cogn Commun Netw* 2019;11:73–84.
43. Chen M, Zhu D. Multi-AUV cooperative hunting control with improved Glasius bio-inspired neural network. *J Navig* 2018;72:759–76.
44. Chen M, Zhu D. Real-time path planning for a robot to track a fast moving target based on improved Glasius bio-inspired neural networks. *Int J Intell Robot Appl* 2019;3:186–95.
45. Willms AR, Yang SX. An efficient dynamic system for real-time robot-path planning. *IEEE Trans Syst Man Cybern B Cybern* 2006;36:755–66.
46. Willms AR, Yang SX. Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance. *IEEE Trans Syst Man Cybern B Cybern* 2008;38:884–93.
47. Li S, Meng MQH, Chen W, et al. SP-NN: A novel neural network approach for path planning. In: 2007 IEEE Interna- tional Conference on Robotics and Biomimetics (ROBIO); 2007 Dec 15-18; Sanya, China. IEEE; 2007. pp. 1355–60.
48. Qu H, Yang SX, Willms AR, Yi Z. Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Trans Neural Netw* 2009;20:1724–39.
49. Qu H, Yi Z, Yang SX. Efficient shortest-path-tree computation in network routing based on pulse-coupled neural networks. *IEEE Trans Cybern* 2013;43:995–010.
50. Zhong Y, Shirinzadeh B, Tian Y. A new neural network for robot path planning. In: 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics; 2008 July 2-5; Xi'an, China. IEEE; 2008. pp. 1361–66.
51. Chen Y, Liang J, Wang Y, et al. Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics. *Soft Comput* 2020;24:13979–95.
52. Bueckert J, Yang SX, Yuan X, Meng MQH. Neural dynamics based multiple target path planning for a mobile robot. In: 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO); 2007 Dec 5-18; Sanya, China. IEEE; 2007. pp. 1047–52.
53. Li H, Yang SX, Biletskiy Y. Neural network based path planning for a multi-robot system with moving obstacles. In: 2008 IEEE International Conference on Automation Science and Engineering; 2008 Aug 23-26; Arlington, USA. IEEE; 2008. pp. 410–19.
54. Yuan X, Yang SX. Multirobot-based nanoassembly planning with automated path generation. *IEEE ASME Trans Mechatron* 2007;12:352–56.
55. Zhu A, Cai G, Yang SX. Theoretical analysis of a neural dynamics based model for robot trajectory generation. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions; 2002 Jun 29-Jul 1; Chengdu, China. IEEE; 2002. pp. 1184–88.
56. Ni J, Yang SX. Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments. *IEEE Trans Neural Netw* 2011;22:2062–77.
57. Yang SX, Luo C. A neural network approach to complete coverage path planning. *IEEE Trans Syst Man Cybern B Cybern* 2004;34:718–24.
58. Godio S, Primatesta S, Guglieri G, Dovis F. A bioinspired neural network-based approach for cooperative coverage planning of UAVs. *Information* 2021;12:51.
59. Luo C, Yang SX, Yuan X. Real-time area-covering operations with obstacle avoidance for cleaning robots. In: IEEE/RSJ International Conference on Intelligent Robots and System; 2002 Sept 30-Oct 4; Lausanne, Switzerland. IEEE; 2002. pp. 2359–64.
60. Yang SX, Luo C, Meng M. A neural computational algorithm for coverage path planning in changing environments. In: IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions; 2002 Jun 29-Jul 1; Chengdu, China. IEEE; 2002. pp. 1174–78.
61. Luo C, Yang SX. A real-time cooperative sweeping strategy for multiple cleaning robots. In: Proceedings of the IEEE Internatinal Symposium on Intelligent Control; 2002 Oct 30-30; Vancouver, Canada. IEEE; 2002. pp. 660–65.
62. Zhang J, Lv H, He D, et al. Discrete bioinspired neural network for complete coverage path planning. *Int J Rob Autom* 2017;32.
63. Luo C, Yang SX. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Trans Neural Netw* 2008;19:1279–98.
64. Luo C, Yang SX, Meng MQH. Real-time map building and area coverage in unknown environments. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation; 2005 Apr 18-22; Barcelona, Spain. IEEE; 2005. pp. 1736–41.
65. Luo C, Yang S, Meng M. Neurodynamics based complete coverage navigation with real-time map building in unknown environments. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2006 Oct 9-15; Beijing, China. IEEE; 2006. pp. 4228–33.
66. Luo C, Yang SX, Li X, Meng MQH. Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots. *IEEE Trans Consum Electron* 2017;64:750–60.
67. Yu Z, Tao J, Xiong J, Luo A, Yang SX. Neural-dynamics-based path planning of a bionic robotic Fish. In: 2019 IEEE Interna-

tional Conference on Robotics and Biomimetics (ROBIO);2019 Dec 6-8; Dali, China. IEEE; 2019. pp. 1803–8.
68. Yu Z, Tao J, Xiong J, Yang SX. Design and analysis of path planning for robotic fish based on neural dynamics model. *Int J Rob Autom* 2021;36.
69. Yan M, Zhu D, Yang SX. A novel 3-D bio-inspired neural network model for the path planning of an AUV in underwater environments. *Intelligent Automation Soft Computing* 2013;19:555–66.
70. Zhu D, Yang SX. Bio-inspired neural network-based optimal path planning for UUVs under the effect of ocean currents. *IEEE Trans Veh Technol* 2021:1–1.
71. Zhu D, Li W, Yan M, Yang SX. The path planning of AUV based on D-S information fusion map building and bio-inspired neural network in unknown dynamic environment. *Int J Adv Robot Syst* 2014;11:34.
72. Cao X, Peng J. A potential field bio-inspired neural network control algorithm for AUV path planning. In: 2018 IEEE International Conference on Information and Automation (ICIA); 2018 Aug 11-13; Fujian, China. IEEE; 2018. pp. 1427–32.
73. Cao X, Chen L, Guo L, Han W. AUV global security path planning based on a potential field bio-Inspired neural network in underwater environment. *Intelligent Automation & Soft Computing* 2021;27:391–407.
74. Zhu A, Yang SX. A neural network approach to dynamic task assignment of multirobots. *IEEE Trans Neural Netw* 2006;17:1278–87.
75. Zhu A, Yang SX. An improved SOM-based approach to dynamic task assignment of multi-robots. In: 2010 8th World Congress on Intelligent Control and Automation; 2010 Jul 7-9; Jinan, China. IEEE; 2010. pp. 2168–73.
76. Yi X, Zhu A, Yang SX, Luo C. A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments. *IEEE Trans Cybern* 2017;47:974–83.
77. Zhu D, Huang H, Yang SX. Dynamic task assignment and path planning of multi-AUV system based on animproved self-organizing map and velocity synthesis method in three-dimensional underwater workspace. *IEEE Trans Cybern* 2013;43:504–14.
78. Huang H, Zhu D, Yuan F. Dynamic task assignment and path planning for multi-AUV system in 2D variable ocean current environment. In: 2012 24th Chinese Control and Decision Conference (CCDC); 2012 May 23-25; Taiyuan, China. IEEE; 2012. pp. 999–012.
79. Zhu D, Cao X, Sun B, Luo C. Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Trans Cogn Commun Netw* 2018;10:304–13.
80. Cao X, Zhu D. Multi-AUV task assignment and path planning with ocean current based on biological inspired self-organizing map and velocity synthesis algorithm. *Intelligent Automation & Soft Computing* 2015;23:31–39.
81. Zhu D, Zhou B, Yang SX. A novel algorithm of multi-AUVs task assignment and path planning based on biologically inspired neural network map. *IEEE Trans Hum Mach Syst* 2021;6:333–42.
82. Rui Z, Zhu D. Cooperative search algorithm For AUVs based on bio-inspired model. In: The 26th Chinese Control and Decision Conference (2014 CCDC); 2014 May 31-Jun 2; Changsha, China. IEEE; 2014. pp. 4569–74.
83. Cao X, Zhu D, Yang SX. Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments. *IEEE Trans Neural Netw Learn Syst* 2016;27:2364–74.
84. Cao X, Zhu D. Multi-AUV underwater cooperative search algorithm based on biological inspired neurodynamics model and velocity synthesis. *J Navig* 2015;68:1075–87.
85. Huang Z, Zhu D. A cooperative hunting algorithm of multi-AUV in 3-D dynamic environment. In: The 27th Chinese Control and Decision Conference (2015 CCDC); 2015 May 23-25; Qingdao, China. IEEE; 2015. pp. 2571–75.
86. Zhu D, Lv R, Cao X, Yang SX. Multi-AUV hunting algorithm based on bio-inspired neural network in unknown environments. *Int J Adv Robot Syst* 2015;12:166.
87. Cao X, Huang Z, Zhu D. AUV cooperative hunting algorithm based on bio-inspired neural network for path conflict state. In: 2015 IEEE International Conference on Information and Automation; 2015 Aug 8-10; Lijang, China. IEEE; 2015. pp. 1821–26.
88. Yang SX, Yuan G, Meng M, Mittal GS. Real-time collision-free path planning and tracking control of a nonholonomic mobile robot using a biologically inspired approach. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation; 2001 May 21-26 ; Seoul, Korea (South). vol. 4. IEEE; 2001. pp. 3402–7.
89. Yuan G, Yang SX, Mittal GS. Tracking control of a mobile robot using a neural dynamics based approach. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation; 2001 May 21-26 ; Seoul, Korea (South). IEEE; 2001. pp. 163–68.
90. Zheng W, Wang H, Zhang Z, Wang H. Adaptive robust finite-time control of mobile robot systems with unmeasurable angular velocity via bioinspired neurodynamics approach. *Eng Appl Artif Intell* 2019;82:330–44.
91. Hu Y, Yang SX. A fuzzy neural dynamics based tracking controller for a nonholonomic mobile robot. In: Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003); 2003 Jul 20-24 Kobe, Japan. IEEE; 2003. pp. 205–10.
92. Zhang HD, Liu SR, Yang SX. A neurodynamics based neuron-PID controller and its application to inverted pendulum. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826); 2004 Aug 26-29; Shanghai, China. IEEE; 2004. pp. 527–32.

93. Li H, Yang SX, Karray F. Optimization of a neural dynamics based controller for a nonholonomic mobile robot using genetic algorithms. In: The Fourth International Conference on Control and Automation, 2003. ICCA; 2003 Jun 12-12; Montreal,Canada. IEEE; 2003. pp. 911–16.
94. Yang SX, Yang H, Meng MQH. Neural dynamics based full-state tracking control of a mobile robot. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04; 2004 Apr 26- May 1; New Orleans, USA. IEEE; 2004. pp. 4614–19.
95. Xu Z, Yang SX, Gadsden SA. Enhanced bioinspired backstepping control for a mobile robot with unscented kalman filter. *IEEE Magazines and Online Publications* 2020;8:125899–908.
96. Pan CZ, Lai XZ, Yang SX, Wu M. A biologically inspired approach to tracking control of underactuated surface vessels subject to unknown dynamics. *Expert Syst Appl* 2015;42:2153 – 61.
97. Mohd Shamsuddin BPNF, Bin Mansor MA. Motion cntrol algorithm for path following and trajectory tracking for unmanned surface vehicle: a review paper. In: 2018 3rd International Conference on Control, Robotics and Cybernetics (CRC). Proceedings. Piscataway, NJ, USA; 2018. pp. 73 – 77.
98. Pan CZ, Lai XZ, Yang SX, Wu M. Backstepping neurodynamics based position-tracking control of underactuated autonomous surface vehicles. In: 2013 25th Chinese Control and Decision Conference (CCDC); 2013 May 25-27; Guiyang, China. IEEE; 2013. pp. 2845–50.
99. Pan C, Lai X, Yang SX, Wu M. A bioinspired neural dynamics-based approach to tracking control of autonomous surface vehicles subject to unknown ocean currents. *Neural Comput Appl* 2015;26:1929–38.
100. Li D, Wang P, Du L. Path planning technologies for autonomous underwater vehicles-a review. *IEEE Access* 2019;7:9745 – 9768.
101. Burdinsky IN. Guidance algorithm for an autonomous unmanned underwater vehicle to a given target. *Optoelectron Instrum Data Process* 2012;48:69 – 74.
102. Karkoub M, Wu HM, Hwang CL. Nonlinear trajectory-tracking control of an autonomous underwater vehicle. *Ocean Eng* 2017;145:188–98.
103. Zhu D, Hua X, Sun B. A neurodynamics control strategy for real-time tracking control of autonomous underwater vehicle. J Navig 2013 aug;67:113–27.
104. Sun B, Zhu D, Ding F, Yang SX. A novel tracking control approach for unmanned underwater vehicles based on bio-inspired neurodynamics. *IJ Mar Sci Tech-japan* 2012;18:63–74.
105. Sun B, Zhu D, Yang SX. A bioinspired filtered backstepping tracking control of 7000-m manned submarine vehicle. *IEEE Trans Ind Electron* 2014;61:3682–93.
106. Jiang Y, Guo C, Yu H. Robust trajectory tracking control for an underactuated autonomous underwater vehicle based on bioinspired neurodynamics. *Int J Adv Robot Syst* 2018;15:172988141880674.
107. Peng Z, Wen G, Rahmani A, Yu Y. Leader–follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach. *Robot Auton Syst* 2013;61:988–96.
108. Yi G, Mao J, Wang Y, Zhang H, Miao Z. Neurodynamics-based leader-follower formation tracking of multiple nonholonomic vehicles. *Assembly Autom* 2018;38:548–57.
109. He Y, Mou J, Chen L, et al. Survey on hydrodynamic effects on cooperative control of Maritime Autonomous Surface Ships. *Ocean Eng* 2021;235.
110. Peng Z, Wang J, Wang D, Han QL. An overview of recent advances in coordinated control of multiple autonomous surface vehicles. *IEEE Trans Industr Inform* 2021;17:732-45.
111. Wang D, Fu M. Adaptive formation control for waterjet USV with input and output constraints based on bioinspired neurodynamics. *IEEE Access* 2019;7:165852–61.
112. Wang D, Ge SS, Fu M, Li D. Bioinspired neurodynamics based formation control for unmanned surface vehicles with line-of-sight range and angle constraints. *Neurocomputing* 2021;425:127–34.
113. Yang Y, Xiao Y, Li T. A survey of autonomous underwater vehicle formation: performance, formation control, and communication capability. *IEEE Commun Surv Tutor* 2021;23:815-41.
114. Hadi B, Khosravi A, Sarhadi P. A review of the path planning and formation control for multiple autonomous underwater vehicles. *J Intel Robot Syst* 2021;101.
115. Ding G, Zhu D, Sun B. Formation control and obstacle avoidance of multi-AUV for 3-D underwater environment. In: Proceedings of the 33rd Chinese Control Conference; 2014 Jul 28-30 ;Nanjing, China. IEEE; 2014. pp. 8347–52.

Li *et al. Intell Robot* 2021;1(1):84-98
DOI: 10.20517/ir.2021.06

Intelligence & Robotics

Research Article

Open Access

Unsupervised monocular depth estimation with aggregating image features and wavelet SSIM (Structural SIMilarity) loss

Bingen Li[1], Hao Zhang[1], Zhuping Wang[1], Chun Liu[2], Huaicheng Yan[3,4], Lingling Hu[1]

[1]Department of Control Science and Engineering, Tongji University, Shanghai 200000, China.
[2]the College of Surveying and Geo-informatics, Tongji University, Shanghai 200000, China.
[3]East China University of Science and Technology, Shanghai 200000, China.
[4]College of Mechatronics and Control Engineering, Hubei Normal University, Huangshi 435000, China.

Correspondence to: Dr. Hao Zhang, Department of Control Science and Engineering, Tongji University, Shanghai 200000, China. E-mail: zhang_hao@tongji.edu.cn

How to cite this article: Li B, Zhang H, Wang Z, Liu C, Yan H, Hu L. Unsupervised monocular depth estimation with aggregating image features and wavelet SSIM (Structural SIMilarity) loss. *Intell Robot* 2021;1(1):84-98. http://dx.doi.org/10.20517/ir.2021.06

Received: 27 Aug 2021 **First Decision:** 4 Sep 2021 **Revised:** 14 Sep 2021 **Accepted:** 15 Sep 2021 **Published:** 12 Oct 2021

Academic Editor: Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

Unsupervised learning has shown to be effective for image depth prediction. However, the accuracy is restricted because of uncertain moving objects and the lack of other proper constraints. This paper focuses on how to improve the accuracy of depth prediction without increasing the computational burden of the depth network. Aggregated residual transformations are embedded in the depth network to extract high-dimensional image features. A more accurate mapping relationship between feature map and depth map can be built without bringing extra network computational burden. Additionally, the 2D discrete wavelet transform is applied to the structural similarity loss (SSIM) to reduce the photometric loss effectively, which can divide the entire image into various patches and obtain high-quality image information. Finally, the effectiveness of the proposed method is demonstrated. The training model can improve the performance of the depth network on the KITTI dataset and decrease the domain gap on the Make3D dataset.

Keywords: Unsupervised depth estimation, computational complexity, aggregated residual transformations, 2D discrete wavelet transform

1. INTRODUCTION

Predicting depth from a single 2D image is a fundamental task in computer vision. It has been studied for many years with widespread applications in reality, such as visual navigation[1], object tracking[2,3], and surgery[4]. Moreover, accurate depth information is vital with considerable influence on the performance of autonomous driving, where expensive laser sensors are usually used. Recent advances in convolutional neural networks (CNNs) show their powerful ability to learn an image's high-dimensional features. Especially, the mapping relationship between image feature and image depth can be built. Generally, monocular depth estimation approaches can be classified into three categories: supervised[5–9], semi-supervised[10], and unsupervised[11–19]. Both supervised and semi-supervised learning rely on the image depth ground truth. Using a laser sensor to obtain the depth ground truth of many images is expensive and difficult. However, unsupervised learning has the advantage of eliminating the dependency on the depth ground truth. Therefore, more and more studies are training monocular depth estimation networks using unsupervised methods from monocular images or stereo pairs. Compared with stereo pairs, a monocular dataset is more general as the input of network. However, it needs to estimate the pose transformation between consecutive frames simultaneously. As a result, a pose estimation network is necessary that outputs relative 6-DoF pose with given sequences of frames as input.

Most unsupervised depth estimation networks[5,8,11] are constructed using typical CNN structures. On the one hand, a series of max-pooling and stride operations may reduce the network's ability to learn image features and cause lower quality of depth map. On the other hand, to improve the performance of the network, deeper convolution layers are designed in depth CNNs. They increase the computational burden of the network and bring extra hardware cost. In most cases, the cost of the network overweighs the benefits generated by the network. To improve the depth estimation performance without increasing the network burden, an end-to-end unsupervised monocular depth network framework is proposed in this paper. Inspired by previous work[20] on the image classification task, aggregated residual transformations (ResNeXt) are migrated to the depth estimation field. Based on typical depth CNNs, the ResNeXt block is embedded to extract more delicate image features in the encoder network. In addition, more accurate mapping relationship between the feature map and depth map can be built without bringing extra network burden. In addition, the accuracy of depth network suffers from some noise (*e.g.*, haze and rain) in the complex images. To reduce the influence of noise, the 2D wavelet discrete transform[21] is applied to SSIM loss, which can recover high-quality clear images. A sample of depth prediction is shown in Figure 1.

In summary, our proposed network can improve depth prediction accuracy without increasing network computational complexity. The contributions of this paper can be summarized as follows:

(1) Based on a ResNeXt block, a novel feature extraction module for depth network is developed to improve the accuracy of depth prediction. It can not only extract high-dimensional image features but also guide the network to more deeply learn the scene to get farther pixel depth.

(2) A wavelet SSIM loss is applied to photometric loss to converge the training network. Various patches with clearer image information computed by DWT are used as input, rather than the whole image, to the loss function, which can remove some noise (daze, rain, *etc.*) from the image.

The rest of this paper is organized as follows. The related work on depth estimation is discussed in Section 2. Section 3 presents an overview of the proposed network architecture and the loss function. Then, some

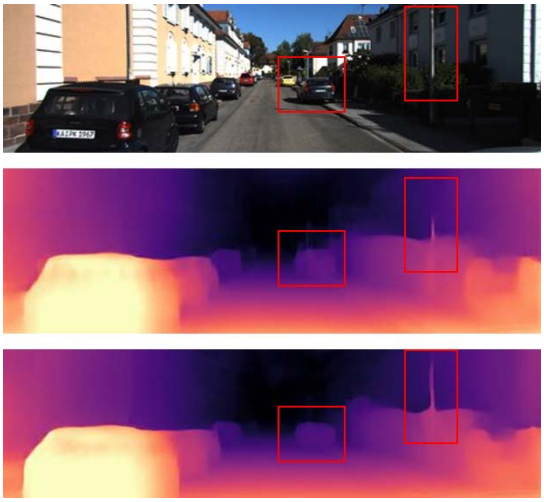


Figure 1. The input image from the KITTI dataset (top); the baseline MonoDepth2 [22] (M, ResNet50, without pre-training) depth prediction; (middle) and our result (bottom).

experiments based on different datasets are presented to verify the performance of the proposed network in Section 4. Finally, the conclusions and future work are introduced in Section 5.

2. RELATED WORK

2.1. Supervised depth estimation

Based on vast training datasets with depth ground truth, depth estimation networks show great performance in recent years. Eigen *et al.* [5] first demonstrated the huge potential of CNNs in depth prediction from a single image. They obtained reliable depth estimation results by using a coarse-to-fine depth network. Further, Liu *et al.* [7] combined CNNs with Markov random fields (MRF) to learn intermediate features, acquiring clearer local details of depth map in the visual effect. Laina *et al.* [8] changed the structure of the depth network and proposed a residual CNNs to model the mapping relationship between monocular image and its corresponding depth map. Instead of using absolute depth ground truth, Chen *et al.* [9] acquired relative depth value labels between the random pixel pairs from the image to train the depth network. In addition, to obtain dense depth map, Kuznietsov *et al.* [10] proposed a semi-supervised method which used both sparse ground truth depth for supervised learning and a photo consistent loss in stereo images for unsupervised learning.

Even though the works mentioned above significantly contributed to depth estimation, these methods still suffer from the limitation of depth ground truth.

2.2. Unsupervised depth estimation

Based on stereo or monocular images, unsupervised learning methods focus on how to design the supervisory signal. The typical solution is to use view synthesis as a proxy task [11,12,14–24], so as to get rid of depth ground truth.

2.2.1. Unsupervised depth estimation from stereo images

Using stereo images is a feasible unsupervised way to train a monocular depth network. A depth network can be obtained by predicting the left–right pixel disparities between stereo pairs during training. It can be

applied when predicting monocular image depth. Garg *et al.*[11] first used stereo pairs to train depth network with known disparities between left and right images and acquired great performance. Inspired by the authors of[11], Godard *et al.*[12] designed a novel loss function which enforced both left-right and right–left disparities consistency produced from stereo images[12]. Zhan *et al.*[13] extended the stereo-based network architecture by increasing the visual odometry network (VO). The performance of Zhan's network was superior to other unsupervised methods at that time. To recover absolute scale depth map from stereo pairs, Li *et al.*[14] proposed a visual odometry system (UnDeepVO), which was capable of estimating the 6-DoF camera pose and recovering the absolute depth value.

2.2.2. Unsupervised depth estimation from monocular images

For monocular depth estimation, it is necessary to design an extra pose network to obtain pose transformation between consecutive frames. Both depth and pose networks are trained together with loss function. Zhou *et al.*[16] pioneered the training of depth networks with monocular video. They proposed two separate networks (SfMLearner) to learn image depth and inter-frame pose transformation. However, the accuracy of the depth network was often limited by the influence of moving objects and occlusion. Their work motivated some researchers to consider these shortcomings. Subsequently, Casser *et al.*[17] developed a separate network (struct2depth) to learn each moving object motion, but their work was based on the condition that the number of moving objects needed to be hypothesized in advance. In addition, researchers found that the optical flow method could be employed to deal with moving object motion. Yin *et al.*[18] developed a cascading network framework (GeoNet) to adaptively learn rigid and non-rigid object motion. Recently, multi-task training methods have been proposed. Luo *et al.*[19] intended to train depth, camera pose, and optical flow networks (EPC++) jointly with 3D holistic understanding. Similarly, Ranjan *et al.*[24] proposed a competitive collaboration mechanism (CC) with depth, camera motion, optical flow, and motion segmentation together. Both Luo and Ranjan's joint network inevitably increased the difficulty of the training network and the computational burden of the network.

From the above works, we can see that most studies aim to improve the accuracy of the depth network by changing the network structure or building robust supervisory signal. It is worth noting that these methods bring network complexity and computational burden while improving the network accuracy. This motivates us to study how to balance both sides. Poggi *et al.*[15] presented an effective pyramid feature extraction network, which can be implemented in real-time on CPU. However, the accuracy of the network cannot satisfy the requirements of practical applications. Xie *et al.*[20] provided a template with aggregated residual transformations (ResNeXt), which achieved a better classification result without increasing network computation. Because of the advantages of ResNeXt, we apply it to the image depth prediction field. The ResNeXt block serves as a feature extraction module of the depth network to learn the image's high-dimensional features. The proposed approach is not only independent of depth ground truth, but also does not increase computational burden.

3. METHOD

The proposed method contains two parts: an end-to-end network framework and a loss function. The network framework consists of a depth network and a pose network, as shown in Figure 2. Given unlabeled monocular sequences, the depth network outputs the predicted depth map, while the pose network outputs the 6-DoF relative pose transformation between adjacent frames. The loss function is made up of the basic photometric loss and the depth smoothness loss, and it couples both networks into the end-to-end network.

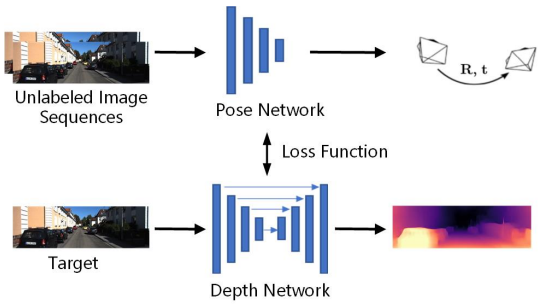


Figure 2. The overall architecture of both the depth network and the pose network.

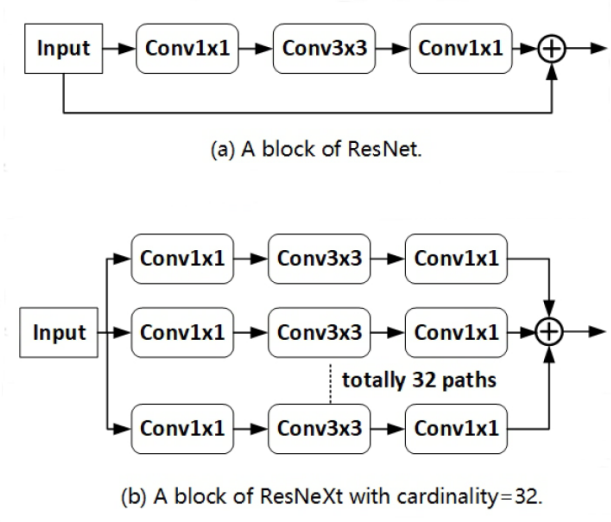


Figure 3. The architecture of ResNet and ResNeXt block: (a) the ResNet block; and (b) the aggregated residual transformations. Both have similar complexity, but the ResNeXt block has better adaptability and expansibility.

3.1. Problem statement

The aim of the unsupervised monocular depth network is to develop a mapping relationship $\Gamma : I(p) \rightarrow D(p)$, where $I(p)$ is an arbitrary image, $D(p)$ is the predicted depth map of the image $I(p)$, and p is per pixel in the image $I(p)$. Establishing a more accurate mapping function Γ is considered in this paper, which includes: (a) a simple and effective network pipeline without increasing network computational complexity; and (b) a high-quality depth map $D(p)$ with subtle details for a given input image $I(p)$.

For Item (a), our focus is to change the basic building blocks of the depth CNN structure using aggregated residual transformations (ResNeXt). In the depth network, ResNeXt serves as feature extraction module to learn the image's high-dimensional features without increasing network computational burden. For Item (b), low-texture regions in the low-scale depth map are weakened, bringing inaccurate image reconstruction. Inspired by the authors of[22], four images with full resolution are reconstructed instead of building four images with different resolutions. Before the four images are reconstructed, the predicted four-scale depth map needs to be resized to the same resolution as input image with bilinear interpolation.

A single image $I(p)$ is considered as the input of the depth network. The designed depth network outputs five-scale feature map $F_{k\times}(k \in 1, 2, 3, 4, 5)$ in the encoder network and four-scale depth map D_n in the decoder

network. The mapping function is designed as

$$D_{n\times}(I(p)) = \Gamma_n((F_{1\times}(I(p), \ldots, (F_{m\times}(I(p)))) \tag{1}$$

where m denotes the number of feature maps, $m = 5$. n represents the scale factor of depth map, $n \in 0, 1, 2, 3$. k denotes the resolution of feature map $F_{k\times}$ is $1/2^k$ of the input resolution.

Then, bilinear interpolation is applied to each predicted depth map $D_{n\times}$ to acquire the full-resolution depth map $R(I(p))$, which is defined as follows:

$$R(I(p)) = UD_{n\times}(I(p)) \tag{2}$$

where U represents bilinear interpolation which recovers the resolution $1/2^n$ of $D_{n\times}$ to the input full resolution.

The full-resolution depth map $R(I(p))$ is necessary to reconstruct the input image. Given two adjacent images with a target view and a source view $\langle I_t(p), I_s(p)\rangle$, and the predicted 6-DoF pose transformation T, a pixel in the target image p_t's mapping homogeneous coordinate $p_{s\to t}$ in the source image I_s is computed as

$$p_{s\to t} \sim KT_{t\to s}R(p_t)K^{-1}P_t \tag{3}$$

where K is camera intrinsic matrix, p_t is set as the normalized coordinate in target image I_t, and $T_{t\to s}$ is a 4×4 matrix transformed by T.

Therefore, the reconstructed target image I_s^t can be obtained by Equation (3) using differentiable bilinear sampling mechanism [16] to sample the corresponding pixel $p_{s\to t}$ on the source image I_s. The reconstructed target image I_s^t is used to calculate the photometric loss in Part D.

3.2. Feature extraction module

Equation (1) is applied to exploit higher-dimensional features and acquire feature map $F_{k\times}$ with more details. Since the ResNeXt block has a great performance on classification task. the feature extraction module is constructed by the ResNeXt block. In contrast to the ResNet used in most depth CNNs, the ResNeXt block aggregates more image features without bringing more network parameters, as shown in Figure 3.

The ResNeXt block puts the input image into 32 parallel groups and learns the image features, respectively. Each group shares the same super-parameters and is designed as a bottleneck structure which cascades three convolution layers with the kernel sizes, respectively, being 1×1, 3×3, and 1×1. The first 1×1 convolution layer extracts high-dimensional abstract features by reducing (or increasing) output channels. Given an input image I with $H\times W\times C'$ resolution, the transformation function T_i of the ith group maps image I to the high-dimensional feature map $T_i(I)$. The aggregated output $f(I)$ is the summation of the output of all the groups, which is defined as follows:

$$f(I) = \sum_{i=1}^{C} T_i(I) \tag{4}$$

where C is the number of groups, $C = 32$, with C as cardinality.

Then, to be closely connected with the input, a residual operation is used, $F(I)$. The aggregated output feature

map for each module is

$$F(I) = I + \sum_{i=1}^{C} T_i(I) \tag{5}$$

3.3. Network architecture

The proposed depth estimation network employs U-Net structure including an encoder network and a decoder network. The encoder network is built by embedding the ResNeXt block[20]. It transforms the three-dimensional monocular image into multi-channel feature map. The decoder network builds the relationship between extracted feature map and the depth map by a series of upsample and convolution (Up-convolution) operations, as shown in Figure 4.

(1) To eliminate texture copy artifacts in the depth map, the Up-convolution operation[22] instead of deconvolution is used to reshape the feature map. (2) Due to max-pooling and stride operations ignoring some local features and causing some details to be lost in the depth image, skip connections are used to merge the corresponding feature maps for encoder network into decoder network and obtain fine image details. (3) Inspired by the authors of[22], we resize all depth maps to the same resolution as input using bilinear interpolation (represented by the U operation in Equation (2)).

The structure for the pose network is designed as a standard ResNet18 encoder, which is similar to the one in[22]. More input images in the pose network bring more accurate depth estimation under certain conditions. However, to reduce the number of training parameters of pose network, the pose network has N ($N = 3$) adjacent images as input. Therefore, the shape for convolutional weights in the first layer is $(3 \times N) \times 64 \times 3 \times 3$ rather than the default $3 \times 64 \times 3 \times 3$ in the pose network. The output of the pose network has $6 * (N - 1)$ channels. In addition, our pose network is trained without pre-training. All convolution layers are activated by ReLU function[25] except for the last layer. When the pose result is evaluated, an image pair is fed into pose network to produce six output channels, the first three-channel is rotation, and the last three-channel is translation.

3.4. Wavelet SSIM loss

In general, the SSIM[26] loss is included in the photometric loss to measure the degree of similarity between images. In this paper, the 2D discrete wavelet transform (DWT) is applied to SSIM to decrease the photometric loss. Firstly, The DWT divides an image into some patches with different frequencies. Then, the SSIM of each patch is computed. To preserve high-frequency image details and avoid producing "holes" or artifacts in some low-texture regions, it can flexibly adjust the weights of each patch of SSIM loss.

In the 2D discrete wavelet transform (DWT), low-pass and high-pass filters are performed on an image to obtain the convolution results. For instance, four filters, f_{LL},f_{LH}, f_{HL}, and f_{HH}, are obtained by the low-pass filter multiplying the high-pass filter. The DWT divides an image into four small patches with different frequencies through these four filters, which can remove unnecessary interference from the images ($e.g.$, haze and rain). Iteratively, the DWT can be formulated as follows:

$$I_{i+1}^{LL}, I_{i+1}^{LH}, I_{i+1}^{HL}, I_{i+1}^{HH} = DWT(I_i^{LL}) \tag{6}$$

where i is the iterative time of DWT. I_0^{LL} is the original image. In this paper, $i = 2$. I_{LL} is the down-sampling image. I_{HL} and I_{LH} are the horizontal and vertical edge detection images, respectively. I_{HH} is the corner detection image.

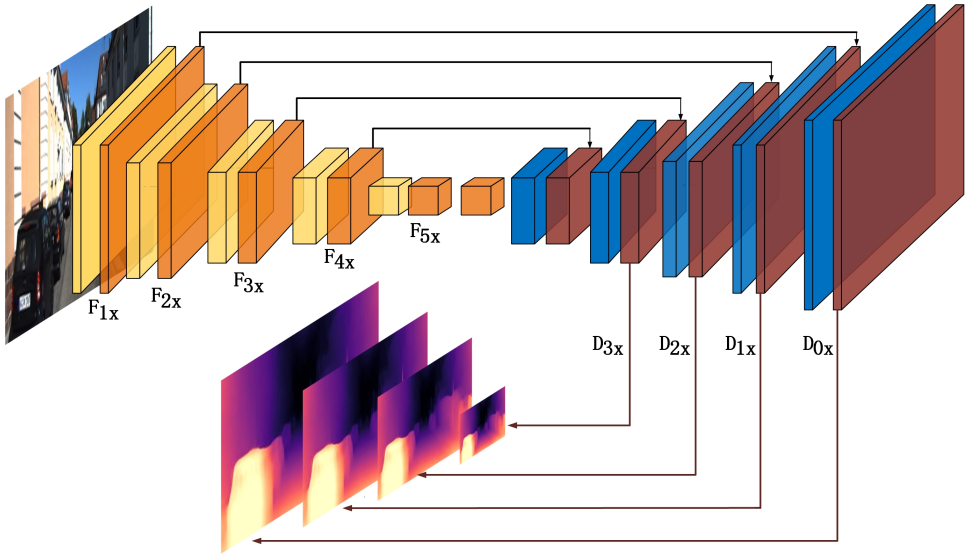


Figure 4. The proposed depth network architecture. The width and height of every cube indicates output channels, and the size is reduced by half every time. The first yellow cube is a convolution block, while the rest of the yellow cubes are ResNeXt blocks. The orange blocks represent the five-scale feature map, F_{kx}. In the decoder network, convolution layers are blue. Upsample and convolution operations are red. D_{nx} is the four-scale depth map.

To preserve high-frequency image details and avoid producing image artifacts, a coarse-to-fine manner is adopted to change the image resolution in the SSIM loss. The DWT divides the image into four patches: $I_i^{LL}, I_i^{HL}, I_i^{LH}, and I_i^{HH}$. Except the low-frequency I_i^{LL}, the SSIM loss of the other three high-frequency patches are computed. Iteratively, I_i^{LL} is divided by DWT to generate different patches to obtain the new SSIM loss. Therefore, the total wavelet SSIM (W-SSIM) loss is

$$L_{W-SSIM(t,s)} = \sum_{0}^{i} r_i L_{SSIM}(t_i^w, s_i^w), w \in \{LL, HL, LH, HH\} \tag{7}$$

The ratios of the four patches are

$$I_{LL} : I_{LH} : I_{HL} : I_{HH} = r^2 : r(1-r) : r(1-r) : (1-r)^2 \tag{8}$$

where r_i is the weight of each patch. The initial value of r is 0.7. t is the target image. s is the source image.

Initially, before the DWT divides the image, the SSIM loss between the target image and source image is calculated. The total wavelet SSIM (L_{WSSIM}) loss is

$$L_{WSSIM} = L_{SSIM}(t, s) + L_{W-SSIM} \tag{9}$$

3.5. Total loss function

There are two main parts in the loss function: the target image photometric loss L_p is calculated by reconstructing the target image, while the smoothness loss L_s of depth image compels the predicted depth map to be smooth, given the input target image I_t and its reconstructed image I_s^t. The details are shown in Equation (3). To make the photometric loss effective and meaningful, some assumptions need to be set: (1) the scenes are Lambertian; and (2) the scenes should be static and unsheltered.

In general, the image photometric loss contains the structural similarity metric (SSIM)[26] and the regularization loss ζ_1. The wavelet SSIM loss is used to replace SSIM loss in photometric loss. Therefore, the image photometric loss is defined as

$$pe = \alpha \frac{1 - L_{WSSIM}(I_t, I_s^t)}{2} + (1 - \alpha) \left\| I_t - I_s^t \right\|_1 \tag{10}$$

where we empirically set $\alpha = 0.85$.

When computing the photometric loss from different source images, most previous approaches average the photometric loss together into every available source images. However, the second assumption requests that each pixel in the target image is also visible to the source image. However, this assumption is easily broken. It is inevitable that some moving objects and occlusions exist in the scene; thus, some pixels are available in one image but are not available in the next image. As a result, inaccurate pixel reconstruction and the photometric error are caused. Following the work in[22], the minimum photometric loss at each pixel in the target image is computed instead of the average photometric loss. Note that this method can only correct the photometric loss but not eliminate it. Therefore, the final per-pixel photometric loss is

$$L_p = \min_t pe(I_t, I_s^t) \tag{11}$$

In addition, the performance of depth network suffers from the influence of moving objects in the image. These moving pixels should not be involved in computing the photometric loss. Therefore, a binary per-pixel mask μ in[22] is applied to automatically recognize moving pixels ($\mu = 0$) and static pixels ($\mu = 1$). The mask μ only includes some pixels whose photometric error of the reconstructed image I_s^t is lower than that of the target image I_t and source image I_s. The mask μ is defined as

$$\mu = [\min(pe(I_t, I_s^t)) > \min(pe(I_t, I_s))] \tag{12}$$

[ ] is the Iverson bracket. The auto-masking photometric loss[22] is

$$L_p = \mu L_p \tag{13}$$

The second-order gradients of the depth map are used to make the depth map smooth. Because the edge or corner in the depth map should be less smooth than other flat regions, the gradient of the depth map should be locally smooth rather than fully smooth. Therefore, a Laplacian[23] is applied to automatically perceive the position of each pixel. Different from the method in[23], it is used at every scale instead of a specific scale. The Laplacian template is second-order differencing with four neighborhoods. It can reinforce object edges and weaken the region of slowly varying intensity. The smoothness loss of this pixel receives a lower weight when the Laplacian is higher. The smoothness loss is defined as follows:

$$L_s = e^{-\nabla^2 I(x_i)} (|\partial_{xx} d_i| + |\partial_{xy} d_i| + |\partial_{yy} d_i|) \tag{14}$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{15}$$

where ∇ is the Laplacian operator.

Therefore, the total loss function is

$$L_{total} = \mu L_p + \lambda L_s \tag{16}$$

The final total loss is averaged per pixel, batch, and scale.

Table 1. The standard evaluation metrics for network

Abs Rel	$\frac{1}{\lvert I\rvert}\sum_I \frac{\lvert d_{ij}^{pred}-d_{ij}^{gt}\rvert}{d_{ij}^{gt}}$
Sq Rel	$\frac{1}{\lvert I\rvert}\sum_I \frac{\lVert d_{ij}^{pred}-d_{ij}^{gt}\rVert}{d_{ij}^{gt}}$
RMSE	$\sqrt{\frac{1}{\lvert I\rvert}\sum_I \lVert d_{ij}^{pred}-d_{ij}^{gt}\rVert^2}$
RMSElog	$\sqrt{\frac{1}{\lvert I\rvert}\sum_I \lVert log d_{ij}^{pred}-log d_{ij}^{gt}\rVert^2}$
δ	$\%of\ d\in I\ \max(\frac{d_{ij}^{pred}}{d},\frac{d}{d_{ij}^{pred}})<t$

4. EXPERIMENTS

To evaluate the effectiveness of our approach, some qualitative and quantitative results are provided about depth and pose prediction. KITTI dataset is the main data source to train and test depth networks. The KITTI odometry split was used to train and test our pose network. Meanwhile, the Make3D dataset was used to evaluate the adaptive ability and generalization of the proposed network.

4.1. Implementation details

The proposed depth network has dense skip connections which can fully learn deep abstract features. The network was trained from scratch without pre-training model weights and post-processing. The Sigmoid output of depth map is $D = 1/(\alpha\sigma + \beta)$, where σ and β make the depth value D between 0.1 and 100 units. In our experiments, the MonoDepth2 [22] was set to standard ResNet50 encoder for monocular depth network, ResNet18 for pose network, and without pre-training. Here, we simplify its name to MD2 for the rest of the paper.

Deep learning framework PyTorch [27] was used to implement our model. For comparison, the KITTI dataset was resized and downsampled to 640×192. The proposed network used Adam [28] optimizer with $\beta_1 = 0.9$ and $\beta_1 = 0.999$ to train 22 epochs. The batch size was set as 4 and the smoothness term γ was set to be 0.001. The learning rate was set to be 10^{-4} for the first 20 epochs and reduced by a factor of 10 for the remaining epochs. The settings for the pose network were the same as in [22]. In addition, a single NVIDIA GeForce TITAN X with 12 GB GPU memory was used in our experiments.

4.2. Evaluation metrics

To evaluate our method, we used some standard evaluation metrics, as shown in Table 1.

$|I|$ is the number of pixels in image I. d_{ij}^{pred} is the predicted depth from model. d_{ij}^{gt} is the depth ground truth. δ_t represents the threshold between the depth ground truth and the predicted depth, which is set to be 1.25, 1.25^2, and 1.25^3, respectively.

4.3. KITTI eigen split

The KITTI Eigen split [16] was used to train the proposed network. Before the network was trained, Zhou's [16] preprocessing was used to remove static images. As a result, the training dataset had 39,810 monocular triplets, which contain 29 different scenes. The validation dataset had 4424 images, and there were 697 testing images. The image depth ground truth of the KITTI dataset was captured by Velodyne laser. Following the work in [22], the intrinsics of all images were same, the principal point of the camera was set as image center, and the focal length was defined as the average of all focal lengths in the KITTI dataset. In addition, the depth predicted results were obtained by using the per-image median ground truth scaling proposed in [16]. When the results were evaluated, the maximum depth value was set to be 80 m and the minimum to be 0.1 m.

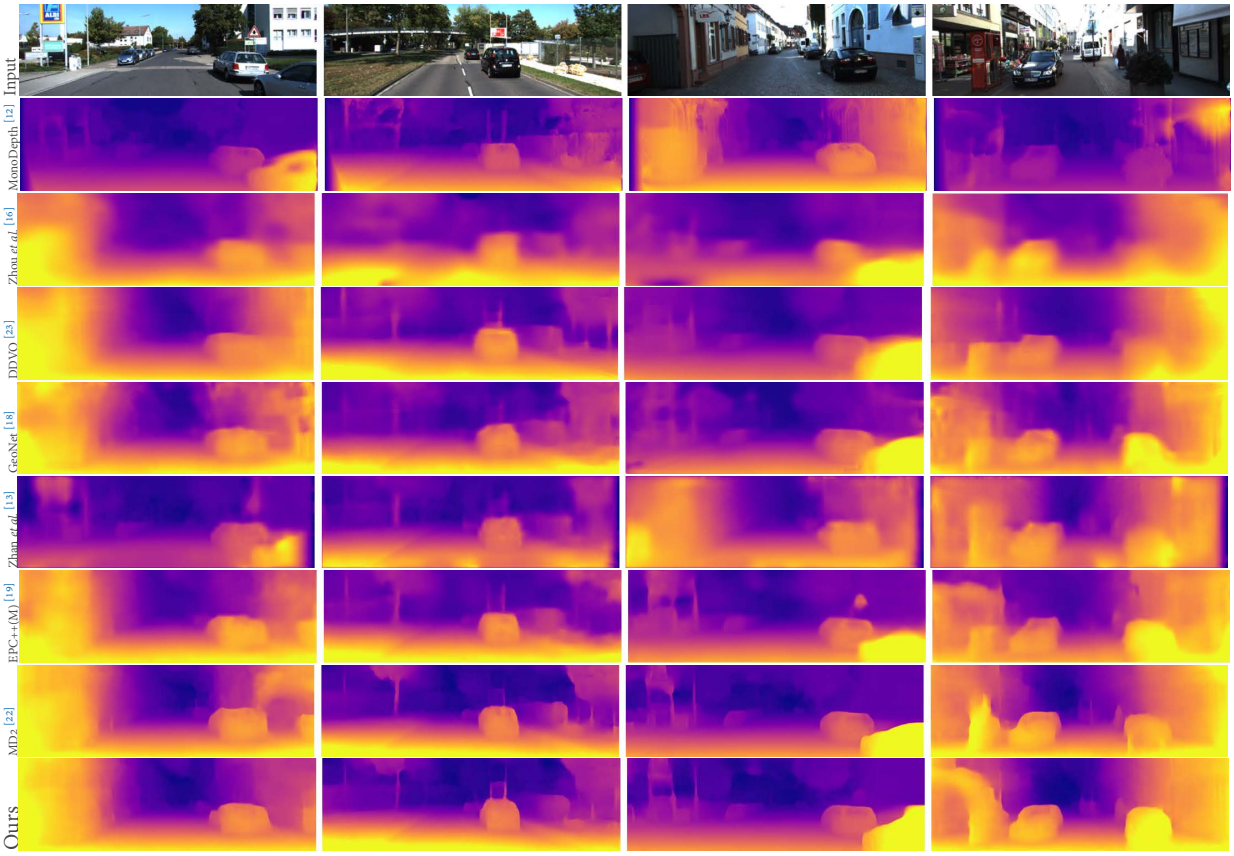


Figure 5. Qualitative results on the KITTI Eigen split. The results are compared with some existing unsupervised methods.

Figure 5 shows some visual examples of predicted depth maps. Our proposed model in the last row generates higher quality depth maps and gets clearer object edges than the other models. Some quantitative results are also provided in Table 2. The evaluation metrics are defined in Table 1. For the first four indices, lower scores are better. For the last three indices, higher scores are better. In Table 2, all results are shown without post-processing[12]. The last row is the predicted result of our proposed method. The accuracy of depth prediction is improved when compared with other methods trained on monocular images. It is demonstrated that the proposed method is effective. Generally, the fewer input images in the pose network have a negative impact on the accuracy of the depth network. Even though only three frames are used to train the pose network at a time, our depth prediction results still outperform the other methods. Note that, some methods in Table 2 [18,19,24] were trained with multiple tasks.

4.4. Additional study

4.4.1. Make3D dataset

The collected scene of the Make3D dataset is different from the KITTI dataset. Therefore, the Make3D dataset is often used to evaluate the adaptability of a network model. Our depth model trained on the KITTI dataset was tested on the Make3D dataset to evaluate its adaptability. The qualitative results are shown in Figure 6. The second column is the depth ground truth. Compared with MD2 [22], the visual results of our model can get the global scene information and capture more object details. It can be seen that our method is useful and has great scene adaptability.

Table 2. The quantitative results. This table shows the results of our method and other existing methods on KITTI Eigen split [16]. The best results in every category are in bold. M denotes the training dataset is monocular. * represents the newer results from GitHub

		Lower is better				Higher is better		
Method	Train	AbsRel	SqRel	RMSE	logRMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou* [16]	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang [29]	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian [30]	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet* [18]	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DDVO [23]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [31]	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [32]	M	0.162	1.352	6.276	0.252	-	-	-
Ranjan [24]	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC++ [19]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth [17]	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
MD2 [22]	M	0.131	1.023	5.064	0.206	0.849	0.951	0.979
Ours	M	0.125	0.992	5.076	0.203	0.858	0.953	0.979

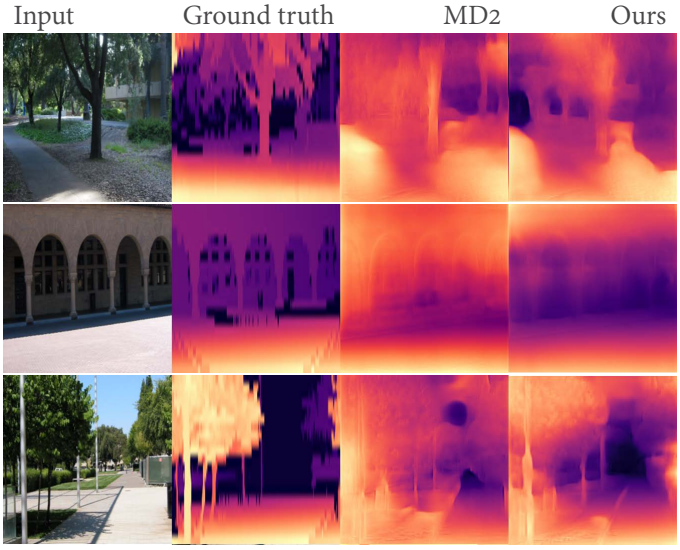


Figure 6. Some predicted depth examples on the Make3D dataset. The models were all trained on KITTI only, monocular, and directly tested on Make3D.

Table 3. Ablation studies on ResNeXt and L_{WSSIM}

		Lower is better				Higher is better		
Method	Train	AbsRel	SqRel	RMSE	logRMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Basic [22]	M	0.131	1.023	5.064	0.206	0.849	0.951	0.979
Basic+ ResNeXt	M	0.127	0.990	5.109	0.205	0.854	0.950	0.978
Basic+ResNeXt+L_{WSSIM}	M	0.125	0.992	5.076	0.203	0.858	0.953	0.979
Basic+ResNeXt+L_{WSSIM}(single scale)	M	0.123	0.980	4.987	0.200	0.862	0.954	0.979

4.4.2. Validating proposed ResNeXt and L_WSSIM

Table 3 shows the result of depth prediction for different components of the proposed method. "Basic" is the MD2 mentioned above. The results clearly prove that the contributions of our proposed terms to the overall performance. It is evident that discrete wavelet transform (DWT) can recover a high-quality clear image and improve the accuracy of depth prediction. The accuracy of depth prediction for both single-scale and multi-scale supervisions are shown. Compared with the multi-scale method, the result of the single-scale method is better. The reason for this phenomenon is hypothesized to be that the low-resolution image has over-smoothed pixel color, which can easily cause inaccurate photometric loss.

Table 4. Model capacity. $params$ **is the number of parameters of depth network,** $totalparams$ **indicates the total parameters for both depth and pose network, and** M **is million unit.**

Method	Params	FLOPs	Total params
MD2(ResNet50) [22]	25.56M	$1.0x10^{10}$	61.8M
ours	25.03M	$1.0x10^{10}$	61.3M

Table 5. Odometry results on the KITTI odometry dataset

Method	Sequence09	Sequence10	Frames
ORB-SLAM [33]	0.014 ± 0.008	0.012 ± 0.011	-
DDVO [26]	0.045 ± 0.108	0.033 ± 0.074	3
Zhou* [16]	0.05 ± 0.039	0.034 ± 0.028	5→2
Mahjourian [30]	0.013 ± 0.010	0.012 ± 0.011	3
GeoNet [18]	0.012 ± 0.007	0.012 ± 0.009	5
EPC++(M) [19]	0.013 ± 0.007	0.012 ± 0.008	3
Ranjan [24]	0.012 ± 0.007	0.012 ± 0.008	5
MD2(M)	0.018 ± 0.009	0.015 ± 0.010	2
ours	0.017 ± 0.010	0.015 ± 0.010	2

4.4.3. Network capacity

To show our proposed network can improve accuracy without increasing network capacity, the number of network parameters and the floating-point operations per second ($FLOPs$) for the network were computed to evaluate the capacity of the proposed network. The quantitative results are shown in Table 4. For the sake of fair comparison, the pose network of MD2 and ours were set as ResNet50. Note that ResNet50 serves as our pose network only for comparison. The pose network adopted in the proposed overall framework is still ResNet18. Compared with MD2, our proposed method improves the accuracy of the depth network without adding extra computational burden, as expected.

4.5. Pose estimation

Our pose model was evaluated on the standard KITTI odometry split [16]. This dataset includes 11 driving sequences. Sequences 00–08 were used to train our pose network without using pose ground truth, while Sequences 09 and 10 were used to evaluate our pose model. The average absolute trajectory error with standard deviation (in meters) was used as evaluation metric. Godard's [22] handling strategy was followed to evaluate the result of the two-frame model on the five-frame snippets. Because Godard's [22] pose estimation results (M, ResNet50 for depth network, and ResNet18 for pose network) are not provided, we retrained and obtained the trained result (MD2).

Only two adjacent frames were taken in our pose model at a time, as shown in Table 5. The output was the relative 6-DoF pose between images. Even though our pose network structure is the same as that in MD2, our pose model obtains better performance than MD2. In addition, the results are comparable to other previous methods. Thus, it is observed that the proposed depth network has a positive effect on pose network.

5. CONCLUSIONS

A versatile end-to-end unsupervised learning framework of monocular depth and pose estimation is developed and evaluated on a dataset in this paper. Aggregated residual transformations (ResNeXt) are embedded in depth network to extract the input image's high-dimensional features. In addition, the proposed wavelet SSIM loss is based on 2D discrete wavelet transform (DWT). Different patches with different frequencies are computed by DWT as the input to the SSIM loss to converge the network, which can recover high-quality clear image patches. The evaluation results show that the performance of depth prediction is improved while the computational burden is reduced. In addition, the proposed method has great adaptive ability on the Make3D

dataset and can decrease the domain gap between different datasets. In future work, how to further optimize the whole system will be considered.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study and performed data analysis, data acquisition and interpretation: Li B
Provided administrative, technical guidance and material support: Zhang H, Wang Z, Hu L

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work is supported by the National Key R&D Program of China (2018YFB1305003), National Natural Science Foundation of China(61922063), and Shanghai Shuguang Project (18SG18).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Zhang K, Chen J, Li Y, Zhang X. Visual tracking and depth estimation of mobile robots without desired velocity information. *IEEE Trans Cybern* 2018;50:361–73.
2. Xiao J, Stolkin R, Gao Y, Leonardis A. Robust fusion of color and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints. *IEEE Trans Cybern* 2017;48:2485–99.
3. Gedik OS, Alatan AA. 3-D rigid body tracking using vision and depth sensors. *IEEE Trans Cybern* 2013;43:1395–405.
4. van der Sommen F, Zinger S,With P. Accurate biopsy-needle depth estimation in limited-angle tomography using multi-view geometry. In: Medical Imaging 2016: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 9786. International Society for Optics and Photonics; 2016. p. 97860D.
5. Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. arXiv preprint arXiv:14062283 2014.
6. Chang Y, Jung C, Sun J. Joint reflection removal and depth estimation from a single image. *IEEE Trans Cybern* 2020.
7. Liu F, Shen C, Lin G. Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. pp. 5162–70.
8. Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N. Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE; 2016. pp. 239–48.
9. Chen W, Fu Z, Yang D, Deng J. Single-image depth perception in the wild. *Advances in Neural Information Processing Systems* 2016;29:730–38.
10. Kuznietsov Y, Stuckler J, Leibe B. Semi-supervised deep learning for monocular depth map prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 6647–55.
11. Garg R, Bg VK, Carneiro G, Reid I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vsion. Springer; 2016. pp. 740–56.

12. Godard C, Mac Aodha O, Brostow GJ. Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 270–79.
13. Zhan H, Garg R, Weerasekera CS, et al. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 340–49.
14. Li R, Wang S, Long Z, Gu D. Undeepvo: Monocular visual odometry through unsupervised deep learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2018. pp. 7286–91.
15. Poggi M, Aleotti F, Tosi F, Mattoccia S. Towards real-time unsupervised monocular depth estimation on cpu. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2018. pp. 5848–54.
16. Zhou T, Brown M, Snavely N, Lowe DG. Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1851–58.
17. Casser V, Pirk S, Mahjourian R, Angelova A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp. 8001–8.
18. Yin Z, Shi J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 1983–92.
19. Luo C, Yang Z, Wang P, et al. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE Trans Pattern Anal Mach Intell* 2019;42:2624–41.
20. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1492–500.
21. Yang HH, Yang CHH, Tsai YCJ. Y-net: Multi-scale feature aggregation network with wavelet structure similarity loss function for single image dehazing. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2020. pp. 2628–32.
22. Godard C, Mac Aodha O, Firman M, Brostow GJ. Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. pp. 3828–38.
23. Wang C, Buenaposada JM, Zhu R, Lucey S. Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 2022–30.
24. Ranjan A, Jampani V, Balles L, et al. Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 12240–49.
25. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Icml; 2010.
26. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 2004;13:600–612.
27. Ketkar N. Introduction to pytorch. In: Deep learning with python. Springer; 2017. pp. 195–208.
28. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:14126980 2014.
29. Yang Z, Wang P, Xu W, Zhao L, Nevatia R. Unsupervised learning of geometry with edge-aware depth-normal consistency. arXiv preprint arXiv:171103665 2017.
30. Mahjourian R, Wicke M, Angelova A. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 5667–75.
31. Zou Y, Luo Z, Huang JB. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 36–53.
32. Yang Z, Wang P, Wang Y, Xu W, Nevatia R. Lego: Learning edge with geometry all at once by watching videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 225–34.
33. Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE T ROBOT* 2015;31:1147–63.

AUTHOR INSTRUCTIONS

1. Submission Overview

Before you decide to publish with *Intelligence & Robotics* (*IR*), please read the following items carefully and make sure that you are well aware of Editorial Policies and the following requirements.

1.1 Topic Suitability

The topic of the manuscript must fit the scope of the journal. Please refer to Aims and Scope for more information.

1.2 Open Access and Copyright

The journal adopts Gold Open Access publishing model and distributes content under the Creative Commons Attribution 4.0 International License. Copyright is retained by authors. Please make sure that you are well aware of these policies.

1.3 Publication Fees

Before December 31, 2024, there are no article processing charges for papers accepted for publication after peer review. OAE subsidizes and helps authors publish their manuscripts totally free. For more details, please refer to OAE Publication Fees.

1.4 Language Editing

All submissions are required to be presented clearly and cohesively in good English. Authors whose first language is not English are advised to have their manuscripts checked or edited by a native English speaker before submission to ensure the high quality of expression. A well-organized manuscript in good English would make the peer review even the whole Editorial handling more smoothly and efficiently.

If needed, authors are recommended to consider the language editing services provided by Charlesworth to ensure that the manuscript is written in correct scientific English before submission. Authors who publish with OAE journals enjoy a special discount for the services of Charlesworth via the following two ways.

Submit your manuscripts directly at http://www.charlesworthauthorservices.com/~OAE;

Open the link http://www.charlesworthauthorservices.com/, and enter Promotion Code “OAE” when you submit.

1.5 Work Funded by the National Institutes of Health

If an accepted manuscript was funded by National Institutes of Health (NIH), the author may inform editors of the NIH funding number. The editors are able to deposit the paper to the NIH Manuscript Submission System on behalf of the author.

2. Submission Preparation

2.1 Cover Letter

A cover letter is required to be submitted accompanying each manuscript. Here is a guideline of a cover letter for authors’ consideration:

List the highlights of the current manuscript and no more than 5 short sentences;

All authors have read the final manuscript, have approved the submission to the journal, and have accepted full responsibilities pertaining to the manuscript’s delivery and contents;

Clearly state that the manuscript is an original work on its own merit, that it has not been previously published in whole or in part, and that it is not being considered for publication elsewhere;

No materials are reproduced from another source (if there is material in your manuscript that has been reproduced from another source, please state whether you have obtained permission from the copyright holder to use them);

Conflicts of interest statement;

If the manuscript is contributed to a Special Issue, please also mention it in the cover letter;

If the manuscript was presented partly or entirely in a conference, the author should clearly state the background information of the event, including the conference name, time, and place in the cover letter.

2.2 Types of Manuscripts

There is no restriction on the length of manuscripts, number of figures, tables and references, provided that the manuscript is concise and comprehensive. The journal publishes Research Article, Review, Technical Note, etc. For more details about paper type, please refer to the following table.

Manuscript Type	Definition	Abstract	Keywords	Main Text Structure
Research Article	A Research Article is a seminal and insightful research study and showcases that often involves modern techniques or methodologies. Authors should justify that their work is of novel findings.	The abstract should state briefly the purpose of the research, the principal results and major conclusions. No more than 250 words.	3-8 keywords	The main content should include four sections: Introduction, Methods, Results and Discussion.
Review	A Review should be an authoritative, well balanced, and critical survey of recent progress in an attractive or a fundamental research field.	Unstructured abstract. No more than 250 words.	3-8 keywords	The main text may consist of several sections with unfixed section titles. We suggest that the author include an "Introduction" section at the beginning, several sections with unfixed titles in the middle part, and a "Conclusions" section at the end.
Technical Note	A Technical Note is a short article giving a brief description of a specific development, technique, or procedure, or it may describe a modification of an existing technique, procedure or device applied in research.	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Editorial	An Editorial is a short article describing news about the journal or opinions of senior Editors or the publisher.	None required	None required	/
Commentary	A Commentary is to provide comments on a newly published article or an alternative viewpoint on a certain topic.	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Perspective	A Perspective provides personal points of view on the state-of-the-art of a specific area of knowledge and its future prospects.	Unstructured abstract. No more than 250 words.	3-8 keywords	/

2.3 Manuscript Structure

2.3.1 Front Matter

2.3.1.1 Title

The title of the manuscript should be concise, specific and relevant, with no more than 16 words if possible.

2.3.1.2 Authors and Affiliations

Authors' full names should be listed. The initials of middle names can be provided. The affiliations and email addresses for all authors should be listed. At least one author should be designated as the corresponding author. In addition, corresponding authors are suggested to provide their Open Researcher and Contributor ID upon submission. Please note that any change to authorship is not allowed after manuscript acceptance. The authors' affiliations should be provided in this format: department, institution, city, postcode, country.

2.3.1.3 Abstract

The abstract should be a single paragraph with word limitation and specific structure requirements (for more details please refer to Types of Manuscripts). It usually describes the main objective(s) of the study, explains how the study was done, including any model organisms used, without methodological detail, and summarizes the most important results and their significance. The abstract must be an objective representation of the study: it is not allowed to contain results that are not presented and substantiated in the manuscript, or exaggerate the main conclusions. Citations should not be included in the abstract.

2.3.1.4 Graphical Abstract

The graphical abstract is essential as this can catch first view of your publication by readers. We recommend you submit an eye-catching figure. It should summarize the content of the article in a concise graphical form. It is recommended to use it because this can make online articles get more attention.

The graphical abstract should be submitted as a separate document in the online submission system. Please provide an image with a minimum of 531 × 1328 pixels (h × w) or proportionally more. The image should be readable at a size of 5 cm × 13 cm using a regular screen resolution of 96 dpi. Preferred file types: TIFF, PSD, AI, JPEG, and EPS files.

2.3.1.5 Keywords

Three to eight keywords should be provided, which are specific to the article, yet reasonably common within the subject discipline.

2.3.2 Main Text

Manuscripts of different types are structured with different sections of content. Please refer to Types of Manuscripts to make sure which sections should be included in the manuscripts.

2.3.2.1 Introduction

The introduction should contain background that puts the manuscript into context, allow readers to understand why the study is important, include a brief review of key literature, and conclude with a brief statement of the overall aim of the work and a comment about whether that aim was achieved. Relevant controversies or disagreements in the field should be introduced as well.

2.3.2.2 Methods

The methods should contain sufficient details to allow others to fully replicate the study. New methods and protocols should be described in detail while well-established methods can be briefly described or appropriately cited. Statistical terms, abbreviations, and all symbols used should be defined clearly. Protocol documents for clinical trials, observational studies, and other non-laboratory investigations may be uploaded as supplementary materials.

2.3.2.3 Results

This section contains the findings of the study. Results of statistical analysis should also be included either as text or as tables or figures if appropriate. Authors should emphasize and summarize only the most important observations. Data on all primary and secondary outcomes identified in the section Methods should also be provided. Extra or supplementary materials and technical details can be placed in supplementary documents.

2.3.2.4 Discussion

This section should discuss the implications of the findings in context of existing research and highlight limitations of the study. Future research directions may also be mentioned.

2.3.2.5 Conclusion

It should state clearly the main conclusions and include the explanation of their relevance or importance to the field.

2.3.3 Back Matter

2.3.3.1 Acknowledgments

Anyone who contributed towards the article but does not meet the criteria for authorship, including those who provided professional writing services or materials, should be acknowledged. Authors should obtain permission to acknowledge from all those mentioned in the Acknowledgments section. This section is not added if the author does not have anyone to acknowledge.

2.3.3.2 Authors' Contributions

Each author is expected to have made substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data, or the creation of new software used in the work, or have drafted the work or substantively revised it.

Please use Surname and Initial of Forename to refer to an author's contribution. For example: made substantial contributions to conception and design of the study and performed data analysis and interpretation: Salas H, Castaneda WV; performed data acquisition, as well as providing administrative, technical, and material support: Castillo N, Young V.

If an article is single-authored, please include "The author contributed solely to the article." in this section.

2.3.3.3 Availability of Data and Materials

In order to maintain the integrity, transparency and reproducibility of research records, authors should include this section in their manuscripts, detailing where the data supporting their findings can be found. Data can be deposited into data repositories or published as supplementary information in the journal. Authors who cannot share their data should state that the data will not be shared and explain it. If a manuscript does not involve such issues, please state "Not applicable." in this section.

2.3.3.4 Financial Support and Sponsorship

All sources of funding for the study reported should be declared. The role of the funding body in the experiment design, collection, analysis and interpretation of data, and writing of the manuscript should be declared. Any relevant grant numbers and the link of funder's website should be provided if any. If the study is not involved with this issue, state "None." in this section.

2.3.3.5 Conflicts of Interest

Authors must declare any potential conflicts of interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. If there are no conflicts of interest, please state "All authors declared that there are no conflicts of interest." in this section. Some authors may be bound by confidentiality agreements. In such cases, in place of itemized disclosures, we will require authors to state "All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their conflicts of interest in this work.". If authors are unsure whether conflicts of interest exist, please refer to the "Conflicts of Interest" of *IR* Editorial Policies for a full explanation.

2.3.3.6 Ethical Approval and Consent to Participate

Research involving human subjects, human material or human data must be performed in accordance with the Declaration of Helsinki and approved by an appropriate ethics committee. An informed consent to participate in the study should also be obtained from participants, or their parents or legal guardians for children under 16. A statement detailing the name of the ethics committee (including the reference number where appropriate) and the informed consent obtained must appear in the manuscripts reporting such research.

Studies involving animals and cell lines must include a statement on ethical approval. More information is available at Editorial Policies.

If the manuscript does not involve such issue, please state "Not applicable." in this section.

2.3.3.7 Consent for Publication

Manuscripts containing individual details, images or videos, must obtain consent for publication from that person, or in the case of children, their parents or legal guardians. If the person has died, consent for publication must be obtained from the next of kin of the participant. Manuscripts must include a statement that written informed consent for publication was obtained. Authors do not have to submit such content accompanying the manuscript. However, these documents must be available if requested. If the manuscript does not involve this issue, state "Not applicable." in this section.

2.3.3.8 Copyright

Authors retain copyright of their works through a Creative Commons Attribution 4.0 International License that clearly states how readers can copy, distribute, and use their attributed research, free of charge. A declaration "© The Author(s) 2021." will be added to each article. Authors are required to sign License to Publish before formal publication.

2.3.3.9 References

References should be numbered in order of appearance at the end of manuscripts. In the text, reference numbers should be placed in square brackets and the corresponding references are cited thereafter. If the number of authors is less than or equal to six, we require to list all authors' names. If the number of authors is more than six, only the first three authors' names are required to be listed in the references, other authors' names should be omitted and replaced with "et al.". Abbreviations of the journals should be provided on the basis of Index Medicus. Information from manuscripts accepted but not published should be cited in the text as "Unpublished material" with written permission from the source.

References should be described as follows, depending on the types of works:

Types	**Examples**
Journal articles by individual authors	Weaver DL, Ashikaga T, Krag DN, et al. Effect of occult metastases on survival in node-negative breast cancer. *N Engl J Med* 2011;364:412-21. [PMID: 21247310 DOI: 10.1056/NEJMoa1008108]
Organization as author	Diabetes Prevention Program Research Group. Hypertension, insulin, and proinsulin in participants with impaired glucose tolerance. *Hypertension* 2002;40:679-86. [DOI: 10.1161/01.HYP.0000035706.28494.09]
Both personal authors and organization as author	Vallancien G, Emberton M, Harving N, van Moorselaar RJ; Alf-One Study Group. Sexual dysfunction in 1,274 European men suffering from lower urinary tract symptoms. *J Urol* 2003;169:2257-61. [PMID: 12771764 DOI: 10.1097/01.ju.0000067940.76090.73]
Journal articles not in English	Zhang X, Xiong H, Ji TY, Zhang YH, Wang Y. Case report of anti-N-methyl-D-aspartate receptor encephalitis in child. *J Appl Clin Pediatr* 2012;27:1903-7. (in Chinese)
Journal articles ahead of print	Odibo AO. Falling stillbirth and neonatal mortality rates in twin gestation: not a reason for complacency. BJOG 2018; Epub ahead of print [PMID: 30461178 DOI: 10.1111/1471-0528.15541]
Books	Sherlock S, Dooley J. Diseases of the liver and billiary system. 9th ed. Oxford: Blackwell Sci Pub; 1993. pp. 258-96.
Book chapters	Meltzer PS, Kallioniemi A, Trent JM. Chromosome alterations in human solid tumors. In: Vogelstein B, Kinzler KW, editors. The genetic basis of human cancer. New York: McGraw-Hill; 2002. pp. 93-113.
Online resource	FDA News Release. FDA approval brings first gene therapy to the United States. Available from: https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm574058.htm. [Last accessed on 30 Oct 2017]

Conference proceedings	Harnden P, Joffe JK, Jones WG, Editors. Germ cell tumours V. Proceedings of the 5th Germ Cell Tumour Conference; 2001 Sep 13-15; Leeds, UK. New York: Springer; 2002.
Conference paper	Christensen S, Oppacher F. An analysis of Koza's computational effort statistic for genetic programming. In: Foster JA, Lutton E, Miller J, Ryan C, Tettamanzi AG, editors. Genetic programming. EuroGP 2002: Proceedings of the 5th European Conference on Genetic Programming; 2002 Apr 3-5; Kinsdale, Ireland. Berlin: Springer; 2002. pp. 182-91.
Unpublished material	Tian D, Araki H, Stahl E, Bergelson J, Kreitman M. Signature of balancing selection in Arabidopsis. *Proc Natl Acad Sci U S A*. Forthcoming 2002.

The journal also recommends that authors prepare references with a bibliography software package, such as EndNote to avoid typing mistakes and duplicated references.

2.3.3.10 Supplementary Materials

Additional data and information can be uploaded as Supplementary Materials to accompany the manuscripts. The supplementary materials will also be available to the referees as part of the peer-review process. Any file format is acceptable, such as data sheet (word, excel, csv, cdx, fasta, pdf or zip files), presentation (powerpoint, pdf or zip files), image (cdx, eps, jpeg, pdf, png or tiff), table (word, excel, csv or pdf), audio (mp3, wav or wma) or video (avi, divx, flv, mov, mp4, mpeg, mpg or wmv). All information should be clearly presented. Supplementary materials should be cited in the main text in numeric order (e.g., Supplementary Figure 1, Supplementary Figure 2, Supplementary Table 1, Supplementary Table 2, *etc*.). The style of supplementary figures or tables complies with the same requirements on figures or tables in main text. Videos and audios should be prepared in English, and limited to a size of 500 MB.

2.4 Manuscript Format

2.4.1 File Format

Manuscript files can be in DOC and DOCX formats and should not be locked or protected.
Manuscript prepared in LaTex must be collated into one ZIP folder (including all source files and images, so that the Editorial Office can recompile the submitted PDF).
When preparing manuscripts in different file formats, please use the corresponding Manuscript Templates.

2.4.2 Length

There are no restrictions on paper length, number of figures, or number of supporting documents. Authors are encouraged to present and discuss their findings concisely.

2.4.3 Language

Manuscripts must be written in English.

2.4.4 Multimedia Files

The journal supports manuscripts with multimedia files. The requirements are listed as follows:
Video or audio files are only acceptable in English. The presentation and introduction should be easy to understand. The frames should be clear, and the speech speed should be moderate;
A brief overview of the video or audio files should be given in the manuscript text;
The video or audio files should be limited to a size of up to 500 MB;
Please use professional software to produce high-quality video files, to facilitate acceptance and publication along with the submitted article. Upload the videos in mp4, wmv, or rm format (preferably mp4) and audio files in mp3 or wav format.

2.4.5 Figures

Figures should be cited in numeric order (e.g., Figure 1, Figure 2) and placed after the paragraph where it is first cited;
Figures can be submitted in format of TIFF, PSD, AI, EPS or JPEG, with resolution of 300-600 dpi;
Figure caption is placed under the Figure;
Diagrams with describing words (including, flow chart, coordinate diagram, bar chart, line chart, and scatter diagram, *etc*.) should be editable in word, excel or powerpoint format. Non-English information should be avoided;
Labels, numbers, letters, arrows, and symbols in figure should be clear, of uniform size, and contrast with the background;
Symbols, arrows, numbers, or letters used to identify parts of the illustrations must be identified and explained in the legend;
Internal scale (magnification) should be explained and the staining method in photomicrographs should be identified;
All non-standard abbreviations should be explained in the legend;
Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial figures and images from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.6 Tables

Tables should be cited in numeric order and placed after the paragraph where it is first cited;
The table caption should be placed above the table and labeled sequentially (e.g., Table 1, Table 2);
Tables should be provided in editable form like DOC or DOCX format (picture is not allowed);
Abbreviations and symbols used in table should be explained in footnote;
Explanatory matter should also be placed in footnotes;
Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial tables from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.7 Abbreviations

Abbreviations should be defined upon first appearance in the abstract, main text, and in figure or table captions and used consistently thereafter. Non-standard abbreviations are not allowed unless they appear at least three times in the text. Commonly-used abbreviations, such as DNA, RNA, ATP, *etc*., can be used directly without definition. Abbreviations in titles and keywords should be avoided, except for the ones which are widely used.

2.4.8 Italics

General italic words like *vs*., *et al*., *etc*., *in vivo*, *in vitro*; *t* test, *F* test, *U* test; related coefficient as *r*, sample number as *n*, and probability as *P*; names of genes; names of bacteria and biology species in Latin.

2.4.9 Units

SI Units should be used. Imperial, US customary and other units should be converted to SI units whenever possible. There is a space between the number and the unit (i.e., 23 mL). Hour, minute, second should be written as h, min, s.

2.4.10 Numbers

Numbers appearing at the beginning of sentences should be expressed in English. When there are two or more numbers in a paragraph, they should be expressed as Arabic numerals; when there is only one number in a paragraph, number < 10 should be expressed in English and number > 10 should be expressed as Arabic numerals. 12345678 should be written as 12,345,678.

2.4.11 Equations

Equations should be editable and not appear in a picture format. Authors are advised to use either the Microsoft Equation Editor or the MathType for display and inline equations.
Display equations should be numbered consecutively, using Arabic numbers in parentheses;
Inline equations should not be numbered, with the same/similar size font used for the main text.

2.4.12 Headings

In the main body of the paper, three different levels of headings may be used.
Level one headings: they should be in bold, and numbered using Arabic numbers, such as **1. INTRODUCTION**, and **2. METHODS**, with all letters capitalized;
Level two headings: they should be in bold and numbered after the level one heading, such as **2.1 Statistical analyses**, **2.2** ..., **2.3**..., *etc*., with the first letter capitalized;
Level three headings: they should be italicized, and numbered after the level two heading, such as *2.1.1 Data distributions*,and *2.1.2 outliers and linear regression*, with the first letter capitalized.

2.4.13 Text Layout

As the electronic submission will provide the basic material for typesetting, it is important to prepare papers in the general editorial style of the journal.
The font is Times New Roman;
The font size is 12pt;
Single column, 1.5× line spacing;
Insert one line break (one Return) before the heading and paragraph, if the heading and paragraph are adjacent, insert a line break before the heading only;
No special indentation;
Alignment is left end;
Insert consecutive line numbers;
For other details please refer to the Manuscript Templates.

2.5 Submission Link

Submit an article via https://oaemesas.com/login?JournalId=ir.

3. Publication Ethics Statement

OAE is a member of the Committee on Publication Ethics (COPE). We fully adhere to its Code of Conduct and to its Best Practice Guidelines.

The Editors of this journal enforce a rigorous peer-review process together with strict ethical policies and standards to guarantee to add high-quality scientific works to the field of scholarly publication. Unfortunately, cases of plagiarism, data falsification, image manipulation, inappropriate authorship credit, and the like, do arise. The Editors of *IR* take such publishing ethics issues very seriously and are trained to proceed in such cases with zero tolerance policy.

Authors wishing to publish their papers in *IR* must abide by the following:

The author(s) must disclose any possibility of a conflict of interest in the paper prior to submission;
The authors should declare that there is no academic misconduct in their manuscript in the cover letter;
Authors should accurately present their research findings and include an objective discussion of the significance of their findings;
Data and methods used in the research need to be presented in sufficient detail in the manuscript so that other researchers can replicate the work;
Authors should provide raw data if referees and the Editors of the journal request;
Simultaneous submission of manuscripts to more than one journal is not tolerated;
Republishing content that is not novel is not tolerated (for example, an English translation of a paper that is already published in another language will not be accepted);
The manuscript should not contain any information that has already been published. If you include already published figures or images, please get the necessary permission from the copyright holder to publish under the CC-BY license;
Plagiarism, data fabrication and image manipulation are not tolerated;
Plagiarism is not acceptable in OAE journals.

Plagiarism involves the inclusion of large sections of unaltered or minimally altered text from an existing source without appropriate and unambiguous attribution, and/or an attempt to misattribute original authorship regarding ideas or results, and copying text, images, or data from another source, even from your own publications, without giving credit to the source.

As to reusing the text that is copied from another source, it must be between quotation marks and the source must be cited. If a study's design or the manuscript's structure or language has been inspired by previous studies, these studies must be cited explicitly.

If plagiarism is detected during the peer-review process, the manuscript may be rejected. If plagiarism is detected after publication, we may publish a Correction or retract the paper.

Falsification is manipulating research materials, equipment, or processes, or changing or omitting data or results so that the findings are not accurately represented in the research record.

Image files must not be manipulated or adjusted in any way that could lead to misinterpretation of the information provided by the original image.

Irregular manipulation includes: introduction, enhancement, moving, or removing features from the original image; the grouping of images that should be presented separately, or modifying the contrast, brightness, or color balance to obscure, eliminate, or enhance some information.

If irregular image manipulation is identified and confirmed during the peer-review process, we may reject the manuscript. If irregular image manipulation is identified and confirmed after publication, we may publish a Correction or retract the paper.

OAE reserves the right to contact the authors' institution(s) to investigate possible publication misconduct if the Editors find conclusive evidence of misconduct before or after publication. OAE has a partnership with iThenticate, which is the most trusted similarity checker. It is used to analyze received manuscripts to avoid plagiarism to the greatest extent possible. When plagiarism becomes evident after publication, we will retract the original publication or require modifications, depending on the degree of plagiarism, context within the published article, and its impact on the overall integrity of the published study. Journal Editors will act under the relevant COPE guidelines.

4. Authorship

Authorship credit of *IR* should be solely based on substantial contributions to a published study, as specified in the following four criteria:

1. Substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data for the work;
2. Drafting the work or revising it critically for important intellectual content;
3. Final approval of the version to be published;
4. Agreement to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

All those who meet these criteria should be identified as authors. Authors must specify their contributions in the section Authors' Contributions of their manuscripts. Contributors who do not meet all the four criteria (like only involved in acquisition of funding, general supervision of a research group, general administrative support, writing assistance, technical editing, language editing, proofreading, *etc*.) should be acknowledged in the section of Acknowledgement in the manuscript rather than being listed as authors.

If a large multiple-author group has conducted the work, the group ideally should decide who will be authors before the work starts and confirm authors before submission. All authors of the group named as authors must meet all the four criteria for authorship.

5. Reviewers Exclusions

You are welcome to exclude a limited number of researchers as potential Editors or reviewers of your manuscript. To ensure a fair and rigorous peer review process, we ask that you keep your exclusions to a maximum of three people. If you wish to exclude additional referees, please explain or justify your concerns—this information will be helpful for Editors when deciding whether to honor your request.

6. Editors and Journal Staff as Authors

Editorial independence is extremely important and OAE does not interfere with Editorial decisions. Editorial staff or Editors shall not be involved in processing their own academic work. Submissions authored by Editorial staff/Editors will be assigned to at least two independent outside reviewers. Decisions will be made by the Editor-in-Chief, including Special Issue papers. Journal staff are not involved in the processing of their own work submitted to any OAE journals.

7. Conflict of Interests

OAE journals require authors to declare any possible financial and/or non-financial conflicts of interest at the end of their manuscript and in the cover letter, as well as confirm this point when submitting their manuscript in the submission system. If no conflicts of interest exist, authors need to state "All authors declared that there are no conflicts of interest". We also recognize that some authors may be bound by confidentiality agreements, in which cases authors need to state "All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their competing interests in this work".

8. Editorial Process

8.1. Pre-Check

New submissions are initially checked by the Managing Editor from the perspectives of originality, suitability, structure and formatting, conflicts of interest, background of authors, *etc*. Poorly prepared manuscripts may be rejected at this stage. If your manuscript does not meet one or more of these requirements, we will return it for further revisions.

Once your manuscript has passed the initial check, it will be assigned to the Assistant Editor, and then the Editor-in-Chief, or an Associate Editor in the case of a conflict of interest, will be notified of the submission and invited to review. Regarding Special Issue paper, after passing the initial check, the manuscript will be successively assigned to the Assistant Editor, and then to the Editor-in-Chief, or an Associate Editor in the case of conflict of interest for the Editor-in-Chief to review. The Editor-in-Chief, or the Associate Editor may reject manuscripts that they deem highly unlikely to pass peer review without further consultation. Once your manuscript has passed the Editorial assessment, the Associate Editor will start to organize peer-review.

All manuscripts submitted to *IR* are screened using CrossCheck powered by iThenticate to identify any plagiarized content. Your study must also meet all ethical requirements as outlined in our Editorial Policies. If the manuscript does not pass any of these checks, we may return it to you for further revisions or decline to consider your study for publication.

8.2. Peer Review

IR operates a single-blind review process, which means that reviewers know the names of authors, but the names of the reviewers are hidden from the authors. The scientific quality of the research described in the manuscript is assessed

by a minimum of two independent expert reviewers. The Editor-in-Chief is responsible for the final decision regarding acceptance or rejection of the manuscript.

All information contained in your manuscript and acquired during the review process will be held in the strictest confidence.

8.3. Decisions

Your research will be judged on scientific soundness only, not on its perceived impact as judged by Editors or referees. There are three possible decisions: Accept (your study satisfies all publication criteria), Invitation to Revise (more work is required to satisfy all criteria), and Reject (your study fails to satisfy key criteria and it is highly unlikely that further work can address its shortcomings). All of the following publication criteria must be fulfilled to enable your manuscript to be accepted for publication:

Originality
The study reports original research and conclusions.
Data availability
All data to support the conclusions either have been provided or are otherwise publicly available.
Statistics
All data have been analyzed through appropriate statistical tests and these are clearly defined.
Methods
The methods are described in sufficient detail to be replicated.
Citations
Previous work has been appropriately acknowledged.
Interpretation
The conclusions are a reasonable extension of the results.
Ethics
The study design, data presentation, and writing style comply with our Editorial Policies.

8.4. Revisions

Authors are required to submit the revised manuscript within one week if minor revision is recommended while two weeks if major revision recommended or one month if additional experiments are needed. If authors need more than one month to revise their manuscript, we usually require the authors to resubmit their paper. We request that a document of point-to-point response to all comments of reviewers and the Editor-in-Chief or the Associate Editor should be supplied along with the revised manuscript to allow quick assessment of your revised manuscript. This document should outline in detail how each of the comments was addressed in the revised manuscript or should provide a rebuttal to the criticism. Manuscripts may or may not be sent to reviewers after revision, dependent on whether the reviewer requested to see the revised version. Apart from in exceptional circumstances, *IR* only supports a round of major revision per manuscript.

9. Contact Us

Journal Contact

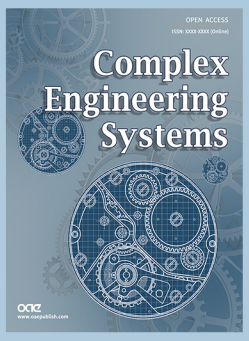
Intelligence & Robotics Editorial Office
Suite 1504, Plaza A, Xi'an National Digital Publishing Base,
No. 996 Tiangu 7th Road, Gaoxin District, Xi'an 710077, Shaanxi, China.

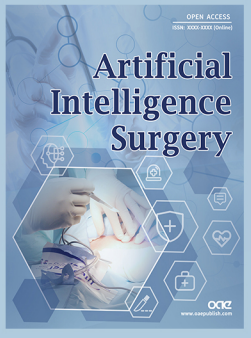
Managing Editor

Lijun Jin
Email: editorial@intellrobot.com

OAE Publishing Inc. (https://oaepublish.com/) is a multidisciplinary open-access publishing company, founded in Los Angeles in 2015. Until now, OAE has been recognized by authoritative organizations in publishing industries, such as the ORCID, COPE, Scientific, Technical and Medical Publishers (STM), Crossref, and EASE.

As of July 2021, more than 1,200 outstanding scholars have joined OAE, who are from world-renowned universities and research institutions, including European Academy of Sciences, American Academy of Invention Sciences, Chinese Academy of Sciences, Royal Academy of Sciences of Belgium, British Academy of Medical Sciences, etc. There are more than 30 journals founded by OAE (https://oaepublish.com/about/journals), such as Intelligence & Robotics, Journal of Materials Informatics, Complex Engineering Systems, Journal of Smart Environments and Green Computing, and Soft Science, etc. Part of journals have been indexed by Scopus and CAS. We are currently working on database application including PubMed and ESCI. Up to July 2021, 2,354 articles have been published online, with 13,131,129 hits and 963,586 downloads. In the future, OAE Publishing Company will continue to found more quality journals with outstanding scholars, to promote the global academic development.





OAE Official Website