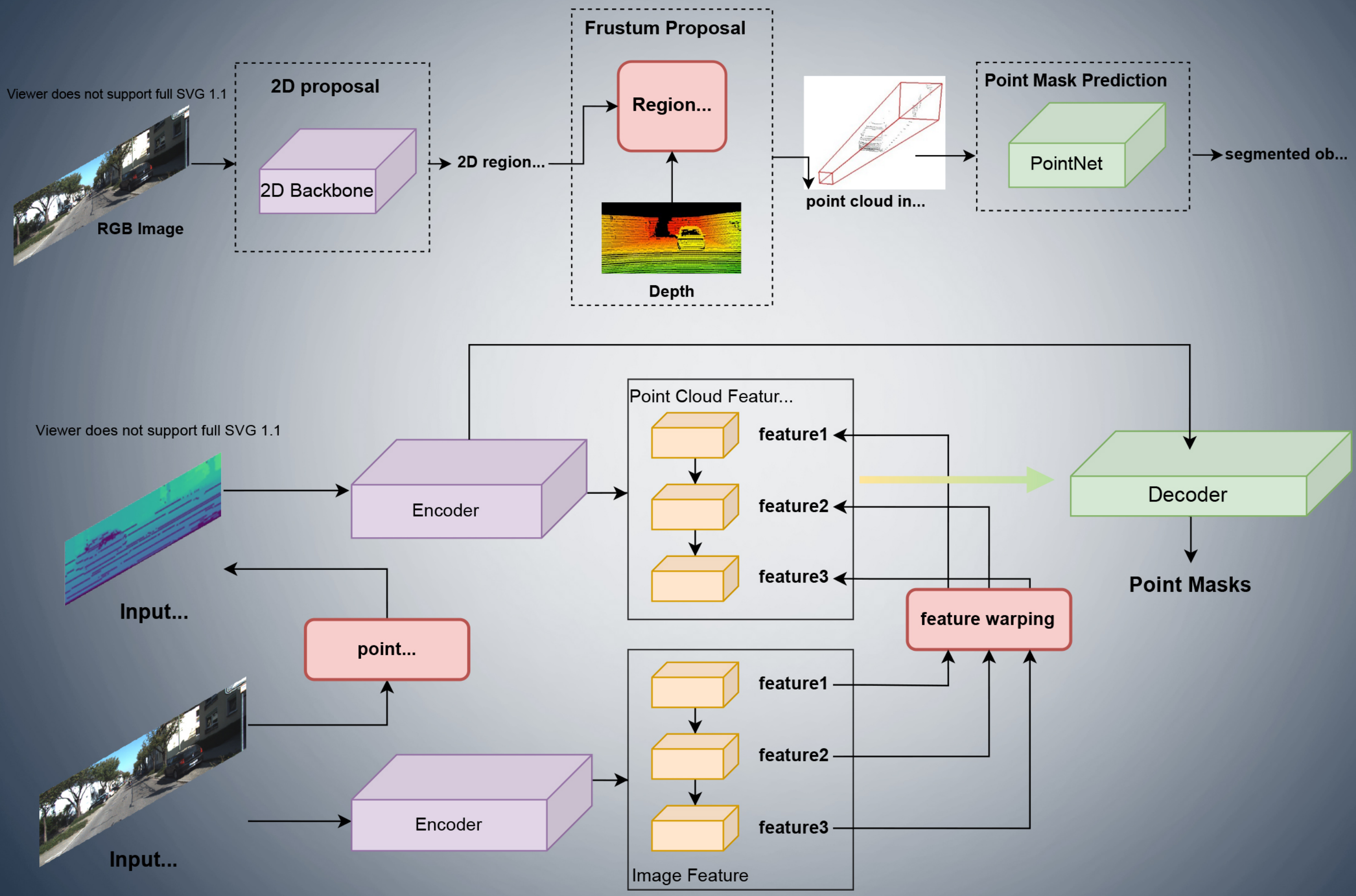
Volume 2, Issue 2, June 2022

INTELLIGENCE & ROBOTICS



Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey

Danni Wu, Zichen Liang, Guang Chen

Open Access
ISSN 2770-3541 (Online)

oae
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
- 2 Executive Editor
- 5 Advisory Editorial Members
- 32 Associate Editors
- 24 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Top-quality unpublished original technical and non-technical application-focused articles are welcome from intelligence and robotics, particularly on the interdisciplinary areas of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

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Intelligence & Robotics (*IR*), ISSN 2770-3541 (Online), publishes top-quality unpublished original technical and non-technical application-focused articles on intelligence and robotics, particularly on the interdisciplinary areas of intelligence and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

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 2, Issue 2

Review

Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey 105
Danni Wu, Zichen Liang, Guang Chen

Research Article

An improved ViBe-based approach for moving object detection 130
Guangyi Tang, Jianjun Ni, Pengfei Shi, Yingqi Li, Jinxiu Zhu

AVDDPG – Federated reinforcement learning applied to autonomous platoon control 145
Christian Boin, Lei Lei, Simon X. Yang

Opponent modeling with trajectory representation clustering 168
Yongliang Lv, Yan Zheng, Jianye Hao

Review

Designs, motion mechanism, motion coordination, and communication of bionic robot fishes: a survey ... 180
Zhiwei Yu, Kai Li, Yu Ji, Simon X. Yan

Wu *et al*. *Intell Robot* 2022;2(2):105-29
DOI: 10.20517/ir.2021.20

Intelligence & Robotics

Review

Open Access

Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey

Danni Wu, Zichen Liang, Guang Chen

School of Automotive Studies, Tongji University, Shanghai 201804, China.

Correspondence to: Prof. Guang Chen, School of Automotive Studies, Tongji University, 4800 Caoan Road, Shanghai 201804, China. E-mail: guangchen@tongji.edu.cn

How to cite this article: Wu D, Liang Z, Chen G. Deep learning for LiDAR-only and LiDAR-fusion 3D perception: a survey. *Intell Robot* 2022;2(2):105-29. http://dx.doi.org/10.20517/ir.2021.20

Received: 31 Dec 2021 **First Decision:** 25 Feb 2022 **Revised:** 10 Mar 2022 **Accepted:** 16 Mar 2022 **Published:** 25 Apr 2022

Academic Editors: Simon X. Yang, Lei Lei **Copy Editor:** Jin-Xin Zhang **Production Editor:** Jin-Xin Zhang

Abstract

The perception system for robotics and autonomous cars relies on the collaboration among multiple types of sensors to understand the surrounding environment. LiDAR has shown great potential to provide accurate environmental information, and thus deep learning on LiDAR point cloud draws increasing attention. However, LiDAR is unable to handle severe weather. The sensor fusion between LiDAR and other sensors is an emerging topic due to its supplementary property compared to a single LiDAR. Challenges exist in deep learning methods that take LiDAR point cloud fusion data as input, which need to seek a balance between accuracy and algorithm complexity due to data redundancy. This work focuses on a comprehensive survey of deep learning on LiDAR-only and LiDAR-fusion 3D perception tasks. Starting with the representation of LiDAR point cloud, this paper then introduces its unique characteristics and the evaluation dataset as well as metrics. This paper gives a review according to four key tasks in the field of LiDAR-based perception: object classification, object detection, object tracking, and segmentation (including semantic segmentation and instance segmentation). Finally, we present the overlooked aspects of the current algorithms and possible solutions, hoping this paper can serve as a reference for the related research.

Keywords: LiDAR, sensor fusion, object classification, object detection, object tracking, segmentation

1. INTRODUCTION

The perception system is crucial for autonomous driving, which enables the autonomous car to understand the surrounding environment with the location, velocity, and future state of pedestrians, obstacles, and other traffic participants. It provides basic and essential information for downstream tasks of autonomous driving (i.e., decision making, planning, and control system). Thus, a precise perception system is vital, which depends on breakthroughs in both hardware and software, i.e., 2D and 3D acquisition technology and perception algorithms.

Sensors equipped on the perception system generally include 2D cameras, RGB-D cameras, radar, and LiDAR. With advantages such as high angular resolution, clear detail recognition, and long-range detection, LiDAR thus becomes indispensable in autonomous driving above the L3 level. LiDAR utilizes pulses of light to translate the physical world into a 3D point cloud in real time with a high level of confidence. By measuring the propagation distance between the LiDAR emitter and the target object and analyzing the reflected energy magnitude, amplitude, frequency, and phase of the reflected wave spectrum on the surface of the target object, LiDAR can present the precise 3D structural information of the target object within centimeter level. According to the scanning mechanism, LiDAR can be divided into three categories: the standard spindle-type, solid-state LiDAR (MEMS), and flash LiDAR. Compared with the standard spindle-type LiDAR, solid-state LiDAR and flash LiDAR provide a solution to high material cost and high mass production cost; therefore, the standard spindle-type LiDAR will be replaced gradually in the future. The application of LiDAR in autonomous cars is gradually gaining market attention. According to Sullivan's statistics and forecasts, the LiDAR market in the automotive segment is expected to reach $8 billion by 2025, accounting for 60% of the total.

In recent decades, deep learning has been attracting extensive attention from computer vision researchers due to its outstanding ability in dealing with massive and unstructured data, which stimulates the growth of environment perception algorithms for autonomous driving. Depending on whether the algorithm concerns the position and pose of the object in real 3D space or just the position of the object in the reflected plane (i.e., image plane), deep learning-based perception algorithms can be divided into 3D and 2D perception. While deep learning-based 2D perception has achieved great progress and thus become a mature branch in the field of computer vision, 3D perception is an emerging topic and yet under-investigated. Relatively, 3D perception outputs abundant information, i.e., height, length, width, and semantic label for each 3D object, to restore the real state of the object in three-dimensional space. In general, the input data of 3D perception tasks contain RGB-D images from depth cameras, images from monocular cameras, binocular cameras, and multi-cameras, and point clouds from LiDAR scanning. Among them, data from LiDAR and multi-camera-based stereo-vision systems achieve higher accuracy in 3D inference. Unlike images from stereo-vision systems, LiDAR point clouds as a relatively new data structure are unordered and possess interaction among points as well as invariance under transformation. These characteristics make deep learning on LiDAR point clouds more challenging. The publication of the pioneering framework PointNet[1] together with PointNet++[2] inspires plenty of works on deep learning for LiDAR point clouds, which will promote the development of autonomous driving perception systems. Hence, this work gives a review of 3D perception algorithms based on deep learning for LiDAR point cloud. However, in real-world applications, a single LiDAR sensor always struggles in heavy weather, color-related detection, and lightly disturbed conditions, which does not fulfill the need of autonomous cars that must perceive surroundings accurately and robustly in all variable and complex conditions. To overcome the shortcomings of a single LiDAR, LiDAR-based fusion[3,4] emerges with improved perception accuracy, reliability, and robustness. Among the LiDAR-fusion methods, the fusion of LiDAR sensors and cameras including visual cameras and thermal cameras is most widely used in the area of robotics and autonomous driving perception. Hence, this paper also reviews deep learning-based fusion methods for LiDAR.

LiDAR-based 3D perception tasks take a LiDAR point cloud (or a LiDAR point cloud fused with images or data from other sensors) as input, and then outputs the category of the target object (3D shape classification); 3D bounding box implying location, height, length, and width with the category of the target object (3D object detection); track ID in a continuous sequence (3D object tracking); segmented label for each point (3D segmentation); etc.[1]. In addition, 3D point cloud registration, 3D reconstruction, 3D point cloud generation, and 6-DOF pose estimation are also tasks worth researching.

Previous related surveys review deep learning methods on LiDAR point cloud before 2021[5–8]. This paper reviews the latest deep learning methods on not only LiDAR point cloud but also LiDAR point cloud fusion (with image and radar). Compared with multi modality fusion surveys[9–11], which cover a wide range of sensors, this paper provides a more detailed and comprehensive review on each related 3D perception task (3D shape classification, 3D object detection, 3D object tracking, and 3D segmentation). The contribution of this paper is summarized as follows:

1. This paper is a survey that focuses on deep learning algorithms with only LiDAR point clouds and LiDAR-based fusion data (especially LiDAR point cloud fused with the camera image) as input in the field of autonomous driving. This work is structured considering four representative 3D perception tasks, namely 3D shape classification, 3D object detection, 3D object tracking, and 3D segmentation.
2. This paper gives a review of methods organized by whether fusion data are utilized as their input data. Moreover, studies and algorithms reviewed in this paper were published in the last decade, which ensures the timeliness and refer-ability of the study.
3. This paper puts some open challenges and possible research directions forward to serve as a reference and stimulate future works.

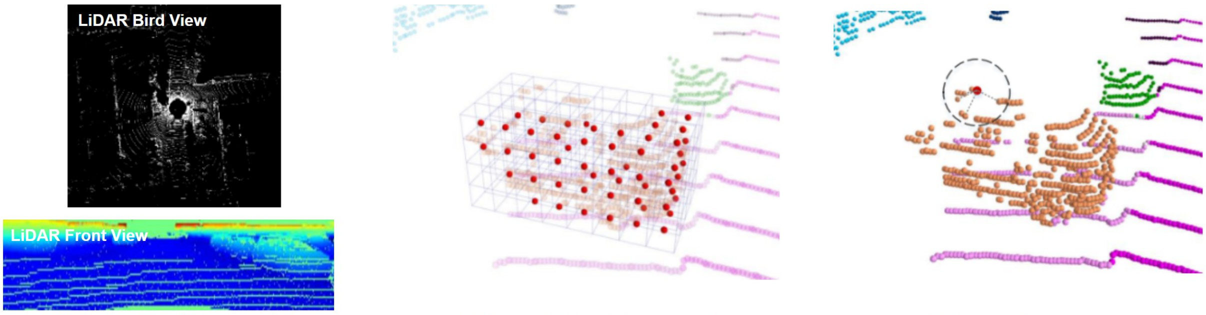
The remainder of this paper is structured as follows. Section 2 provides background knowledge about LiDAR point clouds, including representations and characteristics of LiDAR point cloud, existing LiDAR-based benchmark datasets, and corresponding evaluation metrics. The following four sections give a review of representative LiDAR-only and LiDAR-fusion methods for four 3D perception tasks: Section 3 for 3D shape classification, Section 4 for 3D object detection, Section 5 for 3D object tracking, and Section 6 for 3D semantic segmentation and instance segmentation. Some discussions about overlooked challenges and promising directions are raised in Section 7. At the end, Section 8 draws the conclusions for this paper.

2. BACKGROUND

Point clouds in the field of autonomous driving are generally generated by the on-board LiDAR. The existing mainstream LiDAR emits laser wavelengths of 905 and 1550 nm, which are focused and do not disperse over long distances. When a laser beam of LiDAR hits the surface of an object, the reflected laser carries information of the target object such as location and distance. By scanning the laser beam according to a certain trajectory, the information of the reflected laser points will be recorded. Since the LiDAR scanning is extremely fine, many laser points can be obtained, and thus a LiDAR point cloud is available. The LiDAR point cloud (point clouds mentioned in this paper refer to LiDAR point clouds) is an unordered sparse point set representing the spatial distribution of targets and characteristics of the target surface under the same spatial reference system. There are three approaches basically implemented in deep learning-based methods to process LiDAR point cloud so that processed data can be used as input data to the network: (1) multi-view-based methods; (2) volumetric-based methods; and (3) point-based methods. Multi-view-based methods represent point cloud as 2D views by projecting it onto 2D grid-based feature maps, which can leverage existing 2D convolution methods and

---

[1] Here, we use the term 3D to narrowly describe the tasks with 3D point clouds or 3D point cloud-based fusion data as input and information of the object in real 3D space as output (i.e., category, 3D bounding box, and semantic labels of objects). Broadly speaking, some other works explain 3D tasks as tasks inferring information of the object in real 3D space with any kind of input data.



a) Multi-view Based Representation **b) Volumetric Based Representation** **c) Point Based Representation**

Figure 1. Three approaches for LiDAR point cloud representation: (a) multi-view-based methods; (b) volumetric-based methods; and (c) point-based methods. The image in (a) is original originally from MV3D[12]. The images in (b,c) are original originally from RPVNet[13]

view-pooling layers. Volumetric-based methods discretize the whole 3D space into plenty of 3D voxels, where each point in the original 3D space is assigned to the corresponding voxel following some specific regulations. This representation can preserve rich 3D shape information. Nevertheless, the limitation of performance is inevitable as a result of the spatial resolution and fine-grained 3D geometry loss during the voxelization. On the contrary, point-based methods conduct deep learning methods directly on the point cloud in continuous vector space without transforming the point cloud into other intermediate data representations. This approach avoids the loss caused by transformation and data quantification and preserves the detailed information of the point cloud. The visualization of the three representations is illustrated in Figure 1.

The point cloud carries point-level information (e.g., the x, y, and z coordinates in 3D space, color, and intensities) and keeps invariant under rigid transformation, scaling, and permutation. An azimuth-like physical quantity can be easily acquired from the point cloud, and thus diverse features can be generated for deep learning. Although the point cloud is less affected by the variation of illumination and scale when compared to the image, the point cloud suffers more from the intensity and often ignores sparse information reflected by the surface of objects. The laser emitted by LiDAR cannot bypass obstacles and will be greatly disturbed or even unable to work in the rain, fog, sand, and other severe weather. Thus, challenges exist when extracting features from the spatial-sparse and unordered point sets. Algorithms have evolved from hand-crafted features extraction to deep-learning ones. Among them, point-wise and region-wise methods treat different paths that lead to the same destination. Meanwhile, the cooperation with other sensors shows huge potential to improve the performance through supplementing insufficient information, which may unexpectedly lead to extra computational cost or information redundancy if not well designed. Therefore, studies focus on how to reach a compromise on the cost and the performance when conducting LiDAR-fusion tasks.

With the development of LiDAR, increasing LiDAR point cloud datasets are available, facilitating the training and evaluation among different algorithms. Table 1 [14–28] lists datasets recorded by LiDAR-based visual system. Among them, KITTI[14] provides a comprehensive real-world dataset for autonomous driving, providing a benchmark for 3D object detection, tracking, and scene flow estimation. The evaluation metrics vary for different tasks. For 3D classification, the overall accuracy (OA) and the mean class accuracy (mAcc) are widely used. For 3D object detection, the average precision (AP) and mean average precision (mAP) are mostly-used. For 3D object tracking, precision and success are commonly used as evaluation metrics of single object tracker. Average multi-object tracking Accuracy (AMOTA) and average multi-object tracking precision (AMOTP) are used as evaluation metrics for a 3D multi-object tracker. For 3D segmentation, mean intersection over union (mIoU), OA, and mAcc are widely used for the algorithm evaluation.

Table 1. Dataset recorded by LiDAR-based visual system

Types	Dataset	Year	Data Source	Application
LiDAR-only	Sydney Urban Objects [15]	2013	LiDAR point cloud	Classification
	ScanObjectNN [16]	2019	LiDAR point cloud	Classification
	DALES [17]	2020	LiDAR point cloud	Segmentation
	LASDU [18]	2020	LiDAR point cloud	Segmentation
	Campus3D [19]	2020	LiDAR point cloud	Segmentation
	Toronto-3D [20]	2020	LiDAR point cloud	Segmentation
LiDAR-fusion	KITTI [14]	2012	RGB image + LiDAR point cloud	Majority of tasks
	RueMonge2014 [21]	2014	RGB image + RGB-D image + LiDAR point cloud	Segmentation
	Matterport3D [22]	2017	RGB-D image+ LiDAR point cloud	Segmentation
	H3D [23]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Argoverse [24]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Lyft_L5 [25]	2019	RGB image + LiDAR point cloud	Detection + tracking
	Waymo Open [26]	2020	RGB image + LiDAR point cloud	Detection + tracking
	nuScenes [27]	2020	RGB image + LiDAR point cloud	Detection + tracking
	MVDNet [28]	2021	RaDAR + LiDAR point cloud	Detection

3. 3D SHAPE CLASSIFICATION

Object classification on point cloud is generally known as 3D shape classification or 3D object recognition/classification. There are both inheritance and innovation when transferring 2D object classification to 3D space. For multi-view-based methods, methods for 2D images can be adopted since the point cloud is projected into 2D image planes. However, finding an effective and optimal way to aggregate features of multiple views is still challenging. For point-based methods [29,30], designing novel networks according to the characteristics of the point cloud is the key task. 3D object recognition frameworks usually follow a similar pipeline: Point clouds are first aggregated with an aggregation encoder in order to extract a global embedding. Subsequently, the global embedding is passed through several fully connected layers, after which the object category can be predicted. According to different forms of input data, 3D classifiers can be divided into LiDAR-only classifiers and LiDAR-fusion classifiers. This section reviews existing methods for 3D shape classification. A summary of the algorithms is shown in Table 2, including modalities and representations of data, algorithm novelty, and performance on ModelNet40 [31] dataset for 3D object classification.

3.1. LiDAR-only classification

In terms of diverse representations of the point cloud as input data, LiDAR-only classifiers can be divided into volumetric representation, 2D views representation, and point representation. Different from volumetric representation- and 2D views representation-based models, which preprocess point cloud into voxel or 2D multi-views by projection, point representation-based methods apply a deep learning model on the point cloud directly. Qi *et al.* [1] proposed a path-breaking architecture called PointNet, which works on raw point cloud for the first time. A transformation matrix learned by T-Net can align the input data and a canonical space in order to ensure immutability after certain geometric transformations. Therefore, a global feature can be learned through several multi-layer perceptrons (MLP), T-Net, and max-pooling. Then, the feature is utilized to predict the final classification score by MLP. Shortly after, PointNet++ [2] extracts local features that PointNet [1] ignores at diverse scales and attains deep features through a multi-layer network. It also uses two types of density adaptive layers, multi-scale grouping (MSG) and multi-resolution grouping (MRG), to deal with the feature extraction of unevenly distributed point cloud data. These two works [1,2] can be implemented simply but achieves extraordinary performance at the same time; therefore, several networks are developed on their basis. MomNet [32] is designed on the basis of a simplified version of the PointNet [1] architecture, which consequently requires relatively low computational resources. Inspired by PointNet++ [2], Zhao *et al.* [33] proposed adaptive feature adjustment (AFA) to exploit contextual information in a local region. SRN [34] builds a structural relation network in order to consider local inner interactions. Recently, Yan *et al.* [35] introduced an end-to-end network named PointASNL with an adaptive sampling (AS) module and a local-nonlocal (L-NL) module, achieving excellent performance on the majority of datasets.

While the above methods learn point-wise features through multi-layer perceptrons, some other works adopt 3D convolutional kernels to design convolutional neural networks for point clouds, which can preserve more spatial information of point clouds. One of the typical networks is PointConv[36], which uses a permutation-invariant convolution operation. As an extension of traditional image convolution, the weight functions and the density functions of a given point in PointConv are learned from MLP and kernel density estimation, respectively. Boulch *et al.*[37] built a generalization of discrete convolutions for point clouds by replacing the discrete kernels for grid sampled data with continuous ones. Relation-shape convolutional neural network (RS-CNN)[38] is a hierarchical architecture which leverages the relation-shape convolution (RS-Conv) to learn the geometric topology constraint among points from their relations with an inductive local representation. Inspired by dense connection mode, Liu *et al.*[39] introduced DensePoint, a framework that aggregates outputs of all previous layers through a generalized convolutional operator in order to learn a densely contextual representation of point clouds from multi-level and multi-scale semantics. Apart from continuous convolutional kernels, discrete convolutional kernels play a role in deep learning for point clouds as well. ShellNet[29], a convolution network that utilizes an effective convolution operator called ShellConv, achieves a balance of high performance and short run time. ShellConv partitions the domain into concentric spherical shells and conducts convolutional operation based on this discrete definition. Mao *et al.*[40] proposed InterpConv for object classification, whose key parts are spatially-discrete kernel weights, a normalization term and an interpolation function. Rao *et al.*[41] introduced an architecture named spherical fractal convolutional neural network, in which point clouds are projected into a discrete fractal spherical structure in an adaptive way. Unlike other CNN methods, a novel convolution operator[30] is proposed, which convolves annularly on point clouds and is applied in an annular convolutional neural network (A-CNN), leading to higher performance. Through specified regular and dilated rings along with constraint-based K-NN search methods, the annular convolutional methods can order neighboring points and attain the relationship between ordered points. DRINet[42] develops a dual-representation (i.e., voxel-point and point-voxel) to propagate features between these two representations, performing SOTA on the ModelNet40 dataset with high runtime efficiency.

3.2. LiDAR-fusion classification

Sensors-fusion architectures have become an emerging topic due to their balance among the compatibility with application scenarios, the complementarity of perception information, and the cost. LiDAR is fused with other sensors to deal with specific tasks for autonomous driving. For instance, point clouds and images are fused in order to accomplish the 2D object detection[43,44] and the fusion of LiDAR and radar is applied to localize and track objects more precisely in terms of 3D object detection[4,45]. However, it is desirable to carry out the point cloud based object classification as a single task with fused methods in the field of real-world self-driving cars. Generally, 3D classification is implemented as a branch of 3D object detection architecture to classify targets of a proposal region and help predict the bounding box. Moreover, since the PointNet[1] was proposed in 2017, many studies dealing directly with raw point clouds have been inspired. For 3D classification task, the overall accuracy can achieve 93.6%[16] on the generic benchmark ModelNet40, which satisfies the demand for applications of autonomous car so that 3D classification is not regarded as an independent task. On the other hand, LiDAR-based fusion methods for the object category prediction are not feasible due to the lack of corresponding image datasets aligned with existing point cloud datasets. Only a few works concentrate on the fusion method specifically for 3D classification in the field of autonomous driving. Therefore, this section focuses on the classifier integrated into the LiDAR-fusion 3D detectors or segmentators.

According to the different stages in which sensors data are fused, fusion methods can be divided into early fusion and late fusion. For early fusion, features from different data sources are fused in the input stage by concatenating each individual feature into a unified representation. This representation is sent to a network to get final outputs. For late fusion, the prediction results from the individual uni-modal streams are fused to output the final prediction. Late fusion merges results by summation or averaging in the simplest cases. Compared with early fusion, late fusion lacks the ability to exploit cross correlations among multi-modal data.

Table 2. Experiment results of 3D object classification methods on ModelNet40 benchmark. Here "I", "mvPC", "vPC", "pPC", "rm" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, range map respectively. "OA" represents the overall accuracy that is the mean accuracy for all test instance; "mAcc" represents the mean accuracy that is the mean accuracy for all shape categories. Here the '%' after the number is omitted for simplicity. "-" means the result is not available

Category	Model	Modal. &Repr.	Novelty	OA	mAcc
LiDAR-Only	PointNet[1]	pPC	point-wise MLP+T-Net+global max pooling	89.2	86.2
	PointNet++ [2]	pPC	set abstraction (sampling, grouping, feature learning)+fully connected layers	90.7	90.7
	Momen(e)t[32]	pPC	MLP+max pooling+pPC coordinates and their polynomial functions as input	89.3	86.1
	SRN[34]	pPC	structural relation network(geometric and locational features+MLP)	91.5	-
	PointASNL[35]	pPC	adaptive sampling module+local-nonlocal module	92.9	-
	PointConv[36]	pPC	MLP to approximate a weight function+a density scale	92.5	-
	RS-CNN[38]	pPC	relation-shape convolution(shared MLP+channel-raising mapping)	92.6	-
	DensePoint[39]	pPC	PConv+PPooling(dense connection like)	93.2	-
	ShellNet [29]	pPC	shellconv(KNN+max pooling+shared MLP+conv order)	93.1	-
	InterpConv[40]	pPC	interpolated convolution operation+max pooling	93.0	-
	DRINet[42]	vPC+pPC	sparse point-voxel feature extraction+sparse voxel-point feature extraction	93.0	-
LiDAR-Fusion	MV3D[12]	I&mvPC	3D proposals network+region-based fusion network	-	-
	SCANet[46]	I&mvPC	multi-level fusion+spatial-channel attention+extension spatial upsample module	-	-
	MMF[47]	I&mvPC	point-wise fusion+ROI feature fusion	-	-
	ImVoteNet[48]	I&pPC	lift 2D image votes, semantic and texture cues to the 3D seed points	-	-

Classifiers integrated into two-stage LiDAR-fusion 3D detectors can be divided into two categories: (1) classifiers to distinguish the target and background; and (2) classifiers to predict the final category of the target object. Chen *et al.*[12] designed a deep fusion framework named multi-view 3D networks (MV3D) combining LiDAR point clouds and RGB images. This network designs a deep fusion scheme that alternately performs feature transformation and feature fusion, which belongs to the early fusion architecture. MV3D comprises a 3D proposal network and a region-based fusion network, both of which have a classifier. The classifier in the 3D proposal network regresses to distinguish whether it belongs to the foreground or background, and then the results along with 3D box generated by the 3D box regressor are fed to 3D Proposal Module to generate 3D proposals. The final results are obtained by a multiclass classifier that predicts the category of objects through a deep fusion approach using the element-wise mean for the join operation and fusing regions generated from multi-modal data. Motivated by deep fusion[12], ScanNet[46] proposes multi-level fusion layers fusing 3D region proposals generated by an object classifier and a 3D box regressor to enable interactions among features. ScanNet also introduces the attention mechanism in spatial and channel-wise dimensions in order to capture global and multi-scale context information. The multi-sensor fusion architecture[47] can accomplish several tasks by one framework, including object classification, 3D box estimation, 2D and 3D box refinement, depth completion, and ground estimation. In the 3D classification part, LiDAR point clouds are first projected into ground relative bird's eye view (BEV) representation through the online mapping module, and then features extracted from LiDAR point clouds, and RGB images are fused by the dense fusion module and fed into LiDAR backbone network to predict the probability of the category. This multi-task multi-sensor architecture performs robustly and qualitatively on the TOR4D benchmark. For one-stage 3D fused detectors, the classifier is generally applied in a different way because the one-stage detectors aim to conduct classification and regression simultaneously. Qi *et al.*[48] proposed a one-stage architecture named ImVoteNet, which lifts 2D vote to 3D to improve 3D classification and detection performance. The architecture consists of two parts: One leverages 2D images to pass the geometric, semantic, and texture cues to 3D voting. The other proposes and classifies targets on the basis of a voting mechanism such as Hough voting. The results show that this method boosts 3D recognition with improved mAP compared with the previous best model[49].

4. 3D OBJECT DETECTION

All the deep learning detectors follow a similar idea: they extract the feature from the input data with the backbone and neck of the framework to generate proposals and then classify and locate the objects with a 3D bounding box with the head part. Depending on whether region proposals are generated or not, the object detectors can be categorized into two-stage and single-stage detectors. Two-stage detectors detect the target from the region of interests proposed from the feature map, while single-stage detectors perform tasks based on sliding dense anchor boxes or anchor points from the pyramid map directly. This section summarizes contemporary 3D object detection research, focusing on diverse data modalities from different sensors. Table 3 shows the summary for 3D object detection. Table 4 summarizes experiment results of 3D object detection methods on the KITTI test 3D object detection benchmark.

4.1. LiDAR-only detection

LiDAR-only detection generates a 3D bounding box based on networks that are only fed with a LiDAR point cloud. In general, two-stage detection processes LiDAR data with point-based representation, while single-stage detection performs the task on multiple formats, including point cloud-based, multi-viewed, and volumetric-based representations.

4.1.1. Two-stage detection

For the two-stage detection, segmentation is a widely-used method to remove noisy points and generate proposals in the first sub-module of the detection. One of the typical detection models is IPOD[50], which seeds instance-level proposals with context and local features extracted by projected segmentation. In 2019, STD[51] created point-level spherical anchors and parallel intersection-over-union (IOU) branches to improve the accuracy of the location. Following the proposal scheme of PointRCNN[52] (whose network is illustrated in Figure 2a), PointRGCN[53] introduces a graph convolutional network which aggregates per-proposal/per-frame features to improve the detection performance. Shi *et al*.[54] extended the method of PointRCNN[52] in another way, by obtaining 3D proposals and intra-object part locations with a part-aware module and regressing the 3D bounding boxes based on the fusion of appearance and location features in the part-aggregation framework. HVNet[55] fuses multi-scale voxel features point-wisely, namely hybrid voxel feature encoding. After voxelizing the point cloud at multiple scales, HVNet extracts hybrid voxel features with an attentive voxel feature encoder, and then pseudo-image features are available through scale aggregation in point-wise format. To remedy the proposal size ambiguity problem, LiDAR R-CNN[56] uses boundary offset and virtual point, designing a plug-and-play universal 3D object detector.

4.1.2. Single-stage detection

Unlike the two-stage detector that outputs final fine-grained detection results on the proposals, the single-stage detector classifies and locates 3D objects with a fully convolutional framework and transformed representation. Obviously, this method makes the foreground more susceptible to adjacent background points, thus decreasing the detection accuracy. Multiple methods emerge to solve this problem. For example, VoxelNet[57] extracts voxel-wise features from point clouds in volumetric-based representation with random sampling and normalization, after which it utilizes a 3D-CNN-based framework and region proposal network to detect 3D objects. To bridge the gap between the 3D-CNN-based and 2D-CNN-based detection, the authors of[58] applied PointNet[1] to point clouds to generate vertical-columned representation, which enables point clouds to be processed by the following 2D-CNN-based detection framework. Multi-task learning work[59] introduces a part-sensitive warping module and an auxiliary module to refine the feature extracted from the backbone network by adapting the ROI pooling from R-FCN[60] detection module. As illustrated in Figure 2c, TANet[61] designs a stacked triple attention module and a coarse-to-fine regression module to reduce the disturbance of noisy points and improve the detection performance on hard-level objects. SE-SSD[62] contains a teacher SSD and a student SSD. The teacher SSD produces soft targets by predicting relatively accurate results (after global transformation) from the input point cloud. The student SSD takes augmented input (a novel shape-aware data argumentation) as input, and then is trained with a consistency loss under the supervision of hard-level targets. 3D auto-labeling[63], which aims at saving the cost of human labeling, proposes a novel off-board 3D object detector to exploit complementary contextual information from point cloud sequences, achieving a performance on par with human labels.

Table 3. Summary of 3D object detection methods. Here "I", "mvPC", "vPC", "pPC", "RaPC" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, Radar point cloud respectively

Detector	Category	Model	Modality & Representation	Novelty
Two-stage Detection	LiDAR -Only	IPOD [50]	pPC	a novel point-based proposal generation
		STD [51]	pPC	proposal generation(from point-based spherical anchors)+PointPool
		PointRGCN [53]	pPC	RPN+R-GCN+C-GCN
		SRN [34]	pPC	structural relation network(geometric and locational features+MLP)
		Part-A2 [54]	pPC	intra-object part prediction+RoI-aware point cloud pooling
		HVNet [55]	vPC	multi-scale voxelization+hybrid voxel feature extraction
		LiDAR R-CNN [56]	pPC	R-CNN style second-stage detector(size aware point features)
	LiDAR -Fusion	3D-CVF [64]	I & vPC	CVF(auto-calibrated projection)+adaptive gated fusion network
		Roarnet [65]	I & pPC	RoarNet 2D(geometric agreement search)+RoarNet 3D(RPN+BRN)
		MV3D [12]	I & mvPC	3D proposals network+region-based fusion network
		ScanNet [46]	I & mvPC	multi-level fusion+spatial-channel attention +extension spatial upsample
		MMF [47]	I & mvPC	point-wise fusion+ROI feature fusion
		Pointpainting [66]	I & pPC	image based semantics network+appended (painted) point cloud
		CM3D [67]	I & pPC	pointwise feature fusion+proposal genaration+ROI-wise feature fusion
		MVDNet [28]	RaPC & mvPC	two-stage deep fusion(region-wise feature fusion)
One-stage Detection	LiDAR -Only	VoxelNet [57]	vPC	voxel feature encoding+3D convolutional middle layer+RPN
		PointPillars [58]	pillar points	pillar feature net+backbone(2D CNN)+SSD detection head
		SASSD [59]	pPC	backbone(SECOND)+auxiliary network+PS Warp
		TANet [61]	vPC	Triple Attention module(channel-wise, point-wise, and voxel-wise attention)
		SE-SSD [62]	pPC	teacher and student SSDs+shape aware augumentation+consistency loss
		3D Auto Label [63]	mvPC	motion state classification+static object and dynamic object auto labeling
		ImVoteNet [48]	I & pPC	lift 2D image votes, semantic and texture cues to the 3D seed points
		EPNet [68]	I & pPC	two-stream RPN+LI-Fusion Module+refinement network
	LiDAR-Fusion	CLOCs [69]	I & vPC	a late fusion architecture with any pair of pre-trained 2D and 3D detectors

Table 4. Experiment results of 3D object detection methods on KITTI test 3D object detection benchmark. Average Precision (AP) for car with IoU threshold 0.7, pedestrian with IoU threshold 0.5, and cyclist with IoU threshold 0.5 is shown. "-" means the result is not available

Model	Car			Pedestrian			Cyclist		
	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
IPOD [50]	79.75%	72.57%	66.33%	56.92%	44.68%	42.39%	71.40%	53.46%	48.34%
STD [51]	79.71%	87.95%	75.09%	42.47%	53.29%	38.35%	61.59%	78.69%	55.30%
PointRGCN [53]	85.97%	75.73%	70.60%	-	-	-	-	-	-
Part-A2 [54]	85.94%	77.86%	72.00%	89.52%	84.76%	81.47%	54.49%	44.50%	42.36%
LiDAR R-CNN [56]	85.97%	74.21%	69.18%	-	-	-	-	-	-
3D-CVF [64]	89.20%	80.05%	73.11%	-	-	-	-	-	-
Roarnet [65]	83.71%	73.04%	59.16%	-	-	-	-	-	-
MV3D [12]	71.09%	62.35%	55.12%	-	-	-	-	-	-
SCANet [46]	76.09%	66.30%	58.68%	-	-	-	-	-	-
MMF [47]	86.81%	76.75%	68.41%	-	-	-	-	-	-
CM3D [67]	87.22%	77.28%	72.04%	-	-	-	-	-	-
VoxelNet [57]	77.47%	65.11%	57.73%	39.48%	33.69%	31.51%	61.22%	48.36%	44.37%
PointPillars [58]	79.05%	74.99%	68.30%	52.08%	43.53%	41.49%	75.78%	59.07%	52.92%
SASSD [59]	88.75%	79.79%	74.16%	-	-	-	-	-	-
TANet [61]	84.81%	75.38%	67.66%	54.92%	46.67%	42.42%	73.84%	59.86%	53.46%
SE-SSD [62]	91.49%	82.54%	77.15%	-	-	-	-	-	-
EPNet [68]	89.81%	79.28%	74.59%	-	-	-	-	-	-
CLOCs [69]	88.94%	80.67%	77.15%	-	-	-	-	-	-

4.2. LiDAR-fusion detection

LiDAR-fusion detection enriches the information with the aspect of data sources to improve the performance at a low cost. Its auxiliary input data include RGB images, angular velocity (acceleration), depth images, and so on.

4.2.1. Two-stage detection

The input data of the LiDAR-fusion detector vary in diverse fields with aspects of sampling frequency and data representations. Hence, simple summation or multiplication at the source side contributes little to the improvement of the algorithm performance. In general, two-stage detection fuses the feature map before or after the proposals. To enhance the quality of proposals, 3D-CVF[64] fuses spatial features from images and point clouds in cross-wise views with the auto-calibrated feature projection. Based on PointNet[1], Roarnet[65] designs a two-stage object detection network whose input data contain RGB image and LiDAR point cloud to improve the performance with 3D pose estimation. As for the fusion of ROI-wise feature, Chen *et al.*[12] fused the feature extracted from the bird's eye view and front view of LiDAR as well as the RGB image. As shown in Figure 2b, Scanet[46] applies a spatial-channel attention module and an extension spatial up-sample module to generate proposals of RGB images and point clouds, respectively, in the first stage and then classifies and regresses the 3D bounding box with a novel multi-level fusion method. Meanwhile, some studies adopt multi-fusion methods in the proposed schemes. For instance, the authors of[47] completed a two-stage detection framework with front-end fusion and medium fusion. Its front-end fusion is to merge the sparse depth image (projected from LiDAR point cloud) and RGB image for the image backbone network to extract dense depth feature. The depth feature would be fed into the dense fusion module with LiDAR point clouds and pseudo-LiDAR points to prepare for medium fusion. Vora *et al.*[66] complemented the context information of point cloud with the semantic segmentation results of the image. Through the point painting operation, point clouds are painted by semantic scores, and then the painted point cloud is fed into a point-based 3D detector to produce final results. The pipeline[67] fuses point-wise features and couples 2D–3D anchors (which are generated from images and point clouds, respectively) to improve the quality of proposals in the first stage, after which it handles ROI-wise feature fusion in the second stage. To deal with adverse weather, MVDNet[28] exploits LiDAR and radar's potential complementary advantages. This novel framework conducts a deep late fusion, which means that proposals are generated from two sensors first and then region-wise features are fused. Moreover, MVDNet provides a foggy weather focused LiDAR and radar dataset generated from the Oxford Radar Robotcar dataset. EPNet[68] is a closed-loop two-stage detection network. Its LI-fusion module projects point cloud to images and then generates point-wise correspondence for the fusion. To form the closed-loop, EPNet achieves 3D end-to-end detection on the high definition map and estimates the map on the fly from raw point clouds. ImVoteNet[48] (which is an extension of VoteNet[49]) supplements the point-wise 3D information with the geometrical and semantic features extracted from 2D-images. In its head module, LiDAR-only, image-only, and LiDAR-fusion features all participate in the voting to improve the detection accuracy.

4.2.2. Single-stage detection

Single-stage detectors outperform two-stage detectors in terms of runtime due to their compact network structure. With the goal of high efficiency and accuracy, the fusion of single stage detector is placed in the post-processing stage (i.e., late fusion) in order to maintain the superior single-shot detection performance and improve through supplementary multi-sensor data at the same time. This indicates that only the results of detectors for LiDAR point cloud and other sensor data (e.g., RGB image) are fused in post-processing module without changing any network structure of detectors. CLOCs[69] builds a late fusion architecture with any pair of pre-trained image and LiDAR detectors. The output candidates of LiDAR and image are combined before the non-maximum suppression operation to exploit geometric and semantic consistencies. Individual 2D and 3D candidates are first pre-processed through specific tensor operation so that they are both in a consistent joint representation using sparse tensor. Then, a set of 2D convolution layers are utilized to fuse, which takes the sparse tensor as input and output a processed tensor. The max-pooling operation is conducted on this tensor to map it to the targets (formatted as a score map). Experiment results on the KITTI dataset show that single-stage 3D detector SECOND[70] fusion with 2D detector Cascade R-CNN[71] achieves better performance by a large margin compared to single-modality SECOND. The architecture of CLOCs is shown in Figure 2d.

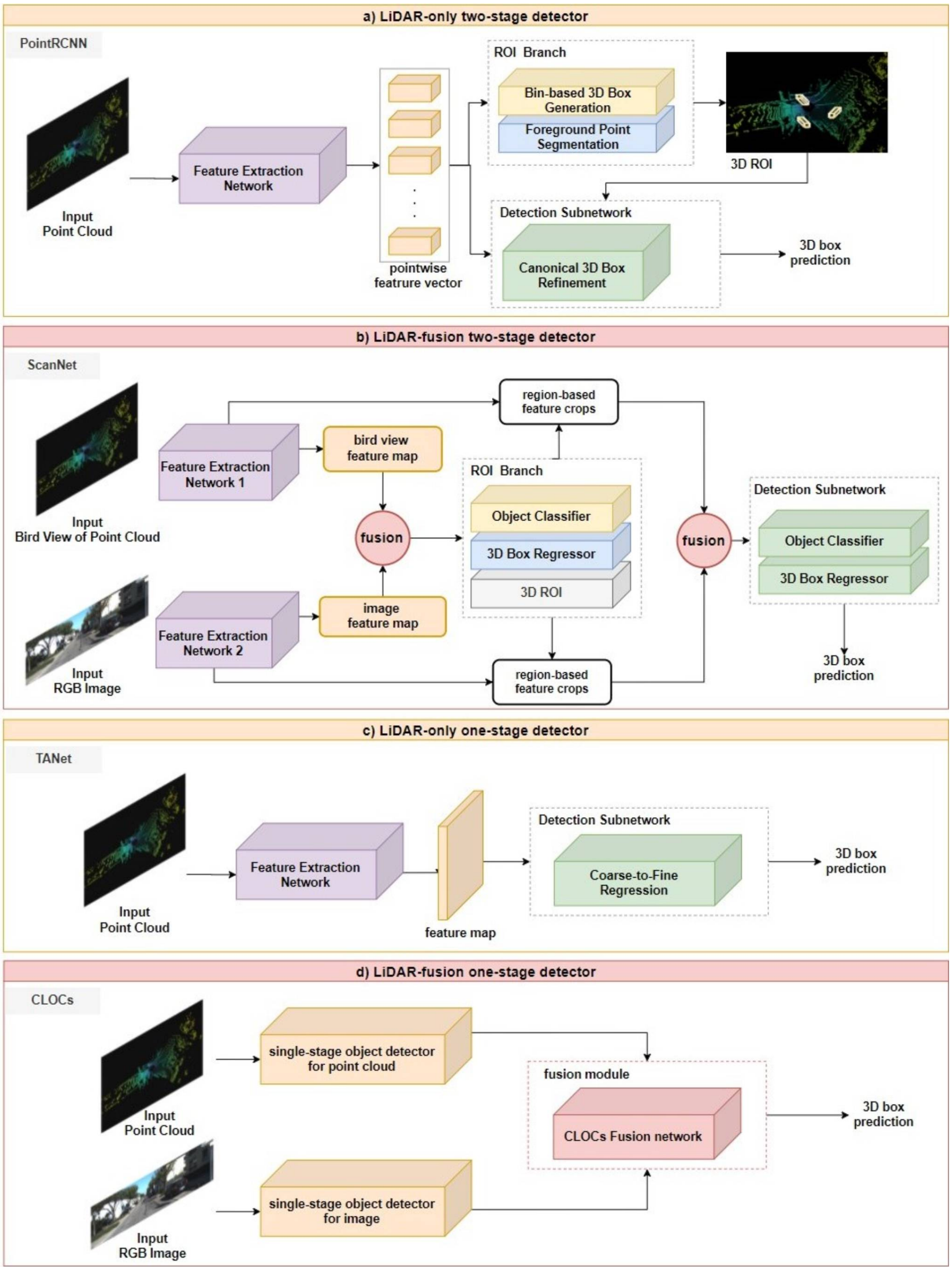

a) LiDAR-only two-stage detector
PointRCNN
Input
Point Cloud
Feature Extraction Network
pointwise featrure vector
ROI Branch
Bin-based 3D Box Generation
Foreground Point Segmentation
3D ROI
Detection Subnetwork
Canonical 3D Box Refinement
3D box prediction
b) LiDAR-fusion two-stage detector
ScanNet
Input
Bird View of Point Cloud
Feature Extraction Network 1
bird view feature map
fusion
image feature map
Feature Extraction Network 2
Input
RGB Image
region-based feature crops
ROI Branch
Object Classifier
3D Box Regressor
3D ROI
region-based feature crops
fusion
Detection Subnetwork
Object Classifier
3D Box Regressor
3D box prediction
c) LiDAR-only one-stage detector
TANet
Input
Point Cloud
Feature Extraction Network
feature map
Detection Subnetwork
Coarse-to-Fine Regression
3D box prediction
d) LiDAR-fusion one-stage detector
CLOCs
Input
Point Cloud
single-stage object detector for point cloud
Input
RGB Image
single-stage object detector for image
fusion module
CLOCs Fusion network
3D box prediction

Figure 2. Typical architectures for two categories of LiDAR-based two-stage 3D detection: (a) LiDAR-only and (b) LiDAR-fusion methods. Typical networks for two categories of LiDAR-based one-stage detector: (c) LiDAR-only and (d) LiDAR-fusion methods.

5. 3D OBJECT TRACKING

All the trackers obey the same rule: they estimate the states of targets contained in the subsequent frames under the guidance of the targets in the first frame. Trackers need to overcome more difficulties, including illumination and scale variation, because trackers perform tasks with richer geometric information and context information compared to image-based trackers and LiDAR-based detectors. Unlike the isolation of single-object tracking and multi-object tracking in the field of the image, in the field of 3D tracking, both trackers are related and the former one can be regarded as a simplified version of the latter one. This section reviews two methods of achieving online 3D tracking: detection and siamese network. Table5 summarizes these works.

5.1. LiDAR-only tracking

As the temporal extension of detection, tracking achieves higher and more precise performance based on appearance similarity and motion trajectory. Tracking-by-detection is an intuitive method. For example, Vaquero *et al.* [72] fused vehicle information segmented from dual-view detectors (i.e., a front view and a bird's eye view) and then utilized extended Kalman filter, Mahalanobis distance, and motion update module to perform 3D tracking. Furthermore, Shi *et al.* [73] performed 3D tracking and domain adaption based on a variant of the 3D detection framework (i.e., PV-RCNN), which comprises temporal information incorporation and classification with RoI-wise features, and so on. In addition, detection results can be enhanced by extra target templates. As a typical example, P2B [74] first matches the proposals with augmented target-specific features and then regresses target-wise centers to generate high-quality detection results for tracking. Following CenterTrack [75], CenterPoint [76] develops an object-center-tracking network through velocity estimation and the point-based detection that views objects as points, achieving more accurate and faster performance.

As for the image-based tracking, the siamese network eliminates the data redundancy and speeds up the task through the conversion from tracking to patch matching, whose idea can be extended in the field of LiDAR-based tracking. Inspired by SAMF [77], Mueller *et al.* [78] designed a correlation filter-based tracker (i.e., SAMF_CA) which incorporates global context in an explicit way. Experiments show that the improved optimization solution achieves a better performance in the single target tracking domain. The work of Zarzar *et al.* [79] shows that the siamese network-based tracking with LiDAR-only data performs well in aerial navigation. Holding the belief that appearance information is insufficient to track, Giancola *et al.* [80] encoded the model shape and candidate shape into latent information with a Siamese tracker. Zarzar *et al.* [81] generated efficient proposals with a siamese network from the BEV representation of point clouds, after which it tracks 3D objects in accordance with the ROI-wise appearance information regularized by the latter siamese framework. PSN [82] first extracts features through a shared PointNet-like framework and then conducts feature augmentation and the attention mechanism through two separate branches to generate a similarity map so as to match the patches. Recently, MLVSNet [83] proposes conducting Hough voting on multi-level features of target and search area instead of only on final features to overcome insufficient target detection in sparse point clouds. Moreover, ground truth bounding box in the first frame can be regarded as a strong cue, enabling a better feature comparison [84], as shown in Figure 3a.

5.2. LiDAR-fusion tracking

Sensors capture data from various views, which is beneficial to supplement insufficient information for trackers. A challenge of tracking-by-detection is how to match the detection results with the context information. The simplest way is to conduct an end-fusion of the tracking results, as done by Manghat *et al.* [85]. In addition, Frossard *et al.* [86] produced precise 3D trajectories for diverse objects in accordance with detection proposals and linear optimization. Introducing the 2D visual information, Complexer-YOLO [87] first performs joint 3D object detection based on the voxelized semantic points clouds (which are fused by image-based semantic information) and then extends the model to multi-target tracking through multi-Bernoulli filter. This work demonstrates the role of scale–rotation–translation, which enables the framework to track in real time.

However, data sampled by different sensors vary in frequency and dimension, and thus it is challenging and not cost-effective to match the similarity among diverse data sources. Recent years have witnessed the emergence of ingenious algorithms while tracking based on a siamese network is still in its infancy. Developed for single object tracking, F-Siamese Tracker [88] extrudes a 2D region-of-interest from a siamese network for the purpose of generating several valid 3D proposals, which would be fed into another siamese network together with a LiDAR template. Although these studies achieve a lot, there is still a long way to go to further integrate point clouds and other sensor data (i.e., RGB images) into the siamese network for LiDAR-fusion tracking. The pipeline of F-Siamese Tracker is explained in Figure 3b.

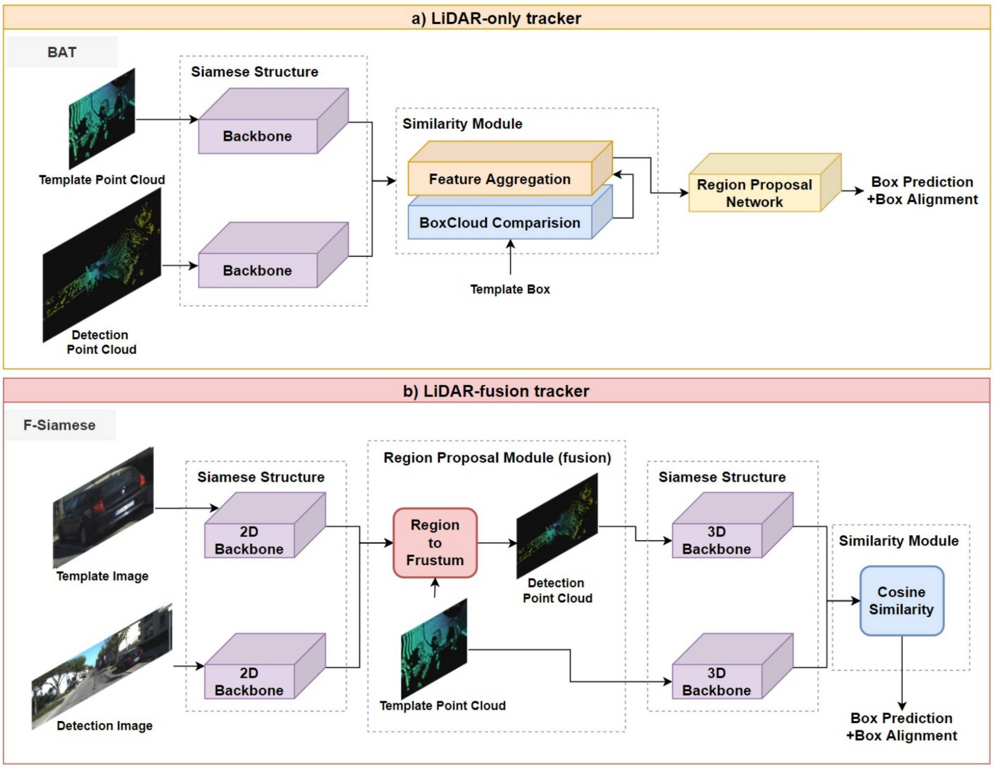


Figure 3. Typical networks for two categories of LiDAR-based tracker: (a) LiDAR-only and (b) LiDAR-fusion methods.

Table 5. Summary of 3D object tracking. Here "I", "mvPC", "vPC", "pPC", "FrustumPC" stands for image, multiple view of point cloud, voxelized point cloud, point cloud, Frustum point cloud respectively

Category	Model	Modality & Representation	Architecture
LiDAR -Only	DualBranch [72]	mvPC	Bbox growing method + multi-hypothesis extended Kalman filter
	PV-RCNN [73]	pPC & vPC	Voxel-to-keypoint 3D scene encoding + keypoint-to-grid RoI feature abstraction
	P2B [74]	pPC	Target-specific feature augmentation + 3D target proposal and verification
	CenterPoint [76]	pillar/vPC	Map-view feature representation + center-based anchor-free head
	SC-ST [80]	pPC	Siamese tracker(resemble the latent space of a shape completion network)
	BEV-ST [81]	mvPC	Efficient RPN+Siamese tracker
	PSN [82]	pPC	Siamese tracker(feature extraction + attention module + feature augumentation)
	MLVSNet [83]	pPC	Multi-level voting+Target-Guided Attention+Vote-cluster Feature Enhancement
	BAT [84]	pPC	Box-aware feature fusion + box-aware tracker
LiDAR -Fusion	MSRT [85]	I&pPC	2D object detector-Faster-RCNN+3D detector-Point RCNN
	MS3DT [86]	I&mvPC	Detection proposals+proposals matching&scoring+linear optimization
	Complexer-YOLO [87]	I&vPC	Frame-wise 3D object detetcion+novel Scale-Rotation-Transalation score
	F-Siamese Tracker [88]	I&FrustumPC	Double Siamese network

6. 3D SEGMENTATION

3D Segmentation methods can be classified into semantic segmentation and instance segmentation, which are both crucial for scene understanding of autonomous driving. 3D Semantic segmentation focuses on per-point semantic label prediction so as to partition a scene into several parts with certain meanings (i.e., per-point class labels), while 3D instance segmentation aims at finding the edge of instances of interest (i.e., per-object masks and class labels). Since Kirillov *et al.*[89] first came up with the concept "panoptic segmentation" that combines semantic segmentation and instance segmentation, several works[90,91] inspired by this concept have been published recently, which build architectures for panoptic segmentation of point cloud. This section specifically focuses on research concerning both 3D semantic segmentation and 3D instance segmentation tasks whose input data are divided into LiDAR point cloud data or LiDAR point cloud fused data. Summaries can be seen in Tables 6 and 7.

6.1. 3D Semantic segmentation

6.1.1. LiDAR-only semantic segmentation

PointNet[1] provides a classic prototype of point cloud semantic segmentation architecture utilizing shared MLPs and symmetrical poolings. On this basis, several dedicated point-wise MLP networks are proposed to attain more information and local structures for each point. PointNet++[2] introduces a novel hierarchical architecture applying PointNet recursively to capture multi-scale local context. Engelmann *et al.*[92] proposed a feature network with K-means and KNN to learn a better feature representation. Besides, an attention mechanism, namely group shuffle attention (GSA)[93] is introduced to exploit the relationships among subsets of point cloud and select a representative one.

Apart from MLP methods, convolutional methods on pure points also achieve some state-of-the-art performance, especially after a fully convolutional network (FCN)[94] is introduced to semantic segmentation, which replaces the fully connected layer with a convolution and thus makes any size of input data possible. Based on the idea of GoogLeNet[95] that takes fisheye cameras and LiDAR sensors data as input, Piewak *et al.*[96] proposed an FCN framework called LiLaNet aiming to label emi-dense LiDAR data point-wisely and multi-class semantically with cylindrical projections of point clouds as input data. The dedicated framework LiLaNet is comprised of a sequence of LiLaBlocks that have various kernels and a 1×1 convolution so that lessons learned from 2D semantic label methods can be converted to the point cloud domain. Recently, a fully convolutional network called 3D-MiniNet[97] extends MiniNet[98] to 3D LiDAR point cloud domain to realize 3D semantic segmentation by learning 2D representations from raw points and passing them to 2D fully convolutional neural network to attain 2D semantic labels. The 3D semantic labels are obtained through re-projection and enhancement of 2D labels.

Based on the pioneering FCN framework, an encoder–decoder framework, U-Net[99] is proposed to conduct multi-scale and large size segmentation. Therefore, several point cloud-based semantic segmentation works extend this framework to 3D space. LU-Net[100] proposes an end-to-end model, consisting of a model that extracts high-level features for each point and an image segmentation network similar to U-Net that takes the projections of these high-level features as input. SceneEncoder[101] presents an encode module to enhance the performance of global information. As shown in Figure 4a, RPVNet[13] exploits fusion advantages of point, voxel, and range map representations of point clouds. After extracting features from the encoder–decoder of three branches and projecting these features into point-based representation, a gated fusion module (GFM) is adopted to fuse features.

Due to the close relationship between the receptive field size and the network performance, a few works concentrate on expanding the receptive fields through dilated/A-trous convolution, which can preserve the spatial resolution at the meanwhile. As an extension of SqueezeSeg[102], the CNN architecture named PointSeg[103] also utilizes SqueezeNet[104] as a backbone network with spherical images generated from point clouds as input. However, PointSeg[103] takes several image-based semantic segmentation networks into consideration and transfers them to the LiDAR domain, instead of using CRF post-processing as in SqueezeSeg[104]. The PointSeg[103] architecture includes three kinds of main layers: fire layer adapted from SqueezeNet[104], squeeze reweighting layer, and enlargement layer where dilated convolutional layers are applied to extend the receptive field. Hua *et al.*[105] introduced a point-wise convolution for 3D point cloud semantic segmentation, which orders point cloud before feature learning and adopts A-trous convolution. Recently, Engelmann *et al.*[106] proposed dilated point convolutions (DPC) to systematically expand the receptive field with an awesome generalization so that it can be applied in most existing CNN for point clouds.

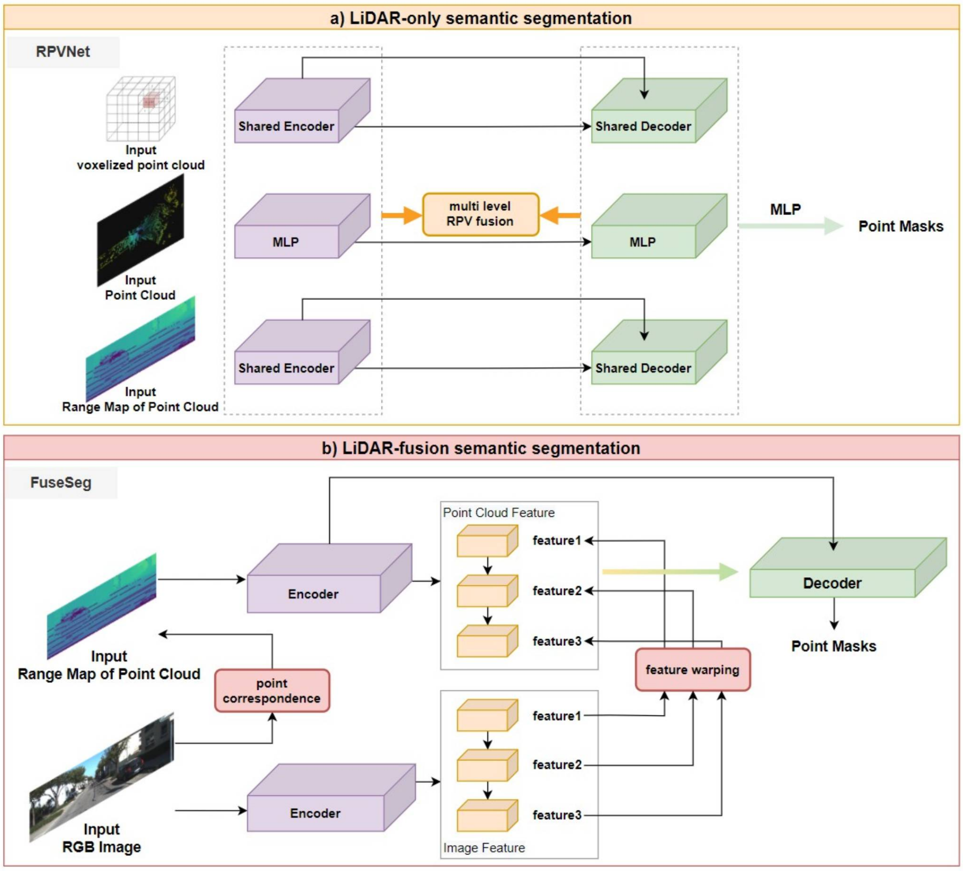


Figure 4. Typical frameworks for two categories of LiDAR-based semantic segmentation: (a) LiDAR-only and (b) LiDAR-fusion methods.

6.1.2. LiDAR-fusion semantic segmentation

One of the challenges existing in point cloud-based semantic segmentation is that the sparseness of the point cloud makes the object seem see-through, thus increasing the difficulty of discernment. Due to the different viewpoints of the RGB camera and LiDAR, RGB images can provide supplementary information about occluding objects. The fusion of RGB images and point clouds for 3D semantic segmentation is intensively researched in recent years due to the achievement of deep learning on 2D image segmentation. 3DMV[107] designs a feature-level fused joint 3D-multi-view prediction network, which combines geometric features of point clouds and color features of RGB images. This work leverages a 2D network to downsample the features extracted from full-resolution RGB input data and then leverages back-projection from a 2D feature into 3D space, rather than just mapping the RGB image on the voxel grid of point cloud. The final results are attained by the 3D convolution layers that take these back-projected 2D features and 3D geometric features as their input. As a result, 3DMV improves 3D semantic segmentation accuracy by 17.2 % in terms of the best volumetric framework at that time. Varga *et al.*[95] proposed an association of fisheye cameras and LiDAR sensors to segment feature-level 3D LiDAR point clouds. In this work, motion correction of point clouds and the undistortion and unwarping process of images are implemented first to ensure the reliability of the information. Subsequently, the undistorted fisheye image is segmented by computing the multiresolution filtered channels and deep CNN channels. Then, to transfer the pixel-wise semantic information to 3D points, the coordinates of 3D points are learned from projections of LiDAR points onto the camera image. With these coordinates, point clouds are augmented with color information and 2D semantic segmentation. Thanks to the well-settled sensor configuration, this super-sensor enables 360-degree environment perception for autonomous cars. MVPNet[108] presents a novel aggregation for feature fusion of point clouds and RGB images. In this work, a proposed multi-view point cloud (MVPC) representation indicates a transformation from 2D image to the 3D point that expresses a discrete approximation of a ground-truth 3D surface by generating a sequence of 1-VPCs and forming predicted MVPC with their union, instead of simply combining projections. FuseSeg[3] proposes a LiDAR point clouds segmentation method that fuses RGB and LiDAR data at feature level and develops a network, whose encoder can be applied as a feature extractor for various 3D perception tasks. Figure 4b demonstrates details of its network. As an extension of SqueezeSeg[102], FuseSeg establishes correspondences between the two input modalities first and warps features extracted from RGB images. Then, the features from images and point clouds are fused by utilizing the correspondences. PMF[109] exploits supplementary advantages between appearance information from RGB images and 3D depth information from LiDAR point clouds. The two-stream network including camera-stream and LiDAR-stream extracts features from projected point cloud and RGB image, and then features from two modalities are fused by a novel residual-based fusion module into LiDAR stream. Additionally, a perception-aware loss contributes to the fusion network's ability. Unlike the ideas above, a novel permutohedral lattice representation method for data fusion is introduced[110]. SParse LATtice Networks (SPLATNet)[110] directly processes a set of points in the representation of a sparse set of samples in a high-dimensional lattice. To reduce the memory and computational cost, SPLATNet adopts a sparse bilateral convolutional layer as the backbone instead. This network incorporates point-based and image-based representations to deal with multi-modal data fusion and processing.

6.2. 3D Instance segmentation

Instance segmentation is the most challenging task of scene understanding because of the necessity to combine object detection and semantic segmentation, which focuses on each individual instance within a class.

6.2.1. LiDAR-only instance segmentation

One of the ideas is a top-down concept (also called the proposal-based method) which detects the bounding box of an instance with object detection methods first and then performs semantic segmentation within the bounding box. GSPN[111] designs a novel architecture for 3D instance segmentation named region-based PointNet (R-PointNet). A generative shape proposal network is integrated into R-PointNet to generate 3D object proposals with instance sensitive features by constructing shapes from the scene, which is converted into a 3D bounding box. The point ROIAlign module aligns features for proposals to refine the proposals and generates segmentation. Different from GSPN[111], the single-stage, anchor-free, and end-to-end 3D-BoNet[112] directly regresses 3D bounding boxes for all instances with a bounding box prediction branch. The backbone network exploits local point features and global features, which are then fed into a point mask prediction branch with a predicted object bounding box, as shown in Figure 5a.

However, the top-down idea ignores the relation between masks and features and extracts masks for each foreground feature, which is redundant. Down-top methods, also named proposal-free methods, may provide a solution for these problems, which performs point-wise semantic segmentation first and then distinguishes different instances. For example, Zhou *et al.*[113] presented an instance segmentation and object detection combined architecture to exploit detailed and global information of objects. It is a two-stage network, containing a spatial embedding (SE)-based clustering and bounding box refinement modules. For instance, segmentation, semantic information is attained by an encoder–decoder network, and object information is attained by SE strategy that takes center points of objects as important information. Aside from the above ideas, utilizing conditional random fields (CRFs) as post-processing methods contributes to the refinement of the label map generated by CNN and further improves the segmentation performance. Inspired by SqueezeNet[104], SqueezeSeg[102] proposes a pioneering lightweight end-to-end pipeline CNN to solve 3D semantic segmentation for road-objects. This network takes transformed LiDAR point cloud as input and then leverages network based on SqueezeNet[104] to extract features and label points semantically, whose results are fed into CRF to refine and output final results. As an extension of SqueezeSeg[102], SqueezeSegV2[114] introduces three novel

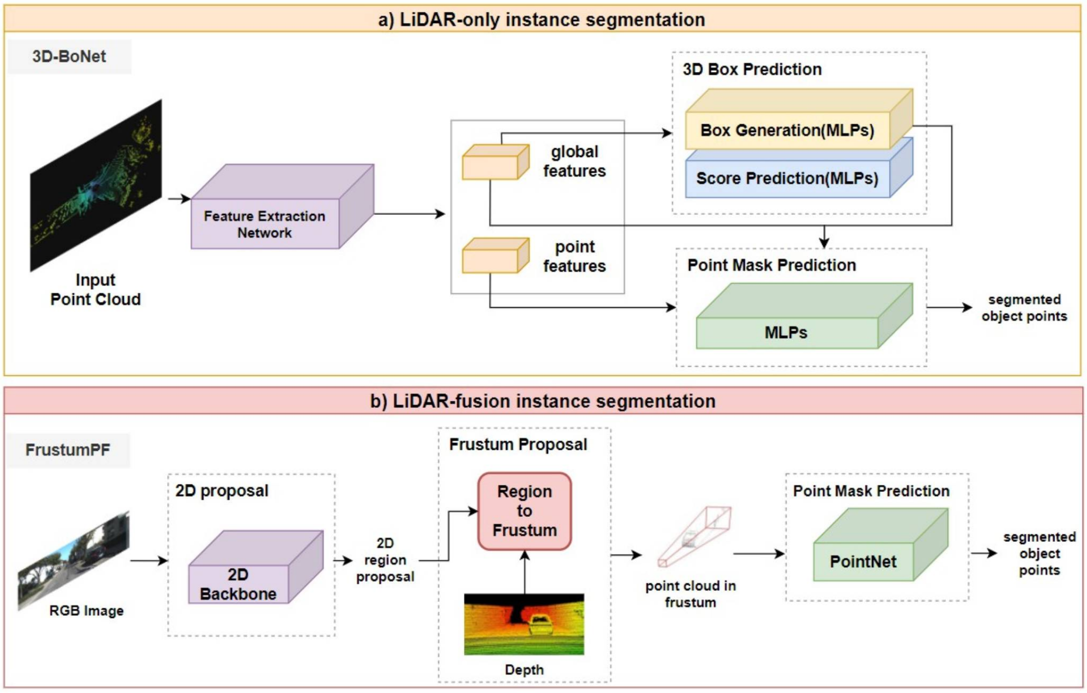


Figure 5. Typical frameworks for two categories of LiDAR-based instance segmentation: (a) LiDAR-only and (b) LiDAR-fusion methods.

modules to dropout noise and improve the accuracy.

6.2.2. LiDAR-fusion instance segmentation

Studies on LiDAR-fusion instance segmentation can also be divided into proposal-based and proposal-free. As for proposal-based methods, 3D-SIS[115] introduces a two-stage image and RGB-D data fused architecture, leveraging both geometric and color signals to jointly and semantically learn features, for instance, segmentation and detection. 3D-SIS consists of two branches, i.e., a 3D detection branch and a 3D mask workflow branch. The backbone of a 3D mask takes projected color, geometry features of each detected object, and 3D detection results as input and outputs final per-voxel mask prediction of each instance. For mask prediction, 3D convolutions with the same spatial resolutions that preserve spatial correspondence with raw point inputs are applied. Then, bounding box prediction generated from 3D-RPN is utilized to attain the key associated mask feature. The final mask of each instance is predicted by a 3D convolution which reduces the dimensionality of features. PanopticFusion[116] presents an online large-scale 3D reconstruction architecture that fuses RGB images and depth images. The 2D instance segmentation network based on Mask-CNN takes the incoming RGB frame as input and fuses both semantic and instance segmentation results to attain point-wise panoptic labels that are integrated into the volumetric map with depth data. As illustrated in Figure 5b, Qi *et al.*[117] proposed a pioneering object detection framework named Fustrum PointNets with point cloud and RGB-D fusion data as input. Frustum PointNets contains three modules: frustum proposal, 3D instance segmentation and a modal 3D box estimation, in order to fuse efficient mature 2D object detector into point cloud domain. The frustum point cloud is extracted from RGB-D data frustum proposal generation first and then is fed into set abstraction layers and point feature propagation layers based on PointNet to predict a mask for each instance by point-wise binary classification. When it comes to proposal-free methods, 3D-BEVIS[118] introduces a framework for 3D semantic and instance segmentation that transfers 2D bird's eye view (BEV) to 3D point space. This framework concentrates on both local point geometry and global context information. 3D instance segmentation network takes point cloud as input, which consists of 2D (i.e., RGB and height above ground) and 3D feature network jointly to exploit point-wise instance features and predicts final instance

Table 6. Summary of 3D semantic segmentation. "I", "mvPC", "vPC", "pPC" and "rm" stands for image, point cloud in multi-view based representation, point cloud in voxel-based representation, point cloud in point-based representation and range map separately

Category	Model	Modality & Representation	Architecture
LiDAR -Only	PointNet [1]	pPC	Point-wise MLP+T-Net+global max pooling
	PointNet++ [2]	pPC	Set abstraction (sampling, grouping, feature learning)+interpolation+skip link concatentation
	KWYND [92]	pPC	Feature network + neighbors definition + regional descriptors
	MPC [93]	pPC	PointNet++-like network+ gumbel subset sampling
	3D-MiniNet [97]	pPC	Fast 3D point neighbor search + 3D MiniNet + post-processing
	LU-Net [100]	pPC & vPC	U-Net for point cloud
	SceneEncoder [101]	pPC	Multi-hot scene descriptor + region similarity loss
	RPVNet [13]	rpc&pPC&vPC	Range-point-voxel fusion network(deep fusion + gated fusion module)
	SqueezeSeg [102]	mvPC	SqueezeNet + conditional random field
	PointSeg [103]	mvPC	SqueezeNet + new feature extract layers
	Pointwise [105]	pPC	Pointwise convolution operator
	Dilated [106]	pPC	Dilated point convolutions
LiDAR -Fusion	3DMV [107]	I & vPC	A novel end-to-end network(back propagation layer)
	SuperSensor [95]	I & mvPC	Associate architecture+360 degree sensor configuration
	MVPNet [108]	I & mvPC	Multi-view point regression network+geometric loss
	FuseSeg [3]	I & rPC	Point correspondece+feature level fusion
	PMF [109]	I & mvPC	Perspective projection+a two-stream network(fusion part)+perception-aware loss

Table 7. Summary of 3D instance segmentation. "I", "mvPC", "vPC", "pPC","FPC" and "rm" stands for image, point cloud in multi-view based representation, point cloud in voxel-based representation, point cloud in point-based representation, point cloud in Frustum representation and range map separately

Category	Model	Modality & Representation	Architecture
LiDAR-Only	GSPN [111]	pPC	Region-based PointNet(generative shape proposal network+Point RoIAlign)
	3D-BoNet [112]	pPC	Instance-level bounding box prediction + point-level mask prediction
	Joint [113]	pPC	Spatial embedding object proposal + local Bounding Boxes refinement
	SqueezeSeg [102]	mvPC	SqueezeNet + conditional random field
	SqueezeSegV2 [114]	mvPC	SqueezeSeg-like + context aggregation module
	3D-BEVIS [118]	mvPC	2D-3D deep model(2D instance feature+3D feature propagation)
LiDAR-Fusion	PanopticFusion [116]	I & vPC	Pixel-wise panoptic labels+a fully connected conditional random field
	Fustrum PointNets [117]	I & FPC	Frunstum proposal+3D instance segmentation(PointNet)

segmentation results through clustering.

7. DISCUSSION

As the upstream and key module of an autonomous vehicle, the perception system outputs its results to downstream modules (e.g., decision and planning modules). Therefore, the performance and reliability of the perception system determine the implementation of downstream tasks, thus affecting the performance of the whole autonomous system. For now, although sensor fusion (Table 8 shows a summary for LiDAR fusion architectures in this paper) can make up for the shortcomings of single LiDAR in bad weather and other aspects, there is still a huge gap between the algorithm design and practical applications in the real world. For this reason, it is necessary to be properly aware of existing open challenges and figure out possible directions to the solution. This section discusses the challenges and possible solutions for LiDAR-based 3D perception.

- **Dealing with large-scale point clouds and high-resolution images.** The need for higher accuracy has prompted researchers to consider larger scale point clouds and higher resolution images. Most the existing algorithms[2,29,36,119] are designed for small 3D point clouds (e.g., 4k points or 1 m × 1 m blocks) without good extending capability to larger point clouds (e.g., millions of points and up to 200 m × 200 m). However, larger point clouds come with a higher computational cost that is hard to afford for self-driving cars with limited computational processing ability. Several recent studies have focused on this problem and proposed some solutions. A deep learning framework for large-scale point clouds named SPG[120] partitions point clouds adaptively to generate a compact yet rich representation by superpoint graph. RandLA-Net[121] leverages random sampling to downsample large-scale point clouds and local feature aggregation module to increase the receptive field size. SCF-Net[122] utilizes the spatial contextual features (SCF) module for large-scale point clouds segmentation. As for sensor fusion, deep learning approaches tackling the fusion of large-scale and high-resolution data should place more emphasis on point-based and multi-view based fusion approaches, which are more scalable than voxel-based ones. Overall, the trade-off between performance and computational cost is inevitable for real application of autonomous driving.

- **A robust representation of fused data.** For deep learning methods, how to pre-process the multi-modal input data is fundamental and important. Although there are several effective representations for point clouds, each of them has both disadvantages and advantages: voxel-based representation has tackled the ordering problem, but, when enlarging the scales of point cloud or increasing the resolution of voxel, the computational cost grows cubically. The quantity of point cloud that can be processed by point based representation methods is limited due to the permutation invariance and computational capacity. A consensus of a unified robust and effective representation for point clouds is necessary. For the data fused with images and point clouds, the representation approaches depend on fusion methods. Image representation-based methods mainly utilizes point clouds projected onto multi-view planes as additional branches of the image. (1) Image representation is not applicable for 3D tasks because the network output results on image plane. (2) Point representation-based methods leverages features or ROI extracted from RGB image as additional channels of point clouds. The performance of this representation is limited by the resolution differences between image (relatively high-resolution) and point clouds (relatively low-resolution). (3) Intermediate data representation methods introduce an intermediate data representation to (e.g., Frustum point cloud and voxelized point cloud). Voxel-based methods are limited in large scale, while frustum based methods have much potential to generate a unified representation based on contextual and structural information of RGB images and LiDAR point clouds.
- **Scene understanding tasks based on data sequences.** The spatiotemporal information implied in the temporally continuous sequence of point clouds and images has been overlooked for a period. Especially for sensor fusion methods, the mismatch of refresh rate between LiDAR and camera causes incorrect time-synchronization between inner perception system and surrounding environment. In addition, predictions based on spatiotemporal information can improve the performance of tasks, such as 3D object recognition, segmentation, and point cloud completion. Research has started to take temporal context into consideration. RNN, LSTM, and derived deep learning models are able to deal with temporal context. Huang *et al.*[123] proposed a multi-frame 3D object detection framework based on sparse LSTM. This work predict 3D objects in the current frame by sending features of each frame and the hidden and memory features from last frame into LSTM module. Yuan *et al.*[124] designed a temporal-channel transformer, whose encoder encodes multi-frame temporal-channel information and decoder decodes spatial-channel information for the current frame. TempNet[125] presents a lightweight semantic segmentation framework for large-scale point cloud sequences, which contains two key modules, temporal feature aggregation (TFA) and partial feature update (PFU). TFA aggregates features only on small portion of key frames with an attentional pooling mechanism, and PFU updates features with the information from non-key frame.

8. CONCLUSIONS

LiDAR captures point-wise information which is less sensitive to illumination than that of cameras. Moreover, it possesses invariance of scale and rigid transformation, showing a promising future in 3D scene understanding. Focusing on the LiDAR-only and LiDAR-fusion 3D perception, this paper first summarizes the LiDAR-based dataset as well as the evaluation metric and then presents a contemporary review of four key tasks: 3D classification, 3D object detection, 3D object tracking, and 3D segmentation. This work also points out the existing challenges and possible development direction. We always hold the belief that LiDAR-only and LiDAR-fusion 3D perception systems would feedback a precise and real-time description of the real-world

Table 8. Fusion stage and fusion methods of LiDAR-fusion tasks. Here, "I" represents image; "L" represents LiDAR point cloud; "R" represents Radar point cloud. Duplicate articles between classification and detection are merged to detection part

Task	Model	Input	FusionStage	Details of the Fusion Method
Classification	ImVoteNet [48]	I&L	Late fusion	Lift 2D image votes, semantic and texture cues to the 3D seed points
Detection	3D-CVF [64]	I&L	Early fusion	Adaptive Gated Fusion: spatial attention maps to mix features according to the region
	Roarnet [65]	I&L	Late fusion	3D detection conducts in-depth inferences recursively with candidate regions from 2D
	MV3D [12]	I&L	Early fusion	Region-based fusion via ROI pooling
	SCANet [46]	I&L	Early fusion	The multi-level fusion module fuses the region-based features
	MMF [47]	I&L	Multi fusion	Region-wise features from multiple views are fused by a deep fusion scheme
	Pointpainting [66]	I&L	Early fusion	Sequential fusion: project point cloud into the output of image semantic seg. network
	CM3D [67]	I&L	Early fusion	Two stage: point-wise feature and ROI-wise feature fusion
	MVDNet [28]	R&L	Early fusion	Region-wise features from two sensors are fused to improve final detection results
	CLOCs [69]	I&L	Late fusion	Output candidates of image and LiDAR point cloud before NMS are fused
Tracking	MSRT [85]	I&L	Late fusion	2D bbox is converted to 3D bbox that are fused to associate between sensor data
	MS3DT [86]	I&L	Early fusion	Object proposals generated by MV3D as input of the match network to link detections
	Compl.-YOLO [87]	I&L	Late fusion	Semantic Voxel Grid: project all relevant voxelized points into the semantic image
	F-Siamese [88]	I&L	Late fusion	2D region proposals are extruded into 3D viewing frustums
Semantic Seg.	3DMV [107]	I&L	Early fusion	3D geometry and per-voxel max-pooled images features are fed into two 3D conv.
	SuperSensor [95]	I&L	Late fusion	Segmentation results from the image space are transferred onto 3D points
	FuseSeg [3]	I&L	Early fusion	Fuse RGB and range image features with point correspondences and feed to net
	PMF [109]	I&L	Early fusion	Residual-based fusion modules fuse image features into LiDAR stream network
Instance Seg.	Pano.Fusion [116]	I&L	Late fusion	2D panoptic segmentation outputs are fused with depth to output volumetric map
	F-PointNets [117]	I&L	Late fusion	Frunstum proposal: extrud each 2D region proposal to a 3D viewing frustum

environment. We hope that this introductory survey serves as a step in the pursuit of a robust, precise, and efficient 3D perception system and guides the direction of its future development.

DECLARATIONS

Authors' contributions

Made substantial contributions to conception and design of the study and performed data analysis and interpretation: Wu D, Liang Z

Performed data acquisition, as well as provided administrative, technical, and material support: Chen G

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

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.

Copyright

REFERENCES

1. Qi CR, Su H, Mo K, Guibas LJ. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 652–60. DOI
2. Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems* 2017;30. DOI
3. Krispel G, Opitz M, Waltner G, Possegger H, Bischof H. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2020. pp. 1874–83. DOI
4. Xu D, Anguelov D, Jain A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 244–53. DOI
5. Guo Y, Wang H, Hu Q, et al. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* 2020. DOI
6. Li Y, Ma L, Zhong Z, et al. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems* 2020;32:3412–32. DOI
7. Liu W, Sun J, Li W, Hu T, Wang P. Deep learning on point clouds and its application: A survey. *Sensors* 2019;19:4188. DOI
8. Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)* 2017;50:1–38. DOI
9. Feng D, Haase-Schütz C, Rosenbaum L, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 2020;22:1341–60. DOI
10. Wang Z, Wu Y, Niu Q. Multi-sensor fusion in automated driving: A survey. *Ieee Access* 2019;8:2847–68. DOI
11. Cui Y, Chen R, Chu W, et al. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems* 2021. DOI
12. Chen X, Ma H, Wan J, Li B, Xia T. Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1907–15. DOI
13. Xu J, Zhang R, Dou J, et al. RPVNet: a deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 16024–33. DOI
14. Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012. pp. 3354–61. DOI
15. De Deuge M, Quadros A, Hung C, Douillard B. Unsupervised feature learning for classification of outdoor 3d scans. In: Australasian Conference on Robitics and Automation. vol. 2; 2013. p. 1. DOI
16. Uy MA, Pham QH, Hua BS, Nguyen T, Yeung SK. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. pp. 1588–97. DOI
17. Varney N, Asari VK, Graehling Q. DALES: a large-scale aerial LiDAR data set for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. pp. 186–87. DOI
18. Ye Z, Xu Y, Huang R, et al. Lasdu: A large-scale aerial lidar dataset for semantic labeling in dense urban areas. *ISPRS International Journal of Geo-Information* 2020;9:450. DOI
19. Li X, Li C, Tong Z, et al. Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In: Proceedings of the 28th ACM International Conference on Multimedia; 2020. pp. 238–46. DOI
20. Tan W, Qin N, Ma L, et al. Toronto-3D: a large-scale mobile lidar dataset for semantic segmentation of urban roadways. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. pp. 202–3. DOI
21. Riemenschneider H, Bódis-Szomorú A, Weissenberg J, Van Gool L. Learning where to classify in multi-view semantic segmentation. In: European Conference on Computer Vision. Springer; 2014. pp. 516–32. DOI
22. Chang A, Dai A, Funkhouser T, et al. Matterport3D: Learning from RGB-D Data in Indoor Environments. In: 2017 International Conference on 3D Vision (3DV). IEEE Computer Society; 2017. pp. 667–76. DOI
23. Patil A, Malla S, Gang H, Chen YT. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In: 2019 International Conference on Robotics and Automation. IEEE; 2019. pp. 9552–57. DOI
24. Chang MF, Lambert J, Sangkloy P, et al. Argoverse: 3d tracking and forecasting with rich maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 8748–57. DOI
25. Kesten R, Usman M, Houston J, et al. Lyft level 5 av dataset 2019. *urlhttps://level5 lyft com/dataset* 2019. Available from: https://level5.lyft.com/dataset.
26. Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 2446–54. DOI
27. Caesar H, Bankiti V, Lang AH, et al. nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11621–31. DOI
28. Qian K, Zhu S, Zhang X, Li LE. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 444–53. DOI
29. Zhang Z, Hua BS, Yeung SK. ShellNet:Efficient point cloud convolutional neural networks using concentric shells statistics. *2019 IEEE/CVF International Conference on Computer Vision* 2019:1607–16. DOI
30. Komarichev A, Zhong Z, Hua J. A-CNN: Annularly convolutional neural networks on point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:7413–22. DOI
31. Wu Z, Song S, Khosla A, et al. 3d shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE conference on

computer vision and pattern recognition; 2015. pp. 1912–20. DOI

32. Joseph-Rivlin M, Zvirin A, Kimmel R. Momen (e) t: Flavor the moments in learning to classify shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019. pp. 0–0. DOI
33. Zhao H, Jiang L, Fu C, Jia J. PointWeb: Enhancing local neighborhood features for point cloud processing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 5560–68. DOI
34. Duan Y, Zheng Y, Lu J, Zhou J, Tian Q. Structural relational reasoning of point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 949–58. DOI
35. Yan X, Zheng C, Li Z, Wang S, Cui S. PointASNL: robust point clouds processing using nonlocal neural networks with adaptive sampling. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 5588–97. DOI
36. Wu W, Qi Z, Fuxin L. PointConv: Deep convolutional networks on 3D point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 9613–22. DOI
37. Boulch A. Generalizing discrete convolutions for unstructured point clouds. In: Biasotti S, Lavoué G, Veltkamp R, editors. Eurographics Workshop on 3D Object Retrieval. The Eurographics Association; 2019. pp. 71–78. DOI
38. Liu Y, Fan B, Xiang S, Pan C. Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 8895–904. DOI
39. Liu YC, Fan B, Meng G, et al. DensePoint: Learning densely contextual representation for efficient point cloud processing. *2019 IEEE/CVF International Conference on Computer Vision* 2019:5238–47. DOI
40. Mao J, Wang X, Li H. Interpolated convolutional networks for 3D point cloud understanding. In: 2019 IEEE/CVF International Conference on Computer Vision; 2019. pp. 1578–87. DOI
41. Rao Y, Lu J, Zhou J. Spherical fractal convolutional neural networks for point cloud recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:452–60. DOI
42. Ye M, Xu S, Cao T, Chen Q. DRINet: A dual-representation iterative learning network for point cloud segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 7447–56. DOI
43. Deng Q, Li X, Ni P, Li H, Zheng Z. Enet-CRF-Lidar: Lidar and camera fusion for multi-scale object recognition. *IEEE Access* 2019;7:174335–44. DOI
44. Wang H, Lou X, Cai Y, Li Y, Chen L. Real-time vehicle detection algorithm based on vision and lidar point cloud fusion. *J Sensors* 2019;2019:8473980:1–:9. DOI
45. Qi CR, Liu W, Wu C, Su H, Guibas LJ. Frustum pointnets for 3D object detection from RGB-D data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition; 2018. pp. 918–27. DOI
46. Lu H, Chen X, Zhang G, et al. SCANet: spatial-channel attention network for 3D object detection. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE; 2019. pp. 1992–96. DOI
47. Liang M, Yang B, Chen Y, Hu R, Urtasun R. Multi-task multi-sensor fusion for 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 7345–53. DOI
48. Qi CR, Chen X, Litany O, Guibas LJ. Imvotenet: boosting 3d object detection in point clouds with image votes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 4404–13. DOI
49. Qi CR, Litany O, He K, Guibas LJ. Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 9277–86. DOI
50. Yang Z, Sun Y, Liu S, Shen X, Jia J. Ipod: Intensive point-based object detector for point cloud. *arXiv preprint arXiv:181205276* 2018. Available from: https://arxiv.org/abs/1812.05276.
51. Yang Z, Sun Y, Liu S, Shen X, Jia J. Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 1951–60. DOI
52. Shi S, Wang X, Li H. Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 770–79. DOI
53. Zarzar J, Giancola S, Ghanem B. PointRGCN: Graph convolution networks for 3D vehicles detection refinement. *arXiv preprint arXiv:191112236* 2019. Available from: https://arxiv.org/abs/1911.12236.
54. Shi S, Wang Z, Shi J, Wang X, Li H. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2020. DOI
55. Ye M, Xu S, Cao T. Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 1631–40. DOI
56. Li Z, Wang F, Wang N. LiDAR R-CNN: An efficient and universal 3D object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 7546–55. DOI
57. Zhou Y, Tuzel O. Voxelnet: end-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 4490–99. DOI
58. Lang AH, Vora S, Caesar H, et al. Pointpillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 12697–705. DOI
59. He C, Zeng H, Huang J, Hua XS, Zhang L. Structure aware single-stage 3D object detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11873–82. DOI
60. Dai J, Li Y, He K, Sun J. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems* 2016;29. Available from: https://proceedings.neurips.cc/paper/2016/hash/577ef1154f3240ad5b9b413aa7346a1e-Abstract.html.

61. Liu Z, Zhao X, Huang T, et al. Tanet: Robust 3d object detection from point clouds with triple attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. pp. 11677–84. DOI
62. Zheng W, Tang W, Jiang L, Fu CW. SE-SSD: self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 14494–503. DOI
63. Qi CR, Zhou Y, Najibi M, et al. Offboard 3D object detection from point cloud sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 6134–44. DOI
64. Yoo JH, Kim Y, Kim J, Choi JW. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: 16th European Conference on Computer Vision, ECCV 2020. Springer; 2020. pp. 720–36. DOI
65. Shin K, Kwon YP, Tomizuka M. Roarnet: a robust 3d object detection based on region approximation refinement. In: 2019 IEEE Intelligent Vehicles Symposium. IEEE; 2019. pp. 2510–15. DOI
66. Vora S, Lang AH, Helou B, Beijbom O. Pointpainting: sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 4604–12. DOI
67. Zhu M, Ma C, Ji P, Yang X. Cross-modality 3d object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. pp. 3772–81. DOI
68. Huang T, Liu Z, Chen X, Bai X. Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision. Springer; 2020. pp. 35–52. DOI
69. Pang S, Morris D, Radha H. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2020. pp. 10386–93. DOI
70. Yan Y, Mao Y, Li B. Second: Sparsely embedded convolutional detection. *Sensors* 2018;18:3337. DOI
71. Cai Z, Vasconcelos N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2019. DOI
72. Vaquero V, del Pino I, Moreno-Noguer F, et al. Dual-Branch CNNs for Vehicle Detection and Tracking on LiDAR Data. *IEEE Transactions on Intelligent Transportation Systems* 2020. DOI
73. Shi S, Guo C, Yang J, Li H. Pv-rcnn: the top-performing lidar-only solutions for 3d detection/3d tracking/domain adaptation of waymo open dataset challenges. *arXiv preprint arXiv:200812599* 2020. Available from: https://arxiv.org/abs/2008.12599.
74. Qi H, Feng C, Cao Z, Zhao F, Xiao Y. P2B: point-to-box network for 3d object tracking in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 6329–38. DOI
75. Zhou X, Koltun V, Krahenbuhl P. Tracking objects as points. In: European Conference on Computer Vision; 2020. pp. 474–90. Available from: https://par.nsf.gov/servlets/purl/10220677.
76. Yin T, Zhou X, Krahenbuhl P. Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 11784–93. DOI
77. Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration. In: European conference on computer vision. Springer; 2014. pp. 254–65. DOI
78. Mueller M, Smith N, Ghanem B. Context-aware correlation filter tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 1396–404. DOI
79. Zarzar Torano JA. Modular autonomous taxiing simulation and 3D siamese vehicle tracking [D]. King Abdullah University of Science and Technology. Thuwal, Saudi Arabia; 2019. Available from: https://repository.kaust.edu.sa/handle/10754/644892.
80. Giancola S, Zarzar J, Ghanem B. Leveraging shape completion for 3d siamese tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 1359–68. DOI
81. Zarzar J, Giancola S, Ghanem B. Efficient tracking proposals using 2D-3D siamese networks on lidar. *arXiv preprint arXiv:190310168* 2019. Available from: https://deepai.org/publication/efficient-tracking-proposals-using-2d-3d-siamese-networks-on-lidar.
82. Cui Y, Fang Z, Zhou S. Point siamese network for person tracking using 3D point clouds. *Sensors* 2020;20:143. DOI
83. Wang Z, Xie Q, Lai YK, et al. MLVSNet: Multi-Level Voting Siamese Network for 3D Visual Tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 3101–10. DOI
84. Zheng C, Yan X, Gao J, et al. Box-aware feature enhancement for single object tracking on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 13199–208. DOI
85. Manghat SK, El-Sharkawy M. A multi sensor real-time tracking with LiDAR and camera. In: 2020 10th Annual Computing and Communication Workshop and Conference. IEEE; 2020. pp. 0668–72. DOI
86. Frossard D, Urtasun R. End-to-end learning of multi-sensor 3d tracking by detection. In: 2018 IEEE International Conference on Robotics and Automation. IEEE; 2018. pp. 635–42. DOI
87. Simon M, Amende K, Kraus A, et al. Complexer-YOLO: real-time 3D object detection and tracking on semantic point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2019. pp. 0–0. DOI
88. Zou H, Cui J, Kong X, et al. F-siamese tracker: a frustum-based double siamese network for 3d single object tracking. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020. pp. 8133–39. DOI
89. Kirillov A, He K, Girshick R, Rother C, Dollár P. Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 9404–13. Available from: https://openaccess.thecvf.com/content_CVPR_2019/papers/Kirillov_Panoptic_Segmentation_CVPR_2019_paper.pdf.
90. Milioto A, Behley J, McCool C, Stachniss C. Lidar panoptic segmentation for autonomous driving. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020. pp. 8505–12. DOI
91. Behley J, Milioto A, Stachniss C. A benchmark for LiDAR-based panoptic segmentation based on KITTI. In: 2021 IEEE International

Conference on Robotics and Automation. IEEE; 2021. pp. 13596–603. DOI
92. Engelmann F, Kontogianni T, Schult J, Leibe B. Know what your neighbors do: 3D semantic segmentation of point clouds. In: Proceedings of the European Conference on Computer Vision Workshops; 2018. pp. 395–409. DOI
93. Yang J, Zhang Q, Ni B, et al. Modeling Point Clouds With Self-Attention and Gumbel Subset Sampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:3318–27. DOI
94. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 3431–40. DOI
95. Varga R, Costea A, Florea H, Giosan I, Nedevschi S. Super-sensor for 360-degree environment perception: Point cloud segmentation using image features. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems; 2017. pp. 1–8. DOI
96. Piewak F, Pinggera P, Schafer M, et al. Boosting lidar-based semantic labeling by cross-modal training data generation. In: Proceedings of the European Conference on Computer Vision Workshops; 2018. pp. 0–0. DOI
97. Alonso I, Riazuelo L, Montesano L, Murillo AC. 3D-MiniNet: Learning a 2D Representation From Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation. *IEEE Robotics and Automation Letters* 2020;5:5432–39. DOI
98. Alonso I, Riazuelo L, Murillo AC. MiniNet: an efficient semantic segmentation convnet for real-time robotic applications. *IEEE Transactions on Robotics* 2020;36:1340–47. DOI
99. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. pp. 234–41. DOI
100. Biasutti P, Lepetit V, Aujol JF, Brédif M, Bugeau A. LU-Net: an efficient network for 3D LiDAR point cloud semantic segmentation based on end-to-end-learned 3D features and U-Net. *2019 IEEE/CVF International Conference on Computer Vision Workshop* 2019:942–50. DOI
101. Xu J, Gong J, Zhou J, et al. SceneEncoder: scene-aware semantic segmentation of point clouds with a learnable scene descriptor. In: 29th International Joint Conference on Artificial Intelligence (IJCAI 2020). International Joint Conferences on Artificial Intelligence; 2021. pp. 601–7. DOI
102. Wu B, Wan A, Yue X, Keutzer K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: 2018 IEEE International Conference on Robotics and Automation. IEEE; 2018. pp. 1887–93. DOI
103. Wang Y, Shi T, Yun P, Tai L, Liu M. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:180706288* 2018. Available from: https://arxiv.org/abs/1807.06288.
104. Iandola FN, Moskewicz MW, Ashraf K, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *ArXiv* 2017;abs/1602.07360. Available from: https://arxiv.org/abs/1602.07360.
105. Hua B, Tran M, Yeung S. Pointwise convolutional neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018. pp. 984–93. DOI
106. Engelmann F, Kontogianni T, Leibe B. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In: 2020 IEEE International Conference on Robotics and Automation. IEEE; 2020. pp. 9463–69. DOI
107. Dai A, Nießner M. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision; 2018. pp. 452–68. DOI
108. Wang J, Sun B, Lu Y. Mvpnet: Multi-view point regression networks for 3d object reconstruction from a single image. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp. 8949–56. DOI
109. Zhuang Z, Li R, Jia K, et al. Perception-aware Multi-sensor Fusion for 3D LiDAR Semantic Segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 16280–90. DOI
110. Su H, Jampani V, Sun D, et al. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2018:2530–39. DOI
111. Yi L, Zhao W, Wang H, Sung M, Guibas L. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:3942–51. DOI
112. Yang B, Wang J, Clark R, et al. Learning object bounding boxes for 3D Instance segmentation on point clouds. *Advances in Neural Information Processing Systems* 2019;32:6740–49. Available from: https://proceedings.neurips.cc/paper/2019/file/d0aa518d4d3bfc721aa0b8ab4ef32269-Paper.pdf.
113. Zhou D, Fang J, Song X, et al. Joint 3D instance segmentation and object detection for autonomous driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 1836–46. DOI
114. Wu B, Zhou X, Zhao S, Yue X, Keutzer K. SqueezeSegV2: improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *2019 International Conference on Robotics and Automation* 2019:4376–82. DOI
115. Hou J, Dai A, Nießner M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2019:4416–25. DOI
116. Narita G, Seno T, Ishikawa T, Kaji Y. PanopticFusion: online volumetric semantic mapping at the level of stuff and things. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2019:4205–12. DOI
117. Qi CR, Liu W, Wu C, Su H, Guibas L. Frustum pointnets for 3D object detection from RGB-D data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2018:918–27. DOI
118. Elich C, Engelmann F, Kontogianni T, Leibe B. 3D Bird's-eye-view instance segmentation. In: German Conference on Pattern Recognition. Springer; 2019. pp. 48–61. DOI
119. Komarichev A, Zhong Z, Hua J. A-CNN: annularly convolutional neural networks on point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. pp. 7413–22. DOI

120. Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 4558–67. DOI
121. Hu Q, Yang B, Xie L, et al. Randla-net: efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 11108–17. DOI
122. Fan S, Dong Q, Zhu F, et al. SCF-Net: learning spatial contextual features for large-scale point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. pp. 14504–13. DOI
123. Huang R, Zhang W, Kundu A, et al. An lstm approach to temporal 3d object detection in lidar point clouds. In: European Conference on Computer Vision. Springer; 2020. pp. 266–82. DOI
124. Yuan Z, Song X, Bai L, Wang Z, Ouyang W. Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology* 2021. DOI
125. Zhou Y, Zhu H, Li C, et al. TempNet: Online Semantic Segmentation on Large-Scale Point Cloud Series. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. pp. 7118–27. DOI

Tang *et al*. *Intell Robot* 2022;2(2):130-44
DOI: 10.20517/ir.2022.07

Intelligence & Robotics

Research Article

Open Access

An improved ViBe-based approach for moving object detection

Guangyi Tang[1]**, Jianjun Ni**[1,2]**, Pengfei Shi**[1,2]**, Yingqi Li**[1]**, Jinxiu Zhu**[1]

[1]College of Internet of Things Engineering, Hohai University, Changzhou 213022, Jiangsu, China.
[2]Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University, Changzhou 213022, Jiangsu, China.

Correspondence to: Prof. Jianjun Ni, College of Internet of Things Engineering, Hohai University, No.200, North Jinling Road, Xinbei District, Changzhou 213022, Jiangsu, China. E-mail: njjhhuc@gmail.com

How to cite this article: Tang G, Ni J, Shi P, Li Y, Zhu J. An improved ViBe-based approach for moving object detection. *Intell Robot* 2022;2(2):130-44. http://dx.doi.org/10.20517/ir.2022.07

Received: 5 Mar 2022 **First Decision:** 14 Apr 2022 **Revised:** 30 Apr 2022 **Accepted:** 12 May 2022 **Published:** 20 May 2022

Academic Editors: Simon X. Yang, Daqi Zhu **Copy Editor:** Fanlin Lan **Production Editor:** Fanlin Lan

Abstract

Moving object detection is a challenging task in the automatic monitoring field, which plays a crucial role in most video-based applications. The visual background extractor (ViBe) algorithm has been widely used to deal with this problem due to its high detection rate and low computational complexity. However, there are some shortcomings in the general ViBe algorithm, such as the ghost area problem and the dynamic background problem. To deal with these problems, an improved ViBe approach is presented in this paper. In the proposed approach, a mode background modeling method is used to accelerate the process of the ghost elimination. For the detection of moving object in dynamic background, a local adaptive threshold and update rate is proposed for the ViBe approach to detect foreground and update background. Furthermore, an improved shadow removal method is presented, which is based on the HSV color space combined with the edge detection method. Finally, some experiments were conducted, and the results show the efficiency and effectiveness of the proposed approach.

Keywords: Moving object detection; ViBe-based approach; dynamic background; shadow detection

1. INTRODUCTION

The real-time detection of moving objects is an essential task in the computer vision field, which has wide applications, including target tracking, video surveillance, abnormal behavior analysis, intelligent robot, etc[1–5]. There are still many challenges of the moving object detection under natural scenes, such as illumination changes, swaying leaves, and shadow changes[6,7]. Therefore, it has attracted more and more attention from researchers recently.

There are many research achievements in moving object detection. For example, Sengar and Mukhopadhyay[8] proposed a motion detection method using block based bi-directional optical flow method. Chen *et al.*[9] proposed an end-to-end deep sequence learning architecture for moving object detection. Li *et al.*[10] presented a novel technique for background subtraction based on the dynamic autoregressive moving average (ARMA) model. These methods used for moving objects detection can be divided into three main types: the optical flow method, the deep learning method, and the difference method. In addition, the difference methods are further divided into three categories[11,12], namely the unsupervised method[13], the supervised method[14,15], and the semi-supervised method[16,17]. There are some drawbacks in the optical flow method, such as complex computation and sensitivity to illumination mutation, which is not suitable for real-time moving objects detection[18]. Compared with traditional algorithms, deep learning methods have the advantages of high detection accuracy and strong fitting ability, but the size of the dataset determines the effect of detection, and it is difficult to meet the needs of deploying in some special scenarios at any time without sufficient samples. At the same time, they have higher requirements on the hardware environment, so the computational cost of deep learning-based algorithms is higher than that of traditional algorithms[19–21]. The background difference method has become the most widely used method for its outstanding superiorities in computation complexity and efficiency, which is the hot spot in moving object detection field[22]. However, the detection results of the background difference method depend on the accuracy of the background model. The way of establishing a robust background model is the key to this method.

There are many methods for moving object detection based on background difference methods, including Gaussian single model (GSM), Gaussian mixture model (GMM), and visual background extractor (ViBe) method[23,24]. ViBe algorithm is a sample-based moving object detection method, which has the advantages of less calculation, small footprint, and fast processing speed. It is suitable for the real-time detection of moving objects. Many researchers are focusing on the ViBe-based method of moving object detection. For example, Talab *et al.*[25] proposed an approach for moving crack detection in video based on ViBe and multiple filtering. Gao and Cheng[26] presented the use of the ViBe algorithm to extract smoke contours and shapes, which finally makes the detection of smoke root more accurate. However, there are some deficiencies of the general ViBe algorithm. For example, when the first frame of the video contains a moving object, there will be a ghost area left in the current location, which will need a long time to be removed. In addition, there is often a shadow problem in moving object detection based on the general ViBe algorithm.

To deal with the problems above for moving object detection based on ViBe method, various improvements have been proposed. For example, Huang *et al.*[27] proposed a moving target detection algorithm based on the improved ViBe algorithm by joining TOM (time of map) mechanism in the process of detection, where both the spatial domain and the time domain information of the pixels were used to eliminate the ghost area. Qiu *et al.*[28] presented a moving object detection method based on the strategy of ViBe algorithm and fused the infrared imaging features, which can establish the pure background in a variety of complex conditions. Yue *et al.*[29] introduced ant colony clustering algorithm and integrated it into the traditional ViBe framework and extended the ViBe based on local modeling to a global modeling algorithm, which can deal with the target adhesion problems but cannot effectively process shadows. The works above improve the performance of the ViBe-based method to some extent. However, few of them considered the problems comprehensively. For example, some methods considered the shadow problem, but they need a long computation time[30,31].

In this paper, an improved ViBe-based approach is proposed, where the problems of moving object detection under natural scenes are fully considered including the ghost area problem, the dynamic background problem, and the shadows problem, and some solutions are presented. Finally, various experiments were conducted under different scenes for moving object detection task. The results show the efficiency and effectiveness of the proposed approach.

The main contributions of this paper are summarized as follows: (1) A new background model based on mode background modeling method is proposed to eliminate the ghost areas quickly; (2) An improved ViBe approach is proposed based on an adaptive foreground detection and background updating method, where the value of the eight neighboring pixels difference between the background and the current frame is used. (3) A novel shadow elimination approach is presented, which is based on the HSV color space combined with the edge detection method. Furthermore, the computation time and background updating mechanism of the proposed approach are discussed.

This paper is organized as follows. Section 2 provides the related works about the ViBe-based method. Section 3 presents the improved ViBe-based method for moving object detection. The moving object detection experiments under various natural scenes are given in Section 4. Section 5 discusses the performance of the proposed approach. Finally, the conclusions are given in Section 6.

2. RELATED WORKS

In the past few years, various foreground target detection methods have been proposed to build powerful and flexible background models that can be used in surveillance scenarios with different challenges. One of the most widely used probabilistic models is the GMM[32], which models each pixel using a mixture of Gaussian models rather than modeling all pixel values as a distribution. For example, Kaewtrakulpong and Pakorn[33] modified the update equation of GMM for improving the accuracy and proposed a shadow detection scheme based on the existing GMM. Hofmann *et al.*[34] used a constantly adapted number of Gaussian distributions of the GMM for each pixel.

As for nonparametric approaches, Barnich and Droogenbroeck[35,36] proposed the ViBe-based method, where the current pixel value is compared to its closest sample within the collection of samples. First, the pixel values of the detected frames are matched with the corresponding models. The threshold value determines whether it belongs to the background or the foreground; for the matching pixel, the background model of the pixel and its neighborhood is updated by a random update mechanism. The method is simple to operate and detects well in static backgrounds but has fixed parameters. This limits the algorithm's ability to adapt to dynamic backgrounds (surface ripples, leaf shaking, etc.), and its neighborhood diffusion update strategy causes slower-moving foreground targets to blend into the background too quickly, increasing false detections. Its single-frame input image initialization strategy creates a "ghost" area when the input image contains foreground targets. In addition, there is often a shadow problem in moving object detection based on the general ViBe algorithm, which affects the accuracy of the background model.

To deal with the problems above for moving object detection based on ViBe method, various improvements have been proposed. For example, Zhu *et al.*[37] proposed a fast and efficient improvement of ViBe algorithm based on the edge characteristic info and neighborhood mean filter, but there are a lot of holes inside the detection area. Chen *et al.*[38] combined physical shadow theory and C1C2C3 color space for the shadow removal. Yang *et al.*[39] used two thresholds to describe the uncertainty in the ViBe-based color video detection, and they used evidence theory to model and handle the uncertainty. Liu *et al.*[40] used the temporal and spatial information of the pixels to initialize the background model, and then combined the background sample set with the neighborhood pixels to determine the complexity of the background and obtain an adaptive segmen-

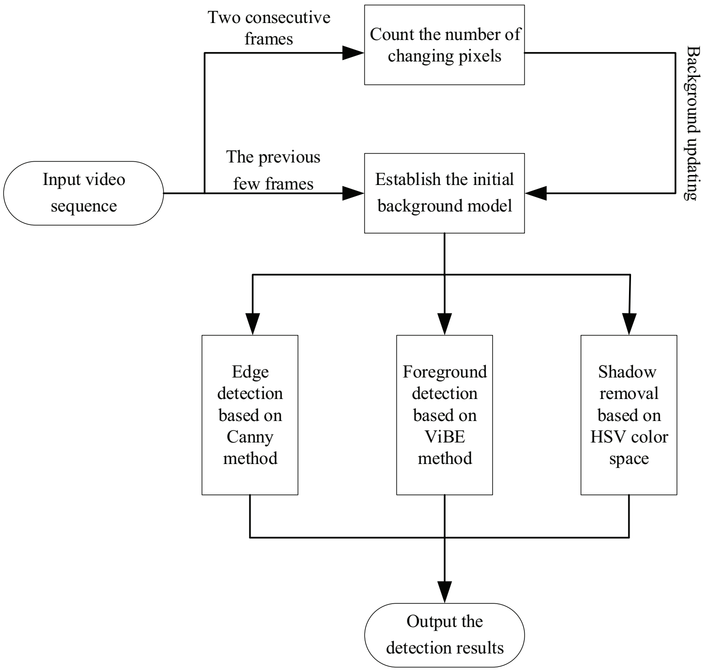

Two consecutive frames
Count the number of changing pixels
Background updating
The previous few frames
Input video sequence
Establish the initial background model
Edge detection based on Canny method
Foreground detection based on ViBE method
Shadow removal based on HSV color space
Output the detection results

Figure 1. Flow diagram of the improved ViBe-based approach for moving object detection.

tation threshold, which can also obtain a better performance in complex dynamic backgrounds but cannot effectively remove shadows.

3. METHODS

In this paper, the problem of moving object detection based on ViBe method is studied. The basic idea of the ViBe method uses neighboring pixels to establish the background model and then compares the background model with the current pixel value to detect the foreground. There are three main steps in the ViBe method, namely the background initialization, the foreground object detection, and the background model updating. Aiming at the problems in the three main steps, some improvements are proposed in this study. The flow chart of the proposed approach is shown in Figure 1 and the main steps of the proposed approach are introduced in detail as follows.

3.1. Mode method based background modeling

The initial background selection is the first step in the ViBe-based method, which will directly influence the detection results. If it can be extracted correctly, the accuracy of the object detection will increase. In general, the ViBe method uses the first frame as the initial background[41], namely

$$B(x, y) = V_F(x, y) \tag{1}$$

where $B(x, y)$ is the pixel value of the background and $V_F(x, y)$ is the pixel value of the first frame in the video. Although the method using the first frame is simple and efficient, it will fail when there is a moving object in the first frame. To deal with this problem, some improvements are proposed, such as the mean method, which needs to store more video frames and has the problem of shadows[42]. In this paper, the mode method is introduced to extract the initial background frame[43]. The basic idea of the mode method for background modeling is that few previous frames are used to obtain an optimized background model. The pixel value of

```
//// The pseudo-code of the Mode Background Method ////

Initialize the parameters; Input the prepared image to matrix $I_m$;
For   $n$ = 1:$N\_Frame$        % N_Frame is first few frames of video;
    $I_y$ = Convert_gray ( $I_m$ )     % Convert the image $I_m$ to a gray image $I_y$;
    $\Theta$ = Save_gray ( $I_y$ )         % Save the grayscale value of $I_y$ to the array;
End for
For   $j$ = 1:$N\_Pixel$              % N_Pixel is the number of pixels in the image;
    $C$ = Ceil ( $\Theta/E$ )
    % Ceil ( ) is a function to rounds the elements of the data to the nearest integers towards
    infinity; E is an integer number.
    $Mf$ = Mode ( $C$ )
    % Mode ( ) is a function to returns the sample mode of C, which is the most frequently
    occurring value in C;
    $Num$ = Count ( $C==Mf$ )
    %Count ( ) is a function to calculate the number of the pixel where C==Mf
    $Mode\_save = Mf$
    % Save the mode value of Mf to the array;
    $R(x, y) = Mode\_save(x, y)*E$
    % Calculate the value of the array;
End for
Return $B(x, y) = R(x, y)$
    % Output the initial background image $I_b$, which is constructed by the pixel B(x, y);
```

Figure 2. The pseudo-code of the mode background method.

the background is calculated by

$$B(x, y) = \frac{\sum_{k=1}^{Num} V_k\,(Mode(C))}{Num} \tag{2}$$

where $Mode(C)$ is a function to return the mode number of the sample, which is the most frequently occurring value in this sample. Num is the number of the mode numbers in the sample. Here, C is defined as follows:

$$C = Ceil(\Theta/E) \tag{3}$$

where $Ceil(\cdot)$ is a function to round the elements of the data to the nearest integers towards infinity. Θ is the grayscale value of the gray image obtained from the original image. To extract most of the numbers appearing in the array Θ by the function $Ceil(\cdot)$, the range of Θ is reduced by dividing an integer E. In this study, E is set as 5, namely the grayscale images are separated with five grayscale levels, which can improve the contrast of different elements in the image and reduce the influence of small speckles on target extraction. The pseudo-code of the mode method for background modeling is shown in Figure 2.

After the background of the video is established, the ViBe method is used to initialize the background model, which is based on the domain model. For each pixel $p(x, y)$ of the background image, the sample set $M(x, y)$ of it is:

$$M(x, y) = \{V_1, V_2, \cdots V_n\}, i = 1, 2, \cdots n \tag{4}$$

where n is the number of the neighboring sample. V_i is the value of a sample that is randomly chosen from the 8-connected neighborhood of each pixel (see Figure 3a). When the sample sets of all the pixels in the background are obtained, a background model is set up.

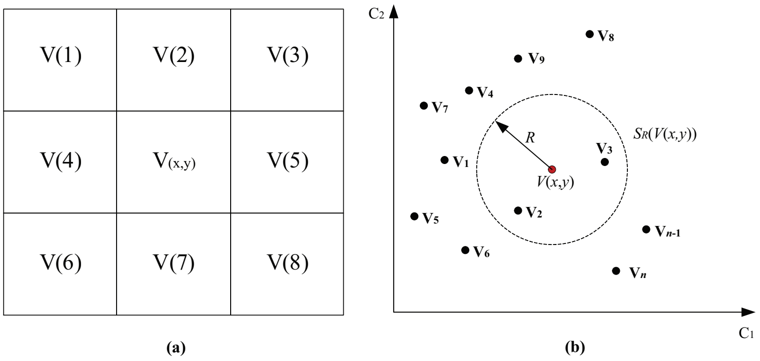


Figure 3. The ViBe-based method: (a) the eight neighbor domain; and (b) the background model of ViBe.

3.2. Adaptive updating mechanism for ViBe method

When the background of the video is established, the next step is to detect the moving objects. The basic discrimination mechanism for the general ViBe method is as follows: for each pixel in the new frame of the video, a sphere $S_R(V(x, y))$ of radius R centered on the value $V(x, y)$ of the pixel is defined (see Figure 3b). Then, the pixel of the new frame can be determined as the background or foreground by[44]:

$$Flag1(x, y) = \begin{cases} 1, & \Psi\{S_R(V(x, y)) \cap M(x, y)\} \leq K \\ 0, & \Psi\{S_R(V(x, y)) \cap M(x, y)\} > K \end{cases} \tag{5}$$

where function $\Psi\{S_R(V(x, y)) \cap M(x, y)\}$ means the cardinality of the set intersection of the sphere $S_R(V(x, y))$ and the collection of $M(x, y)$. K is a threshold. If $Flag1(x, y) = 1$, it means the pixel point $p(x, y)$ belongs to foreground. Otherwise, it means the pixel point $p(x, y)$ belongs to background.

The last step is to randomly update the background model with each new frame. Because of the strong statistical correlation between a pixel and its neighboring pixel, when a pixel is detected as the background pixel, it has a probability of $1/\alpha$ to update model sample set (where α is called update rate). Meanwhile, it also has the probability of $1/\alpha$ to update the background model of neighboring pixels.

From the discrimination mechanism of the original ViBe algorithm in Figure 3a,b, we can see that the detection radius R and the update rate α are two very important parameters. In general, the detection radius R should be larger and the update rate α should be smaller in the dynamic background, to make more pixels be classified as background, and vice versa. However, in the general ViBe algorithm, the values of the parameters R and α are predefined by the designers, which reduce the adaptivity of the ViBe algorithm. Because the value of the eight neighboring pixels difference between the background and the current frame is the factor that can reflect the complex degree of background, it is used to determine the values of the detection radius R and the update rate α adaptively. Namely,

$$R = \begin{cases} R_0 \cdot (1 + a), & a > \tau_0 \\ R_0 \cdot (1 - a), & a \leq \tau_0 \end{cases} \tag{6}$$

$$\alpha = \begin{cases} \alpha_0 \cdot (1 - a), & a > \tau_0 \\ \alpha_0 \cdot (1 + a), & a \leq \tau_0 \end{cases} \tag{7}$$

where R_0 and α_0 are the initial values of the detection radius R and the update rate α; τ_0 is a threshold; and a is a parameter to judge the change of the current scenario, which is calculated by

$$a = \frac{\sum D_{k+1}(x, y)}{N} \tag{8}$$

Here, N is the number of pixels. $D(x, y)$ is the difference of the pixels between two images $I_{k+1}(x, y)$ and $I_k(x, y)$, namely

$$D_{k+1}(x, y) = \begin{cases} 0, & |I_{k+1}(x, y) - I_k(x, y)| < \tau_1 \\ 1, & \text{Otherwise} \end{cases} \tag{9}$$

where τ_1 is a threshold to reduce the effects of the moving objects.

Remark: The mode background method can eliminate the foreground target that appears in the previous frames. The subsequent frames of the video sequence continuously update the "ghost" area to set it as the background, which can effectively speed up the "ghost" area removal.

3.3. Shadow removal strategy

Shadow is a common problem in moving object detection, and how to remove the shadow is a hot topic in the field of computer vision[45,46]. In this paper, an improved method based on the HSV color space is used to complete the shadow removal task. The main reason for using the HSV color space is that it is very close to the characteristics of human vision considering the existing methods, which is more accurate than RGB color space for shadow removal. However, there are many parameters of the traditional HSV that need to be set in different video environments, such as the thresholds used for the shadow judgment[47]. In addition, when there is no significant difference on the color attribute between the moving object and the shaded area, the accuracy of shadow removal based on the traditional HSV color space will be decreased. To deal with these problems, an improved shadow removal strategy is proposed in this paper. The basic idea of the proposed method is that the shadow area can be effectively distinguished by using the characteristics of shadow intensity reduction and color invariance theory, because the HSV color space can directly reflect the color characteristics of the image. The main procedures of the proposed method are as follows:

(1) The HSV space transformation is done. Then, the values of the H, S, and V components are obtained. Since the value V is a direct measure of the brightness of the color, the brightness of these pixels is significantly reduced in the shadow part. The difference of the brightness is denoted as $D_V(x, y)$, which is defined as follows:

$$D_V(x, y) = t_V(x, y)/B_V(x, y) \tag{10}$$

where $t_V(x, y)$ is the V value of current image frame. $B_V(x, y)$ is the V value of background frame. For any pixel points $p(x, y)$, the brightness value between the current frame and background frame is used to determine whether the current pixel is a shadow point. The decision strategy is as follows:

$$Flag2(x, y) = \begin{cases} 1, & \tau_2 \le D_V(x, y) \le \tau_3 \\ 0, & \text{Otherwise} \end{cases} \tag{11}$$

where $Flag2(x, y)$ is a flag. τ_2 and τ_3 are two thresholds for shadow detection.

(2) When the chromaticity of the object is similar to the shadow, the shadow area will be enlarged based on the brightness detection above. To deal with this problem, an improved method is proposed based on the forming mechanism of shadow. Namely, for each shadow pixel $p(x, y)$, its darkness level is limited, because it is darkened for the blocking out of the illumination source, but there is the presence of ambient illumination. In addition, the shadow pixels are mostly in gray areas. The decision strategy is as follows:

$$Flag3(x, y) = \begin{cases} 1, & t_S(x, y) \le \lambda_1 \text{ and } t_V(x, y) \ge \lambda_2 \\ 0, & \text{Otherwise} \end{cases} \tag{12}$$

where $Flag3(x, y)$ is a flag; $t_S(x, y)$ is the saturation of the pixel point of current image frame; λ_1 is the maximum value of the saturation in the gray range; and λ_2 is the minimum value within the gray range. Then, the shadow area can be detected by:

$$I(x, y) = \begin{cases} 1, & Flag2(x, y) = 1 \text{ and } Flag3(x, y) = 1 \\ 0, & \text{Otherwise} \end{cases} \tag{13}$$

At the same time, to ensure the integrity of the foreground targets, the Canny edge detection is performed after finding the different image between the current frame and the background frame[48].

The whole work flow of the proposed approach for the moving objects detection is as follows:

Step1: Initialize the background model based on the mode method.

Step2: Convert the current frame and the background model to gray space, and then detect the foreground objects which include shadows, based on the proposed ViBe algorithm with the adaptive detection radius R and update rate α.

Step3: Convert HSV color space transformation for the current frame and detect the shadows by the color invariance theory at the shadow and the background.

Step4: Carry out an "AND" operation on the results obtained from Steps 2 and 3 to remove the shadow of the foreground targets.

Step5: Find the difference image between the current image frame and the background frame and perform Canny edge detection.

Step6: Carry out an "OR" operation on the results obtained from Steps 4 and 5 to ensure the integrity of the foreground objects.

4. RESULTS

To test the performance of the proposed approach, some experiments were carried out on several benchmark datasets including Highway, Bungalow, Cars, and People[49,50]. These experiments were coded by Python on a computer with 8G RAM and i7-4720HQ 2.60GHz CPU. Seven indices were used to evaluate the performance of detection: recognition rate of foreground (RE), recognition rate of background (SP), false positive rate (FPR), false negative rate (FNR), percentage of wrong classification (PWC), precision (PRE), and F-score (F) (see[51] for the details of these indices). For these indices, the larger are the RE, SP, PRE, F, the more accurate is the detected target area, and the smaller are the FPR, FNR, PWC, the more accurate is the detected background. The values of the parameters used in these experiments are the same and listed in Table 1. To show the efficiency of the proposed improved approach (I-ViBe), it was compared with the Gaussian mixture model based method (GMM) and the general ViBe-based method (G-ViBe). In the general ViBe-based method, the detection radius R and the update rate α are equal to R_0 and α_0 in the proposed approach.

4.1. The experiment for single object detection

To test the basic performance of the proposed approach, two experiments were conducted where only one object was detected. The datasets used for this experiment were Walk (Clip1) and Bungalows (Clip2). Two clips of the two videos were used to test the three detection methods, where the frame with the moving object was used as the detection frame (see Figure 4b). The results of the two experiments are shown in Figure 4. The evaluations for the three methods are listed in Table 2.

The results in Figure 4 show that all the three methods can detect the moving object effectively in this simple experiment, and the results in Table 2 show that the proposed approach has better detection results in most of the indices than the other two methods. In addition, the detection results on Walk (Clip1) show that the general ViBe cannot deal with the ghost problem, while the proposed ViBe can remove the ghost area very well. The detection results on Bungalows (Clip2) show that the proposed ViBe can remove the shadow more

Table 1. Parameters of the proposed method

Parameters	Values	Remarks
E	5	A given threshold in Equation (3)
K	1	A given threshold in Equation (5)
R_0	20	The initial detection radius
α_0	16	The initial update rate
τ_0	0.2	A given threshold in Equations (6) and (7)
τ_1	1	A given threshold in Equation (9)
τ_2	0.2	A given threshold in Equation (11)
τ_3	0.7	A given threshold in Equation (11)
λ_1	43	A given threshold in Equation (12)

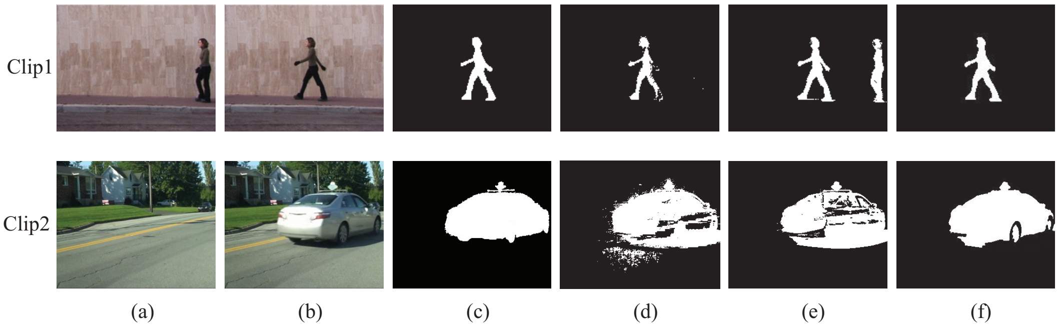


Figure 4. The moving object detection experiments on the video Walk (Clip1) and Bungalows (Clip2): (a) the first frame; (b) the frame for detection; (c) the ground-truth; (d) the result of GMM; (e) the result of G-ViBe; and (f) the result of I-ViBe.

Table 2. The valuation of the three methods for moving object detection in Walk and Bungalows

The valuation indices	The video clip of Walk			The video clip of Bungalows		
	GMM [32]	G-ViBe [35]	I-ViBe	GMM [32]	G-ViBe [35]	I-ViBe
SP	**0.9995**	0.9804	0.9985	0.9310	0.9401	**0.9854**
RE	0.7758	0.9483	**0.9612**	0.8570	0.7999	**0.9636**
FPR	**0.0004**	0.0195	0.0014	0.0689	0.0598	**0.0145**
FNR	0.2241	0.0516	**0.0387**	0.1429	0.2000	**0.0363**
PWC	0.0060	0.0203	**0.0021**	0.0826	0.0879	**0.0186**
PRE	**0.9779**	0.5647	0.9272	0.7395	0.7702	**0.9390**
F	0.8652	0.7079	**0.9493**	0.7939	0.7847	**0.9511**

effectively than the other two methods (see Figure 4e,f).

4.2. The experiment for multiple objects detection

To test the performance of the proposed approach in multiple moving objects detection, two experiments were conducted on the dataset Highway (Clip1) and People (Clip2). The results are shown in Figure 5, and the evaluations for the three methods in this experiment are shown in Table 3.

The results of the experiment on Highway (Clip1) show that there are lots of errors based on the GMM method and the general ViBe method, because there are some leaves shaking in the background having similar color attribute with the vehicles. However, the proposed approach can deal with this problem efficiently, which is

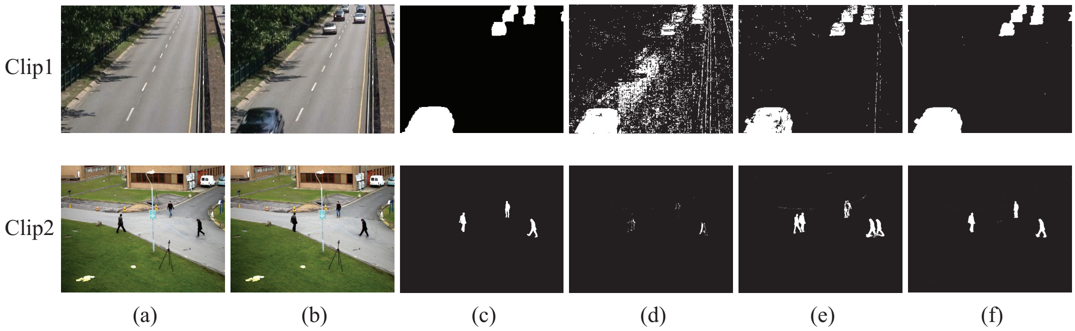


Figure 5. The moving object detection experiments on the video Highway (Clip1) and People (Clip2): (a) the first frame; (b) the frame for detection; (c) the ground-truth; (d) the result of GMM; (e) the result of G-ViBe; and (f) the result of I-ViBe.

Table 3. The valuation of the three methods for moving objects detection on Highway and People

The valuation indices	The video clip1			The video clip2		
	GMM [32]	G-ViBe [35]	I-ViBe	GMM [32]	G-ViBe [35]	I-ViBe
SP	0.8765	0.9910	**0.9979**	**0.9998**	0.9915	0.9992
RE	0.7843	0.8554	**0.9674**	0.1545	0.8616	**0.9849**
FPR	0.1234	0.0089	**0.0020**	**0.0001**	0.0084	0.0007
FNR	0.2156	0.1445	**0.0325**	0.8454	0.1383	**0.0150**
PWC	0.1307	0.0196	**0.0041**	0.0059	0.0097	**0.0008**
PRE	0.3532	0.8918	**0.9713**	0.8834	0.5110	**0.8943**
F	0.4871	0.8732	**0.9694**	0.2630	0.6415	**0.9374**

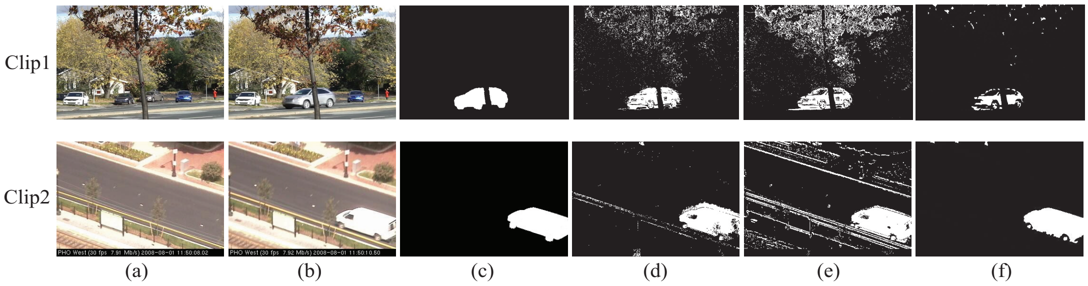


Figure 6. The moving object detection experiments under challenging conditions: (a) the first frame; (b) the frame for detection; (c) the ground-truth; (d) the result of GMM; (e) the result of G-ViBe; and (f) the result of I-ViBe.

combined with the edge information (see Figure 5 and Table 3). Furthermore, there are also ghost problems in the detection results of the experiment on People (Clip2) based on the G-ViBe, because the first frame includes the moving objects (see Figure 5e).

4.3. The experiment under challenging conditions

To further test the performance of the proposed method for moving object detection under some challenging conditions, two extensive experiments were conducted in the dataset of Fall (Clip1) and Boulevard (Clip2), respectively. In the Fall dataset, the background is changing obviously because of the leaves shaking violently. In the Boulevard dataset, the video is blurry due to the shake of the camera. The results of these experiments are shown in Figure 6 and Table 4.

The results in the two experiments show that the performances of all the three methods decrease under dy-

Table 4. The valuation of the three methods for moving object detection under challenging conditions

The valuation indices	The video Clip1			The video Clip2		
	GMM [32]	G-ViBe [35]	I-ViBe	GMM [32]	G-ViBe [35]	I-ViBe
SP	0.9547	0.8765	**0.9960**	0.9850	0.9025	**0.9974**
RE	**0.8755**	0.7496	0.7184	0.8643	0.9258	**0.9574**
FPR	0.0452	0.1234	**0.0039**	0.0149	0.0974	**0.0025**
FNR	**0.1244**	0.2503	0.2815	0.1356	0.0741	**0.0425**
PWC	0.0481	0.1285	**0.0106**	0.0219	0.0957	**0.0051**
PRE	0.4245	0.2016	**0.8202**	0.7819	0.4173	**0.9617**
F	0.5718	0.3178	**0.7659**	0.8210	0.5753	**0.9595**

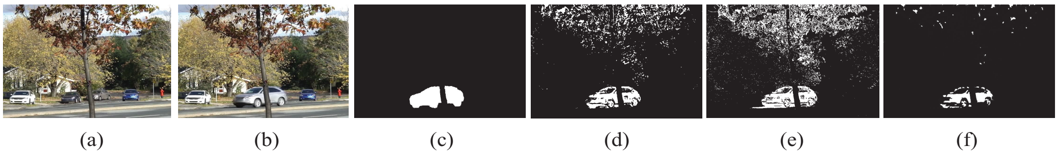

(a) (b) (c) (d) (e) (f)

Figure 7. The moving object detection experiments on the video Fall: (a) the first frame; (b) the frame for detection; (c) the ground-truth; d) the result of G-ViBe; (e) the result of F-ViBe; and (f) the result of I-ViBe.

namic environments. The main reason is that all the three methods are based on the mechanism of background subtraction. However, the performance of the proposed approach does not decrease dramatically compared with other two methods (see the values of PRE and F in Table 4). This performance of the proposed approach is very important for the real application of moving object detection.

5. DISCUSSION

The results presented in Section 3 show that the proposed approach can deal with the ghost area problem and remove the shadow very well. In addition, the evaluation indices of the proposed approach are better than the GMM method and the general ViBe method. In this section, some performances of the proposed approach are discussed, including the computation complexity and the background updating mechanism.

One key part of the ViBe-based approach is the background updating mechanism, so the performance of the improvement in this part for the proposed method is discussed first. An experiment was conducted in the dataset of Fall, where the proposed approach was compared with two methods. The first one is the general ViBe. The second one is a method which has the same parameters and work flow as the proposed approach, except that the background updating mechanism is based on the fixed detection radius and updating rate, and this method is called F-ViBe. The experimental results of Section 3.3 are used as reference, as shown in Figure 7 and Table 5. The experimental results show that the proposed approach can deal with the dynamic environment better than the other two methods. Thus, the background updating mechanism is very efficient for moving object detection under complex environment. In addition, the detection radius and updating rate of the F-ViBe method are given by the designer, which need more experience and time.

Another important index of the moving object detection method is the real time problem, because the speed of the moving object is very high sometimes. The proposed approach has two main differences with the general ViBe method, the background modeling and updating mechanism and the shadow removal strategy. Thus, the time needed in all the three experiments of Clip1 in Section 3 is divided into two parts, the time for the background modeling and the time for moving object detection (including background updating).

Table 5. The moving object detection experiments based on different background discrimination mechanism

The valuation indices	G-ViBe [35]	F-ViBe	I-ViBe
SP	0.8765	0.9738	**0.9960**
RE	**0.7496**	0.7394	0.7184
FPR	0.1234	0.0261	**0.0039**
FNR	**0.2503**	0.2605	0.2815
PWC	0.1285	0.0328	**0.0106**
PRE	0.2016	0.4551	**0.8202**
F	0.3178	0.5634	**0.7659**

Table 6. The moving object detection experiments based on different background discrimination mechanism

The video	Computation time (s)	GMM [32]	G-ViBe [35]	I-ViBe
Clip1 of Section 4.1	background modeling		**0.4042**	1.9662
$(180 * 144)$	object detection	**0.0337**	0.1028	0.1508
Clip1 of Section 4.2	background modeling		**0.4077**	2.0265
$(320 * 240)$	object detection	**0.0353**	0.1049	0.2503
Clip1 of Section 4.3	background modeling		**0.4966**	2.2822
$(720 * 480)$	object detection	**0.0798**	0.1229	0.4138

The results in Table 6 show that more time for the object detection is needed using G-ViBe and I-ViBe than the GMM method, because the GMM method selects the initial background frame randomly. For high resolution videos, the proposed ViBe method takes more time to compare the values of pixels in each channel of the HSV space, so the time for object detection increases. In addition, the results show that more time is needed in the ViBe based approach during the background modeling process, which can be off-line proceeded and will not affect the real-time moving object detection. For off-line processing, multiple images of the detection area can be collected in advance, and the mode background method can be used for modeling. In the subsequent detection tasks, there is no need to repeat the modeling. Thus, the proposed approach has a better comprehensive performance than both the GMM method and the G-ViBe method, although the computation time of the proposed approach is relatively higher than the other two methods, which is a problem for further study.

6. CONCLUSIONS

In this paper, we present an improved moving object detection approach based on ViBe algorithm. During the process of foreground region extraction, the initial background is obtained by the previous few frames and then updated by the value of the eight neighboring pixel difference between the background and the current frame. In addition, a shadow removal strategy is adopted by combining the HSV color space and the edge information. Most of the parameters in the proposed method are calculated adaptively, which is very important for the adaptivity of moving object detection method. The experiments showed that the proposed approach can deal with moving object detection efficiently in various situations, such as the severe shadow problems in the foreground and the presence of moving objects in the first frame. In addition, the proposed approach can be used for real-time moving object detection. In future work, some more efficient methods based on artificial intelligence algorithms should be studied to improve the accuracy and real-time ability for moving object detection.

DECLARATIONS

Authors' contributions

Funding acquisition: Ni J
Project administration: Ni J, Shi P
Writing-original draft: Tang G
Writing-review and editing: Li Y, Zhu J

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work has been supported by the National Natural Science Foundation of China (61873086) and the Science and Technology Support Program of Changzhou (CE20215022).

Conflicts of interest

All authors declared that they have no conflicts of interest to this work.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Ni J, Zhang X, Shi P, Zhu J. An Improved kernelized correlation filter based visual tracking method. *Mathematical Problems in Engineering* 2018;2018:1-12. [DOI: 10.1155/2018/6931020]
2. Zhang Z, Chen B, Yang M. Moving target detection based on time reversal in a multipath environment. *IEEE Trans Aerosp Electron Syst* 2021;57:3221-36. [DOI: 10.1109/TAES.2021.3074131]
3. Ni J, Yang L, Wu L, Fan X. An improved spinal neural system-based approach for heterogeneous AUVs cooperative hunting. *Int J Fuzzy Syst* 2018;20:672-86. [DOI: 10.1007/s40815-017-0395-x]
4. Tivive FHC, Bouzerdoum A. Toward moving target detection in through-the-wall radar imaging. *IEEE Trans Geosci Remote Sensing* 2021;59:2028-40. [DOI: 10.1109/TGRS.2020.3005199]
5. Ni J, Gong T, Gu Y, Zhu J, Fan X. An improved deep residual network-based semantic simultaneous localization and mapping method for monocular vision robot. *Comput Intell Neurosci* 2020;2020:7490840. [DOI: 10.1155/2020/7490840]
6. Lu X, Mao X, Liu H, Meng X, Rai L. Event camera point cloud feature analysis and shadow removal for road traffic sensing. *IEEE Sensors J* 2022;22:3358-69. [DOI: 10.1109/JSEN.2021.3138736]
7. Negri P. Estimating the queue length at street intersections by using a movement feature space approach. *IET Image Processing* 2014;8:406-16. [DOI: 10.1049/iet-ipr.2013.0496]
8. Sengar SS, Mukhopadhyay S. Motion detection using block based bi-directional optical flow method. *Journal of Visual Communication and Image Representation* 2017;49:89-103. [DOI: 10.1016/j.jvcir.2017.08.007]
9. Chen Y, Wang J, Zhu B, Tang M, Lu H. Pixelwise deep sequence learning for moving object detection. *IEEE Trans Circuits Syst Video Technol* 2019;29:2567-79. [DOI: 10.1109/TCSVT.2017.2770319]
10. Li J, Pan ZM, Zhang ZH, Zhang H. Dynamic ARMA-based background subtraction for moving objects detection. *IEEE Access* 2019;7:128659-68. [DOI: 10.1109/ACCESS.2019.2939672]
11. Garcia-Garcia B, Bouwmans T, Silva AJR. Background subtraction in real applications: challenges, current models and future directions. *Computer Science Review* 2020;35:100204. [DOI: 10.1016/j.cosrev.2019.100204]
12. Cristani M, Farenzena M, Bloisi D, Murino V. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP J Adv Signal Process* 2010;2010. [DOI: 10.1155/2010/343057]
13. Javed S, Narayanamurthy P, Bouwmans T, Vaswani N. Robust PCA and robust subspace tracking: a comparative evaluation. In: 2018 IEEE Statistical Signal Processing Workshop (SSP). Freiburg im Breisgau, Germany; 2018. pp. 836–40. [DOI: 10.1109/SSP.2018.8450718]
14. Mandal M, Vipparthi SK. An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs. IEEE Transactions on Intelligent Transportation Systems 2021:Article in Press. [DOI:

10.1109/TITS.2021.3077883]
15. Minematsu T, Shimada A, Uchiyama H, Taniguchi R. Analytics of deep neural network-based background subtraction. *J Imaging* 2018;4:78. [DOI: 10.3390/jimaging4060078]
16. Giraldo JH, Javed S, Sultana M, Jung SK, Bouwmans T. The emerging field of graph signal processing for moving object segmentation. In: International Workshop on Frontiers of Computer Vision. Virtual, Online; 2021. pp. 31–45. [DOI: 10.1007/978-3-030-81638-4-3]
17. Giraldo JH, Javed S, Bouwmans T. Graph Moving Object Segmentation. *IEEE Trans Pattern Anal Mach Intell* 2022;44:2485-503. [DOI: 10.1109/TPAMI.2020.3042093]
18. Sengar SS, Mukhopadhyay S. Moving object area detection using normalized self adaptive optical flow. *Optik* 2016;127:6258-67. [DOI: 10.1016/j.ijleo.2016.03.061]
19. Ni J, Chen Y, Chen Y, et al. A survey on theories and applications for self-driving cars based on deep learning methods. *Applied Sciences* 2020;10:2749. [DOI: 10.3390/app10082749]
20. Wang Y, Zhu L, Yu Z. Foreground detection for infrared videos with multiscale 3-D fully convolutional network. *IEEE Geosci Remote Sensing Lett* 2019;16:712-6. [DOI: 10.1109/LGRS.2018.2881053]
21. Ni J, Shen K, Chen Y, Cao W, Yang SX. An improved deep network-based scene classification method for self-driving cars. *IEEE Trans Instrum Meas* 2022;71:1-14. [DOI: 10.1109/TIM.2022.3146923]
22. Mahmoudabadi H, Olsen MJ, Todorovic S. Detecting sudden moving objects in a series of digital images with different exposure times. *Computer Vision and Image Understanding* 2017;158:17-30. [DOI: 10.1016/j.cviu.2017.01.004]
23. Wang C, Cheng J, Chi W, Yan T, Meng MQH. Semantic-aware informative path planning for efficient object search using mobile robot. *IEEE Trans Syst Man Cybern, Syst* 2021;51:5230-43. [DOI: 10.1109/TSMC.2019.2946646]
24. Liu Z, An D, Huang X. Moving target shadow detection and global background reconstruction for VideoSAR based on single-frame imagery. *IEEE Access* 2019;7:42418-25. [DOI: 10.1109/ACCESS.2019.2907146]
25. Talab AMA, Huang Z, Xi F, Haiming L. Moving crack detection based on improved VIBE and multiple filtering in image processing techniques. *IJSIP* 2015;8:275-86. [DOI: 10.14257/ijsip.2015.8.2.27]
26. Gao Y, Cheng P. Full-scale video-based detection of smoke from forest fires combining ViBe and MSER algorithms. *Fire Technol* 2021;57:1637-66. [DOI: 10.1007/s10694-020-01052-3]
27. Huang W, Liu L, Yue C, Li H. The moving target detection algorithm based on the improved visual background extraction. *Infrared Physics & Technology* 2015;71:518-25. [DOI: 10.1016/j.infrared.2015.06.011]
28. Qiu S, Tang Y, Du Y, Yang S. The infrared moving target extraction and fast video reconstruction algorithm. *Infrared Physics & Technology* 2019;97:85-92. [DOI: 10.1016/j.infrared.2018.11.025]
29. Yue Y, Xu D, Qian Z, Shi H, Zhang H. AntViBe: improved vibe algorithm based on ant colony clustering under dynamic background. *Mathematical Problems in Engineering* 2020;2020:1-13. [DOI: 10.1155/2020/7478626]
30. Nagarathinam K, Kathavarayan RS. Moving shadow detection based on stationary wavelet transform and zernike moments. *IET Computer Vision* 2018;12:787-95. [DOI: 10.1049/iet-cvi.2017.0273]
31. Khare M, Srivastava RK, Khare A. Moving shadow detection and removal-a wavelet transform based approach. *IET Computer Vision* 2014;8:701-17. [DOI: 10.1049/iet-cvi.2014.0028]
32. Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. In: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99). vol. 2. Fort Collins, CO, USA; 1999. pp. 246–52. [DOI: 10.1109/CVPR.1999.784637]
33. KaewTraKulPong P, Bowden R. In: An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. Boston, MA: Springer US; 2002. pp. 135–44. [DOI: 10.1007/978-1-4615-0913-4-11]
34. Hofmann M, Tiefenbacher P, Rigoll G. Background segmentation with feedback: The Pixel-Based Adaptive Segmenter. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. Providence, RI, United states; 2012. pp. 38–43. [DOI: 10.1109/CVPRW.2012.6238925]
35. Barnich O, Van Droogenbroeck M. ViBE: a powerful random technique to estimate the background in video sequences. In: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. Taipei, Taiwan; 2009. pp. 945–48. [DOI: 10.1109/ICASSP.2009.4959741]
36. Barnich O, Van Droogenbroeck M. ViBe: a universal background subtraction algorithm for video sequences. *IEEE Trans Image Process* 2011;20:1709-24. [DOI: 10.1109/TIP.2010.2101613]
37. Zhu F, Jiang P, Wang Z. ViBeExt: The extension of the universal background subtraction algorithm for distributed smart camera. In: 2012 International Symposium on Instrumentation Measurement, Sensor Network and Automation (IMSNA). vol. 1. Sanya, Hainan, China; 2012. pp. 164–68. [DOI: 10.1109/MSNA.2012.6324539]
38. Chen F, Zhu B, Jing W, Yuan L. Removal shadow with background subtraction model ViBe algorithm. In: 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA). Toronto, ON, Canada; 2013. pp. 264–69. [DOI: 10.1109/IMSNA.2013.6743265]
39. Yang Y, Han D, Ding J, Yang Y. An improved ViBe for video moving object detection based on evidential reasoning. In: 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). Baden-Baden, Germany: IEEE; 2016. pp. 26–31. [DOI: 10.1109/MFI.2016.7849462]
40. Liu L, Chai Gh, Qu Z. Moving target detection based on improved ghost suppression and adaptive visual background extraction. *J Cent South Univ* 2021;28:747-59. [DOI: 10.1007/s11771-021-4642-9]
41. Bo G, Kefeng S, Daoyin Q, Hongtao Z. Moving object detection based on improved ViBe algorithm. *IJSEIA* 2015;9:225-32. [DOI:

10.14257/ijsh.2015.9.12.23]

42. Zhang X, Liu K, Wang X, Yu C, Zhang T. Moving Shadow Removal Algorithm Based on HSV Color Space. *TELKOMNIKA* 2014;12. [DOI: 10.11591/telkomnika.v12i4.4991]
43. Zhang B, Jiao D, Lv X. A target detection algorithm for SAR images based on regional probability statistics and saliency analysis. *International Journal of Remote Sensing* 2019;40:1394-410. [DOI: 10.1080/01431161.2018.1524593]
44. Tian Y, Wang D, Jia P, Liu J. Moving Object Detection with ViBe and Texture Feature. In: Pacific Rim Conference on Multimedia. Xi'an, China; 2016. pp. 150–59. [DOI: 10.1007/978-3-319-48890-5_15]
45. Liu Z, Yin H, Mi Y, Pu M, Wang S. Shadow Removal by a Lightness-Guided Network With Training on Unpaired Data. *IEEE Trans Image Process* 2021;30:1853-65. [DOI: 10.1109/TIP.2020.3048677]
46. Hu X, Fu CW, Zhu L, Qin J, Heng PA. Direction-aware spatial context features for shadow detection and removal. *IEEE Trans Pattern Anal Mach Intell* 2020;42:2795-808. [DOI: 10.1109/TPAMI.2019.2919616]
47. Huang W, Kim K, Yang Y, Kim YS. Automatic Shadow Removal by Illuminance in HSV Color Space. *csit* 2015;3:70-5. [DOI: 10.13189/csit.2015.030303]
48. Long Z, Zhou X, Zhang X, Wang R, Wu X. Recognition and classification of wire bonding joint via image feature and SVM model. *IEEE Trans Compon, Packag Manufact Technol* 2019;9:998-1006. [DOI: 10.1109/TCPMT.2019.2904282]
49. Goyette N, Jodoin PM, Porikli F, Konrad J, Ishwar P. Changedetection.net: A new change detection benchmark dataset. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. Providence, RI, USA; 2012. pp. 1–8. [DOI: 10.1109/CVPRW.2012.6238919]
50. Zhang H, Qian Y, Wang Y, Chen R, Tian C. A ViBe Based Moving Targets Edge Detection Algorithm and Its Parallel Implementation. *Int J Parallel Prog* 2020;48:890-908. [DOI: 10.1007/s10766-019-00628-z]
51. Zhang E, Li Y, Duan J. Moving object detection based on confidence factor and CSLBP features. *The Imaging Science Journal* 2016;64:253-61. [DOI: 10.1080/13682199.2016.1168977]

Boin *et al*. *Intell Robot* 2022;2(2):145-67
DOI: 10.20517/ir.2022.11

Intelligence & Robotics

Research Article

Open Access

Check for updates

AVDDPG – Federated reinforcement learning applied to autonomous platoon control

Christian Boin, Lei Lei, Simon X. Yang

School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada.

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: Boin C, Lei L, Yang SX. AVDDPG – Federated reinforcement learning applied to autonomous platoon control. *Intell Robot* 2022,2(2):145-66. http://dx.doi.org/10.20517/ir.2022.11

Received: 27 Mar 2022 **First Decision:** **Revised:** **Accepted:** 20 May 2022 **Published:** 30 May 2022

Academic Editors: Xin Xu, Wai Lok Woo **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

Abstract

Since 2016 federated learning (FL) has been an evolving topic of discussion in the artificial intelligence (AI) research community. Applications of FL led to the development and study of federated reinforcement learning (FRL). Few works exist on the topic of FRL applied to autonomous vehicle (AV) platoons. In addition, most FRL works choose a single aggregation method (usually weight or gradient aggregation). We explore FRL's effectiveness as a means to improve AV platooning by designing and implementing an FRL framework atop a custom AV platoon environment. The application of FRL in AV platooning is studied under two scenarios: (1) Inter-platoon FRL (Inter-FRL) where FRL is applied to AVs across different platoons; (2) Intra-platoon FRL (Intra-FRL) where FRL is applied to AVs within a single platoon. Both Inter-FRL and Intra-FRL are applied to a custom AV platooning environment using both gradient and weight aggregation to observe the performance effects FRL can have on AV platoons relative to an AV platooning environment trained without FRL. It is concluded that Intra-FRL using weight aggregation (Intra-FRLWA) provides the best performance for controlling an AV platoon. In addition, we found that weight aggregation in FRL for AV platooning provides increases in performance relative to gradient aggregation. Finally, a performance analysis is conducted for Intra-FRLWA versus a platooning environment without FRL for platoons of length 3, 4 and 5 vehicles. It is concluded that Intra-FRLWA largely out-performs the platooning environment that is trained without FRL.

Keywords: Deep reinforcement learning, autonomous driving, federated reinforcement learning, platooning

1. INTRODUCTION

In recent years, federated learning (FL) and its extension federated reinforcement learning (FRL) have become a popular topic of discussion in the artificial intelligence (AI) community. The concept of FL was first proposed by Google with the development of the federated averaging (FedAvg) aggregation method[1]. FedAvg provided an increase in the performance of distributed systems while also providing privacy advantages when compared to centralized architectures for supervised machine learning (ML) tasks[1–3]. FL's core ideology was initially motivated by the need to train ML models from distributed data sets across mobile devices while minimizing data leakage and network usage[1].

Research on the topics of reinforcement learning (RL) and deep reinforcement learning (DRL) has made great progress over the years; however, there remain important challenges for ensuring the stable performance of DRL algorithms in the real world. DRL processes are often sensitive to small changes in the model space or hyper-parameter space, and as such the application of a single trained model across similar systems often leads to control inaccuracies or instability[4,5]. In order to overcome the stability challenges that DRL poses, often a model must be manually customized to accommodate the finite differences amongst similar agents in a distributed system. FRL aims to overcome the aforementioned issues by allowing agents to share private information in a secure way. By utilizing an aggregation method, such as FedAvg[1], systems with many agents can have decreased training times with increased performance.

Despite the popularity of FL and FRL, to the best of our knowledge at the time of this study, there are no works applying FRL to platoon control. In general, there are two types of "models" for AV decision making: vehicle-following modeling and lane-changing modeling[6]. For the purposes of this study, the vehicle-following approach known as co-operative adaptive cruise control (CACC) is explored. Vehicle following models are based on following a vehicle on a single lane road with respect to a leading vehicle's actions[7]. CACC is a multi-vehicle control strategy where vehicles follow one another in a line known as a platoon, while simultaneously transmitting vehicle data amongst each other[8]. CACC platoons have been proven to improve traffic flow stability, throughput and safety for occupants[8,9]. Traditionally controlled vehicle following models have limited accuracy, poor generalization from a lack of data, and a lack of adaptive updating[7].

We are motivated by the current state-of-the-art for CACC AV Platoons, along with previous works related to FRL, to apply FRL to the AV platooning problem and observe the performance benefits it may have on the system. We propose an FRL framework built atop a custom AV platooning environment in order to analyse FRL's suitability for improving AV platoon performance. In addition, two approaches are proposed for applying FRL amongst AV platoons. The first proposed method is inter-platoon FRL (Inter-FRL), where FRL is applied to AVs across different platoons. The second proposed method is intra-platoon FRL (Intra-FRL), where FRL is applied to AVs within the same platoon. We investigate the possibility of Inter-FRL and Intra-FRL as a means to increase performance using two aggregation methods: averaging model weights and averaging gradients. Furthermore, the performance of Inter-FRL and Intra-FRL using both aggregation methods is studied relative to platooning environments trained without FRL (no-FRL). Finally, we compare the performance of Intra-FRL with weight averaging (Intra-FRLWA) against a platooning environment trained without FRL for platoons of length 3, 4 and 5 vehicles.

1.1. Related works

In this subsection, the current state-of-the-art is presented for FRL and DRL applied to AV's. In addition the contributions of this paper are presented.

1.1.1. Federated reinforcement learning

There are two main areas of research in FRL currently: horizontal federated reinforcement learning (HFRL), and vertical federated reinforcement learning (VFRL). HFRL has been selected as the algorithm of choice for the purposes of this study. HFRL and VFRL differ with respect to the structure of their environments and aggregation methods. All agents in an HFRL architecture use isolated environments. It follows that each agent's action in an HFRL system has no effect on the other agents in the system. An HFRL architecture proposes the following training cycle for each agent: first, a training step is performed locally, second, environment specific parameters are uploaded to the aggregation server, and lastly, parameters are aggregated according to the aggregation method and returned to each agent in the system for another local training step. HFRL may be noted to have similarities to "Parallel RL". Parallel RL is a long studied field of RL, where agent gradients are transferred amongst each other[5,10,11].

Reinforcement learning is often a sequential learning process, and as such data is often non-IID with a small sample space[12]. HFRL provides the ability to aggregate experience while increasing the sample efficiency, thus providing more accurate and stable learning[13]. Some of the current works applying HFRL to a variety of applications are summarized below.

A study by Lim *et al.* aims to increase the performance of RL methods applied to multi-IoT device systems. RL models trained on single devices are often unable to control devices in a similar albeit slightly different environment[5]. Currently, multiple devices need to be trained separately using separate RL agents[5]. The methods proposed by Lim *et al.* sped up the learning process by 1.5 times for a two agent system. In a study by Nadiger *et al.*, the challenges in the personalization of dialogue managers, smart assistants and more are explored. RL has proven to be successful in practice for personalized experiences; however, long learning times and no sharing of data limit the ability for RL to be applied at scale. Applying HFRL to atari non-playable characters in pong showed a median improvement of 17% for the personalization time[10]. Lastly, Liu *et al.* discuss RL as a promising algorithm for smart navigation systems, with the following challenges: long training times, poor generalization across environments, and storing data over long periods of time[14]. In order to address these problems, Liu *et al.* proposed the architecture 'Lifelong FRL', which can be categorized as an HFRL problem. It is found the Lifelong FRL increased the learning rate for smart navigation system when tested on robots in a cloud robotic system[14].

The successes of the FedAvg algorithm as a means to improve performance and training times for systems have inspired further research into how aggregation methods should be applied. The design of the aggregation method is crucial in providing performance benefits to that of the base case where FRL is not applied. The FedAvg[3] algorithm proposed the averaging of gradients in the aggregation method. In contrast, Liang *et al.* proposed using model weights in the aggregation method for AV steering control[15]. Thus, FRL applications can differ based upon the selection of which parameter to use in the aggregation method. A study by Zhang *et al.* explores applying FRL to a decentralized DRL system optimizing cellular vehicle-to-everything communication[16]. Zhang *et al.* utilize model weights in the aggregation method, and describe a weighting factor dividing the sum batch size for all agents by the training batch size for a specific agent[16]. In addition, the works of Lim *et al.* explore how FRL using gradient aggregation can improve convergence speed and performance on the OpenAI-gym environments CartPole-V0, MountainvehicleContinuous-V0, Pendulum-V0 and Acrobot-V1[17]. Lim *et al.* determined that aggregating gradients using FRL creates high performing agents for each of the OpenAI-gym environments relative to models trained without FRL[17]. In addition, Wang *et al.* apply FRL to heterogeneous edge caching[18]. Wang *et al.* show the effectiveness of FRL using weight aggregation to improve hit rate, reduce average delays in the network and offload traffic[18]. Lastly, Huang *et al.* apply FRL using model weight aggregation to Service Function Chains in network function virtualization enabled networks[19]. Huang *et al.* observe that FRL using model weight aggregation provides benefits to convergence speed, average reward and average resource consumption[19].

Despite the differences in FRL applications within the aforementioned studies, each study maintains a similar goal: to improve the performance of each agent within the system. None of the aforementioned works explore the differences in whether gradient or model weight aggregation is favourable in performance, and many of the works apply FRL to distributed network or communications environments. It is the goal of this study to conclude whether model weight or gradient aggregation is favourable for AV platooning, as well as be one of the first (if not the first) to apply FRL to AV platooning.

1.1.2. Deep reinforcement learning applied to AV platooning

In recent years, there has been a surge in autonomous vehicle (AV) research, likely due to the technologies potential for increasing road safety, traffic throughput and fuel economy[6,20]. Two areas of research are often considered when delving into an AV model: supervised learning or RL[20]. Driving is considered a multi-agent interaction problem, and due to the large variability of road data, it can be quite challenging (or near impossible) to gather a data set variable enough to train a supervised model[21]. Driving data is collected from humans, which can also limit an AI's ability to that of human level[6]. In contrast, RL methods are known to generalize quite well[20]. RL approaches are model-free and a model may be inferred by the algorithm while training.

In order to improve the limitations of vehicle following models, DRL has been a steady area of research in the AV community, with many authors contributing works to DRL applied to CACC[8,9,22,23]. In a study by Lin *et al.*, a DRL framework is designed to control a CACC AV platoon[22]. The DRL framework uses the deep deterministic policy gradient (DDPG)[24] algorithm and is found to have near-optimal performance[22]. In addition, Peake *et al.* identify limitations in platooning with regard to the communication in platooning[23]. Through the application of a multi-agent reinforcement learning process, i.e. a policy gradient RL and LSTM network, the performance of a platoon containing 3-5 vehicles is improved upon that of current RL applications to platooning[23]. Furthermore, Model Predictive Control (MPC) is the current state-of-the-art for real-time optimal control practices[25]. The study performed by Lin *et al.* applies both MPC and DRL methodologies to the AV platoon problem, observing a DRL model trained using the DDPG algorithm produces merely a 5.8% episodic cost higher than the current state-of-the-art[25]. The works of Yan *et al.* propose a hybrid approach to the AV platooning problem where the platoon is modeled as a Markov Decision Process (MDP) in order to collect two rewards from the system at each time step simultaneously[26]. This approach also incorporates jerk, the rate of change of acceleration in the calculation of the reward for each vehicle in order to ensure passenger comfort[26]. The hybrid strategy led to increased performance to that of the base DDPG algorithm, as the proposed framework switches between using classic CACC modeling and DDPG depending on the performance degradation of the DDPG algorithm[26]. In another study by Zhu *et al.*, a DRL model is formulated and trained using DDPG to be evaluated against real world driving data. Parameters such as time to collision, headway, and jerk were considered in the DRL model's reward function[27]. The DDPG algorithm provided favourable performance to that of the analysed human driving data, with regard to more efficient driving via reduced vehicle headways, and improved passenger comfort with lower magnitudes of jerk[27]. As Vehicle-to-Everything (V2X) communications are envisioned to have a beneficial impact on the performance of platoon controllers, the works of Lei *et al.* investigates the value of V2X communications for DRL-based platoon controllers. Lei *et al.* emphasizes the trade-off between the gain of including exogenous information in the system state for reducing uncertainty and the performance erosion due to the curse-of-dimensionality[28].

When formulating the AV platooning problem as a DRL model DDPG is prominently selected as the algorithm for training. DDPG's ability to handle continuous actions space and complex state's is perfect for the CACC platoon problem. However, despite the DDPG algorithm's success in literature, there are still instability challenges related to the algorithm along with a time consuming hyper-parameter tuning process to account for the minute differences in vehicle models/dynamics amongst platoons. As previously discussed, FRL provides advantages in these areas where information sharing can accelerate performance during training and improve the performance of the system as a whole. In addition, the ability to share experience across like models has been proven to allow for fast convergence of models, which further optimizes the performance of DDPG when applied to AV platoons[5].

1.2. Contributions

To the best of our knowledge, no works at the time of this study existed covering the specific topic of FRL applied to platoon control. Many of the works existing on FRL have shown the benefits of FRL with regard to the increased rate of convergence and overall system performance with distributed networks, edge caching and communications[16–19]. Furthermore, of the works cited in this study, the works closely related to FRL for platoon control are those of Peake *et al.* and Liang *et al.*[15,23]. In contrast to Liang *et al.*, where FedAvg is applied successfully to control the steering angle of a single vehicle, we apply FRL to an AV platooning problem where the control of multiple vehicles' positions and spacing are required[15]. Peake *et al.* explore multi-agent reinforcement learning and its ability to improve the performance of AV platoons experiencing communication delays[23]. Although Peake *et al.* are also successful in their approach, there is no specific reference to FRL in the paper[23]. In addition, a variety of existing works on FRL choose to use either gradients or model weights in the FRL aggregation method. This study explores how both aggregation methods can provide benefits to the AV platooning problem and, most importantly, which provides a better result. Finally, this study further distinguishes its approach from existing literature by declaring two possible ways to apply FRL to AV platooning:

1. Intra-FRL: where multi-vehicle platoons share data during training to increase the performance of vehicles within the same platoon.
2. Inter-FRL: where multi-vehicle platoons share data during training across platoons amongst vehicles in the exact same platoon position to increase performance.

In contrast to existing literature, where it is common to average the parameters across each model in the system, for Intra-FRL, we propose a directional averaging where follower vehicles incorporate the preceding vehicle parameters in the computation of the gradients or weights. Thus, in Intra-FRL, the leading vehicle trains independently of those following. The AV platoon provides a unique playground environment suitable for exploring the suitability of FRL as a means to increase the performance of systems with regard to convergence rate and performance.

2. PROPOSED FRAMEWORK

In this section, a state space model is formulated and presented for the AV platooning problem. Next, the MDP model is presented, outlining the platoon system's state space, action space and reward function. Lastly, the FRL DDPG algorithm design and application to AV platooning are described.

2.1. CACC CTHP model formulation

Consider a platoon P of vehicles $\mathcal{V} = V_1, V_2, ..., V_n$ where the leader of the platoon is V_1.

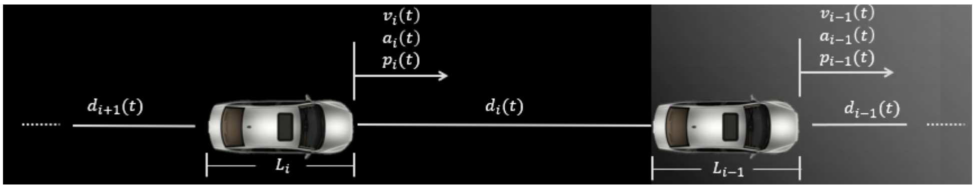


Figure 1. An example platoon modeled with system parameters.

As illustrated in Figure 1, for a general vehicle (V_i), the position of V_i's front bumper is defined as p_i. The velocity, acceleration and control input of V_i are denoted as v_i, a_i and u_i. Furthermore, the acceleration of V_i's predecessor may be denoted as a_{i-1}. The control input for V_i is defined as u_i (whether V_i should accelerate or decelerate). V_i's drive-train dynamics coefficient is defined as τ_i, where large values of τ_i indicate larger response times for a given input u_i to generate acceleration a_i. Lastly, the length of V_i is denoted as L_i. The system dynamics for V_i are thus provided below as

$$\begin{aligned} \dot{p_i}(t) &= v_i(t) \\ \dot{v_i}(t) &= a_i(t) \\ \dot{a_i}(t) &= -\frac{1}{\tau_i}a_i(t) + \frac{1}{\tau_i}u_i(t) \\ \dot{a_{i-1}}(t) &= -\frac{1}{\tau_{i-1}}a_{i-1}(t) + \frac{1}{\tau_{i-1}}u_{i-1}(t) \end{aligned} \tag{1}$$

The headway $d_i(t)$ in a CACC model is the positional difference of the current vehicle relative to the rear bumper of its leader, which can be derived as[22,29]

$$d_i(t) = p_{i-1}(t) - p_i(t) - L_{i-1}. \tag{2}$$

In addition, the desired headway $d_{r,i}(t)$ is defined as

$$d_{r,i}(t) = r_i + h_i v_i(t), \tag{3}$$

where r_i is the standstill distance, and h_i is the time-gap for V_i to maintain relative to it's predecessor V_{i-1}. The position error e_{pi} and the velocity error e_{vi} are defined as:

$$\begin{aligned} e_{pi}(t) &= d_i(t) - d_{r,i}(t) \\ e_{vi}(t) &= v_{i-1}(t) - v_i(t) \end{aligned} \tag{4}$$

Therefore, the state of V_i can be defined as $x_i(t) = \begin{bmatrix} e_{pi}(t) & e_{vi}(t) & a_i(t) & a_{i-1}(t) \end{bmatrix}^\top$, and the derivative of the state is:

$$\begin{aligned} \dot{e_{pi}}(t) &= e_{vi}(t) - h_i a_i(t), \\ \dot{e_{vi}}(t) &= a_{i-1}(t) - a_i(t), \\ \dot{a_i}(t) &= -\frac{1}{\tau_i}a_i(t) + \frac{1}{\tau_i}u_i(t), \\ \dot{a_{i-1}}(t) &= -\frac{1}{\tau_{i-1}}a_{i-1}(t) + \frac{1}{\tau_{i-1}}u_{i-1}(t). \end{aligned} \tag{5}$$

The state space formula for V_i is thus given as

$$\dot{x_i}(t) = A_i x_i(t) + B_i u_i(t) + C_i u_{i-1}(t), \tag{6}$$

where A_i, B_i, and C_i are defined below as

$$A_i = \begin{bmatrix} 0 & 1 & -h_i & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -\frac{1}{\tau_i} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_{i-1}} \end{bmatrix} \qquad B_i = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau_i} \\ 0 \end{bmatrix} \qquad C_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\tau_{i-1}} \end{bmatrix}. \tag{7}$$

2.2. MDP model formulation

The AV platooning problem can be formulated as an MDP problem, where the optimization objective is to minimize the previously defined e_{pi}, e_{vi}, u_i and lastly jerk.

2.2.1. State space

The state space formula (6) can be discretized using the forward euler method giving the system equation below

$$x_{i,k+1} = A_{Di}x_{i,k} + B_{Di}u_{i,k} + C_{Di}u_{i-1,k}, \tag{8}$$

where $x_{i,k} = [e_{pi,k}, e_{vi,k}, a_{i,k}, a_{i-1,k}]$ is the observation state for the MDP problem that includes the position error $e_{pi,k}$, velocity error $e_{vi,k}$, acceleration $a_{i,k}$, and the acceleration of the predecessor vehicle $a_{i-1,k}$ at time step k. Moreover, A_{Di}, B_{Di}, and C_{Di} are given as

$$A_{Di} = \begin{bmatrix} 1 & T & -Th_i & 0 \\ 0 & 1 & -T & T \\ 0 & 0 & -\frac{T}{\tau_i}+1 & 0 \\ 0 & 0 & 0 & -\frac{T}{\tau_{i-1}}+1 \end{bmatrix} \qquad B_{Di} = \begin{bmatrix} 0 \\ 0 \\ \frac{T}{\tau_i} \\ 0 \end{bmatrix} \qquad C_{Di} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{T}{\tau_{i-1}} \end{bmatrix}. \tag{9}$$

2.2.2. Action space

Each vehicle within a single lane platoon follows the vehicle in front of it, and as such the only action the vehicle may take to maintain a desired headway is to accelerate, or decelerate. The action for the system is defined as the control input $u_{i,k}$ to the vehicle.

2.2.3. Reward function

The design of a reward in a DDPG system is critical to providing good performance within the system. In the considered driving scenario, it is logical to minimize position error, velocity error, the amount of time spent accelerating and the jerkiness of the driving motion. The proposed reward thus includes the normalized position error, $e_{pi,k}$, velocity error $e_{vi,k}$, control input $u_{i,k}$ and lastly the jerk. The vehicle reward $c_{i,k}$ is given

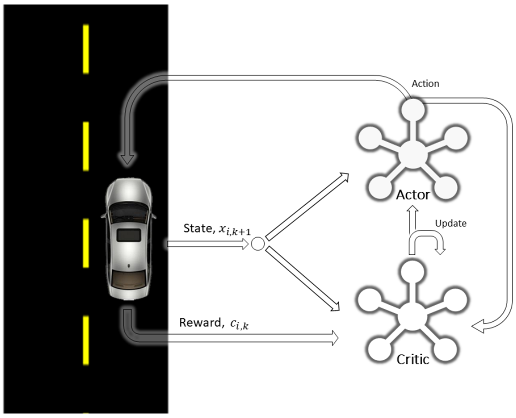


Figure 2. High level flow diagram of the DDPG model for a general vehicle V_i in a platoon.

below, where a b, c and d are system hyperparameters.

$$c_{i,k} = -\left(a\frac{|e_{pi,k}|}{\max(e_{pi,k})} + b\frac{|e_{vi,k}|}{\max(e_{vi,k})} + c\frac{|u_{i,k}|}{\max(u_{i,k})} + d\frac{|\dot{a}_{i,k}|}{2\max(a_{i,k})}\right) \tag{10}$$

2.3. FRL DDPG algorithm

In this section, the design for implementing the FRL DDPG algorithm on the AV platooning problem is presented.

2.3.1. DDPG model description

The DDPG algorithm is composed of an actor, μ and a critic, Q. The actor produces actions $u_t \in \mathbf{U}$ given some observation $x_t \in \mathbf{X}$ and the critic makes judgements on those actions while training using the Bellman equation[12,24]. The actor is updated by the policy gradient[24]. The critic network uses its weights θ^q to approximate the optimal action-value function $Q(x, u|\theta^q)$[24]. The actor network uses weights θ^μ to represent the agents' current policy $\mu(x|\theta^\mu)$ for the action-value function[24]. The actor $\mu(x) : \mathbf{X} \rightarrow \mathbf{U}$ maps the observation to the action. Experience replay is used to mitigate the issue of training samples not being independent and identically distributed due to their generation from sequential explorations[24]. Two additional models, the target actor μ' and critic Q' are used in DDPG to stabilize the training of the actor and critic networks by updating parameters slowly based on the target update coefficient τ. A sufficient value of τ is chosen such that stable training of μ and Q is observed. Figure 2 provides a high level simplified overview of how the DDPG algorithm interacts with a single vehicle in a platoon.

2.3.2. Inter and intra FRL

Modifications to the base DDPG algorithm are needed in order to implement Inter-FRL and Intra-FRL. In order to implement FedAvg the following modifications are required:

1. An FRL server: responsible for averaging the system parameters for use in a global update
2. Model weight aggregation: storing of each model's weights for use in aggregation
3. Model gradient aggregation: storing of each model's gradients for use in aggregation

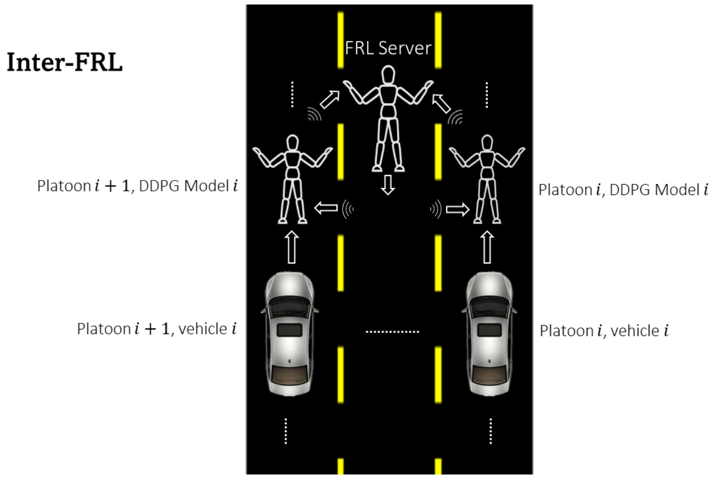

Inter-FRL
FRL Server
Platoon $i+1$, DDPG Model i
Platoon i, DDPG Model i
Platoon $i+1$, vehicle i
Platoon i, vehicle i

Figure 3. Inter-FRL.

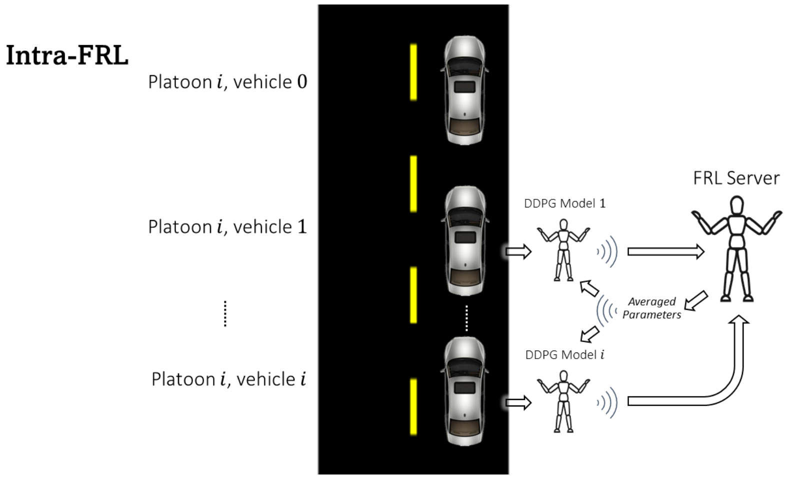


Figure 4. Intra-FRL.

In order to perform FRL, it has been proven that including an update delay between global FRL updates is beneficial for performance[5]. In addition, turning off FRL partway through training is important to allow each agent to refine their models independently of each other such that they can perform best with respect to their environments[5]. Lastly, it has also been shown that global updates and local updates should not be performed in the same episode[15].

Two methods of aggregation are implemented in the system design, Inter-FRL (see Figure 3), and Intra-FRL (see Figure 4). The proposed system is capable of aggregating both the model weights and gradients for each model so that either type of parameter may be averaged for use in global updates. The FRL server has the responsibility of averaging the parameters (model weights or gradients) across each agent in the system.

The pseudo-code for the Inter/Intra-FRL algorithm is presented in Algorithm 1. The system is designed to allow the training of any number of equal length platoons. At the lowest level, a DDPG agent exists for each vehicle in each platoon. As such, a DRL model must be initialized for each vehicle in the whole system. Each DDPG agent trains separately from the others before data is uploaded to the FRL server. Federated averaging is applied at a given time delay known as the FRL update delay, while being terminated at a given episode as defined by the cutoff ratio as seen in Table 3. Currently, Algorithm 1 is synchronous, and the FRL server is also synchronous.

3. EXPERIMENTAL RESULTS

In this section, the experimental setup for applying both Inter and Intra-FRL to the AV platooning environment is presented. The AV platooning environment and Inter/Intra FRL algorithms are implemented in Python 3.7 using Tensorflow 2.

3.1. Experimental setup

The parameters specific to the AV platoon environment are summarized in Table 1. The time step interval is $T = 0.1s$, and each training episode is composed of 600 time steps. Furthermore, the coefficients a, b, c and d given in the reward function (10) are a means to define how much each component of (10) contributes to the calculation of the reward. These coefficients may be tuned in order to determine a balance amongst each component, leading to better optimization during training. The coefficients were tuned using a grid search strategy and are listed as $a = 0.4$, $b = c = d = 0.2$.

Each DDPG agent consists of a replay buffer, and networks for the actor, target actor, critic and target critic. The actor network contains four layers: an input layer for the state, two hidden layers with 256 and 128 nodes, respectively, and an output layer. Both hidden layers use batch normalization and the relu activation function. The output layer uses the tanh() activation function. The output layer is scaled by the high bound for the control output, in this case 2.5 m/s^2. The critic network is structured with two separate input layers for state and action. These two layers are concatenated together, and fed into a single hidden layer before the output layer. The layer with the state input has 48 nodes, the relu activation function and batch normalization. The same is applied for the action layer, but instead with 256 nodes. The post concatenation layer uses 304 input nodes, followed by a hidden layer with 128 nodes, again with relu activation and batch normalization applied. The output of the critic uses a linear activation function. Ornstein-Uhlenbeck noise is applied to the model's predicted action, u_i. The structure of the models is presented in Figure 5a and 5b. All except the final layers of the actor and critic networks were initialized within the range $\left[-\frac{1}{\sqrt{f\,an\,in}}, \frac{1}{\sqrt{f\,an\,in}}\right]$, where-as the final layer is initialized using a random uniform distribution bounded by $[-3 \times 10^{-3},\ 3 \times 10^{-3}]$. Table 2 presents the hyperparameters specific to the DDPG algorithm.

The hyperparameters specific to Inter and Intra-FRL are presented in Table 3. During a training session with FRL, both local updates and FRL updates with aggregated parameters are applied to each DDPG agent in the system. FRL updates usually occur at a given frequency known as the FRL update delay, and furthermore, FRL updates may be terminated at a specific training episode as defined by the FRL cutoff ratio. The FRL update delay is defined as the time in seconds between FRL updates during a training episode. The FRL cutoff ratio is the ratio of the number of episodes where FRL updates are applied divided by the total number of episodes in a training session. Note that the aggregation method denotes whether the model gradients or weights are averaged during training using FRL.

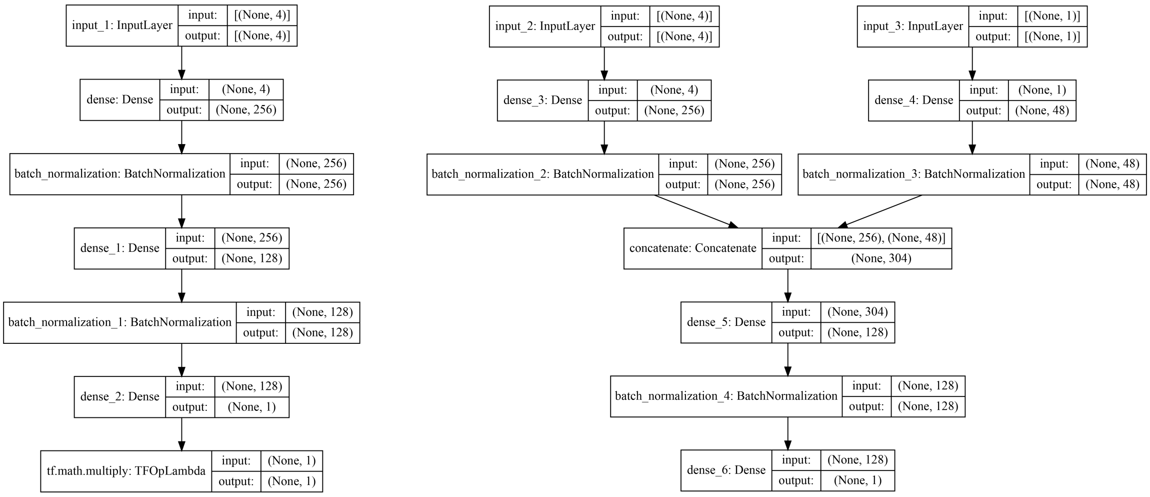
For the purposes of this study, an experiment is defined as a training session for a specific configuration of hyper-parameters, using the algorithm defined in Algorithm 1. During each experiment training session, model parameters were trained through the base DDPG algorithm or FRL in accordance with Algorithm 1. Once training has concluded, a simulation is performed using a custom built evaluator API. The evaluator

Algorithm 1: FRL applied to an AV platoon.

```
for each platoon $p \in platoons$ do
    for $v \in vehicles$ do
        initialize replay buffer $R_i$;
        initialize actor $\mu_i$, critic $Q_i$, target actor $\mu_i'$, target critic $Q_i'$;
    end
end
for $episode \in training\_episodes$ do
    for $p \in platoons$ do
        collect all vehicles states $x_{i,k}$ from $p$;
    end
    for $step \in steps\_per\_episode$ do
        for $p \in platoons$ do
            for $v \in vehicles$ do
                collect actions $u_{i,k}$ from actor;
            end
            advance the platoon $p$, with $u_{i,k}$;
            collect $(x_{i,k}, x_{i,k+1}, c_{i,k}, terminal)$ from $p$;
        end
        for $p \in platoons$ do
            for $v \in vehicles$ do
                add $(x_{i,k}, x_{i,k+1}, c_{i,k}, terminal)$ to replay buffer $R_i$;
                if FRL update is not required then
                    train $\mu_i$, $Q_i$, $\mu_i'$, $Q_i'$ locally;
                end
                append gradients of $\mu_i$ and $Q_i$ to all_gradients;
                append weights of $\mu_i$ and $Q_i$ to all_weights;
            end
        end
        if FRL update required then
            if gradient averaging enabled then
                avg_gradients ← global_update(all_gradients);
                train $\mu_i$, $Q_i$ using avg gradients;
            end
            if weight averaging enabled then
                avg_weights ← global_update(all_weights);
                update weights $\mu_i$, $Q_i$, $\mu_i'$, $Q_i'$ using avg weights;
            end
        end
    end
end
Function global_update(params) is
    upload params to FRL server;
    collect averaged params from FRL server;
    return averaged params;
end
```

Table 1. Parameters of the AV platoon environment

Parameter	Value
Time step T interval	0.1 s
Number of time steps per training episode	600
Time gap h_i	1 s
Driveline dynamics coefficient τ	0.1 s
Maximum absolute control input u_{max}	2.5 m/s^2
Reward coefficient a	0.4
Reward coefficient b	0.2
Reward coefficient c	0.2
Reward coefficient d	0.2



(a) The actor network for V_i. **(b)** The critic network for V_i.

Figure 5. Actor and critic networks for V_i.

Table 2. Hyperparameters for the DDPG Algorithm

Hyperparameter	Value
Actor learning rate	5e-05
Critic learning rate	0.0005
Batch size	64
Noise	Ornstein-Uhlenbeck Process with $\theta = 0.15$, $\sigma = 0.02$
Weights and Biases Initialization	random uniform distribution $[-3 \times 10^{-3}, 3 \times 10^{-3}]$ (final layer), $\left[-\frac{1}{\sqrt{fan\,in}}, -\frac{1}{\sqrt{fan\,in}}\right]$ (other layers)

Table 3. FRL Specific Initial Hyperparameters

FRL type	Aggregation method	Hyperparmeter	Value
Inter-FRL	Gradients	FRL update delay	0.1
Inter-FRL	Gradients	FRL cutoff ratio	0.8
Inter-FRL	Weights	FRL update delay	30
Inter-FRL	Weights	FRL cutoff ratio	1.0
Intra-FRL	Gradients	FRL update delay	0.4
Intra-FRL	Gradients	FRL cutoff ratio	0.5
Intra-FRL	Weights	FRL update delay	0.1
Intra-FRL	Weights	FRL cutoff ratio	1.0

performs simulations for a single 60 second episode using the trained models, calculating the cumulative reward of the model(s) in the experiment. The entire project is designed and implemented using Python3, and

Table 4. Performance after training across 4 random seeds. Each simulation result contains 600 time steps

Training method	Seed 1	Seed 2	Seed 3	Seed 4	Average system reward	Standard deviation
No-FRL	-3.73	-2.89	-4.69	-3.38	-3.67	0.66
Inter-FRLGA	-2.79	-2.81	-3.05	-2.76	-2.85	0.11
Inter-FRLWA	-2.64	-2.88	-2.92	-2.93	-2.84	0.12

Tensorflow. As previously stated, each vehicle in the platoon is modelled using the CACC CTHP model described in Section 3. For the purposes of this study, multiple sets of DRL experiments were conducted, using 4 random seeds (1-4) for training and a single random seed (6) across all evaluations.

3.2. Inter-FRL

In order to evaluate the effectiveness of Inter-FRL relative to the base case where a DRL model is trained using DDPG without FRL, 4 experiments are conducted without Inter-FRL (no-FRL), and 8 with. For each of the 12 conducted experiments, 2 platoons with 2 vehicles each were trained using one of the four random seeds. Once training across the four seeds has completed, the cumulative reward for a single evaluation episode is evaluated. For the experiments using Inter-FRL, two aggregation methods are examined. First, the gradients of each model are averaged during training, and second, the model weights are averaged. The multi platoon system trains and shares the aggregated parameters (gradients or weights) amongst vehicles with the same index across platoons. The federated server is responsible for performing the averaging, and each vehicle performs a training episode with the averaged parameters in addition to their local training episodes in accordance with the FRL update delay and FRL cutoff ratio (see Table 3). Note that here-after Inter-FRL with gradient aggregation is denoted Inter-FRLGA, and Inter-FRL with weight aggregation is denoted Inter-FRLWA.

3.2.1. Performance across 4 random seeds

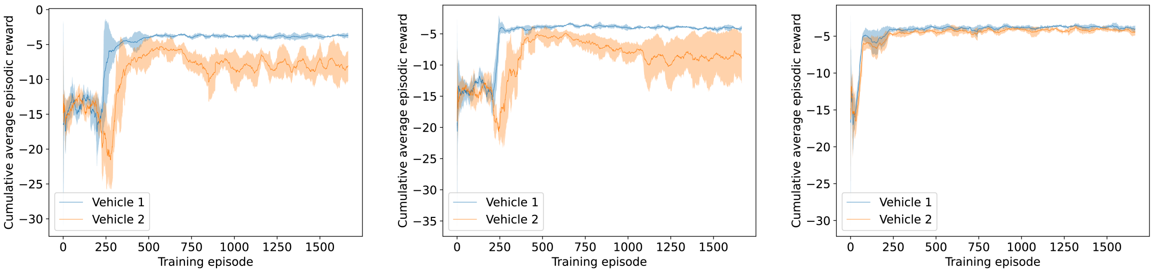
The performance for each of the systems is calculated by averaging the cumulative reward of each vehicle in the 2 vehicle 2 platoon system, as summarized in Table 4. For each of the 3 cases (base case, Inter-FRLGA and Inter-FRLWA), training sessions were run using 4 random seeds. In order to determine the highest performing system overall, an average and standard deviation is obtained from the result of training using the 4 random seeds. From Table 4, it is observed that both Inter-FRL scenarios using gradient and weight aggregation provide large performance increases to that of the base case.

3.2.2. Convergence properties

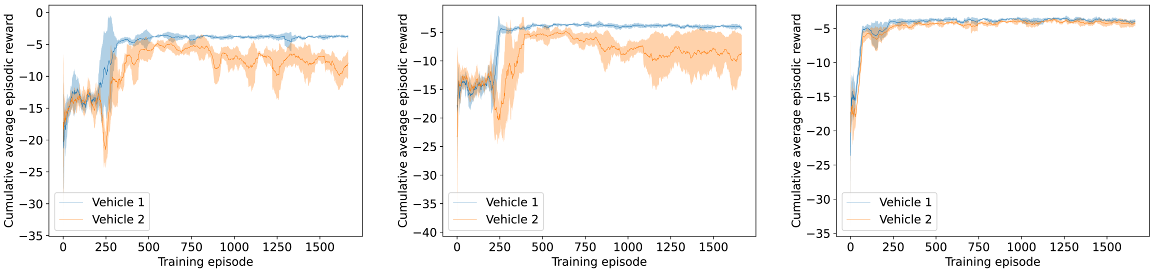
The cumulative reward is calculated over each training episode, and a moving average is computed over 40 episodes to generate Figure 6a-6f. It can be seen that the cumulative reward for Inter-FRLWA not only converges more rapidly than both no-FRL and Inter-FRLGA, but Inter-FRLWA also appears to have a more stable training session as indicated by the lower magnitude of the shaded area (the standard deviation across the four random seeds).

3.2.2. Test results for one episode

In Figure 7a and 7b, a simulation is performed over a single training episode plotting the jerk, along with the control input $u_{i,k}$, acceleration $a_{i,k}$, velocity error $e_{vi,k}$, and position error $e_{pi,k}$ for each platoon. There are 2



(a) No-FRL: Platoon 1 **(b)** Inter-FRLGA: Platoon 1 **(c)** Inter-FRLWA: Platoon 1



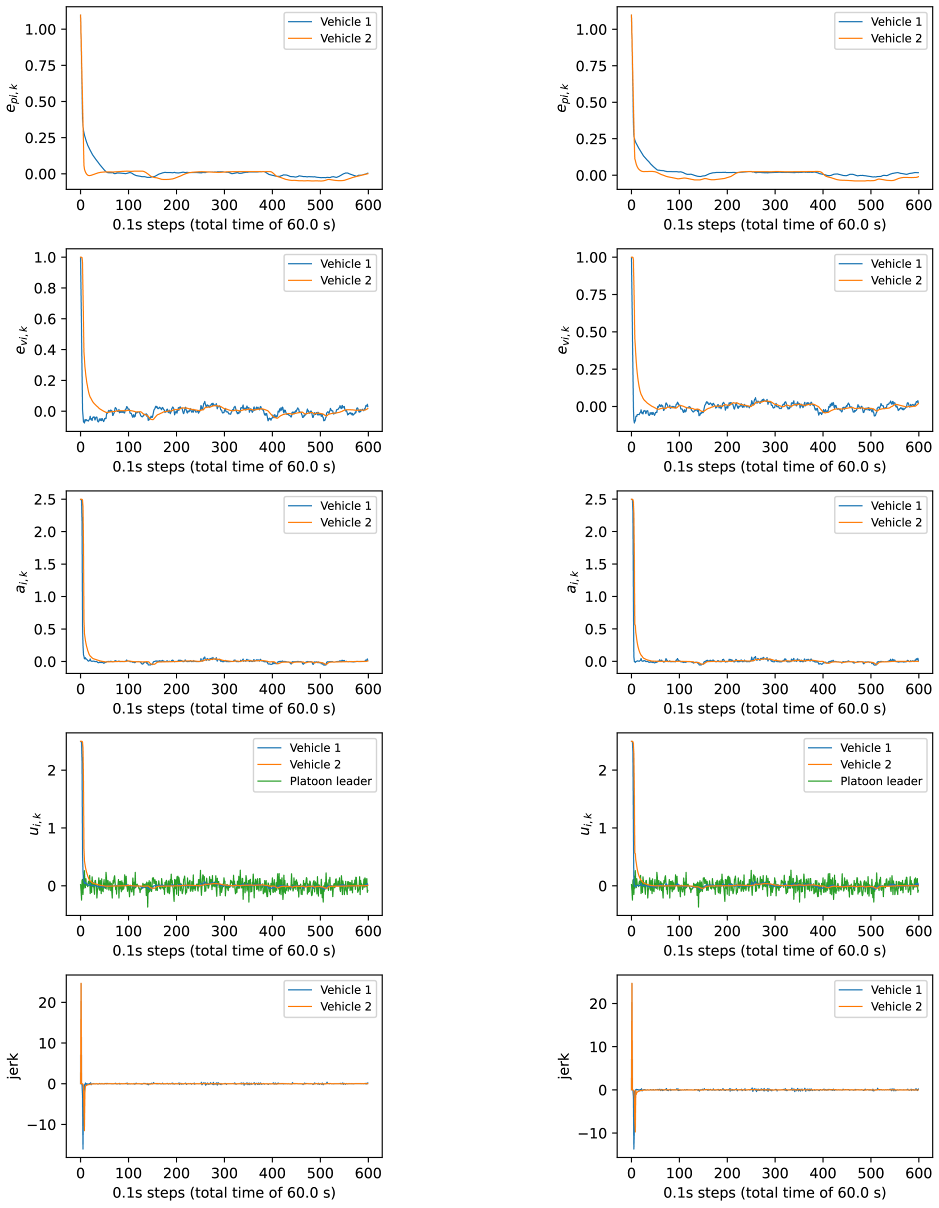
(d) No-FRL: Platoon 2 **(e)** Inter-FRLGA: Platoon 2 **(f)** Inter-FRLWA: Platoon 2

Figure 6. Average performance across 4 random seeds for a 2 platoon 2 vehicle scenario trained without FRL (Figure 6a, 6d), with Inter-FRLGA (Figure 6b, 6e), and with Inter-FRLWA (Figure 6c, 6f). The shaded areas represent the standard deviation across the 4 seeds.

platoons in the Inter-FRL scenario, and a simulation is provided for each platoon. The simulation environment is subject to initial conditions of ($e_{pi} = 1.0\ m$, $e_{vi} = 1.0\ m/s$, $a_i = 0.03\ m/s^2$). It can be seen that each DDPG agent for both vehicles within both platoons quickly responds to the platoon leader's control input $u_{i,k}$ to bring the position error, velocity error and acceleration error to 0. In addition, each DDPG agent closely approximates the Gaussian random input of the platoon leader, eliminating noise in the response to maintain smooth tracking across the episode. Finally, each DDPG agent in the platoon also minimizes the jerk effectively. These results are indicative of both a good design of the reward function (10), and also a suitable selection of parameters a, b, c and d in (10).

3.3. Intra-FRL

In order to evaluate the effectiveness of Intra-FRL relative to the base AV platooning scenario, 4 experiments are conducted without Intra-FRL (no-FRL), and 8 with. For each of the conducted experiments, 1 platoon with 2 vehicles is trained using 4 random seeds. A single platoon is required for studying Intra-FRL as parameters are shared amongst vehicles within the platoon (no sharing is performed from vehicle's in one platoon to another). Once training across the four seeds is completed, the cumulative reward for a single evaluation episode is evaluated. Similar to the experiments using Inter-FRL, two aggregation methods are examined. First, the gradients of each model are averaged during training, and second, the model weights are averaged. The platoon trains and shares the aggregated parameters (gradients or weights) from vehicle to vehicle such that data is averaged and updated amongst vehicles within the same platoon. The federated server is responsible for performing the averaging, and each vehicle performs a training episode with the averaged parameters in addition to their local training episodes in accordance with the FRL update delay and FRL cutoff ratio (see Table 3). Note that here-after Intra-FRL with gradient aggregation is denoted Intra-FRLGA, and Intra-FRL with weight aggregation is denoted Intra-FRLWA.



(a) Platoon 1 **(b)** Platoon 2

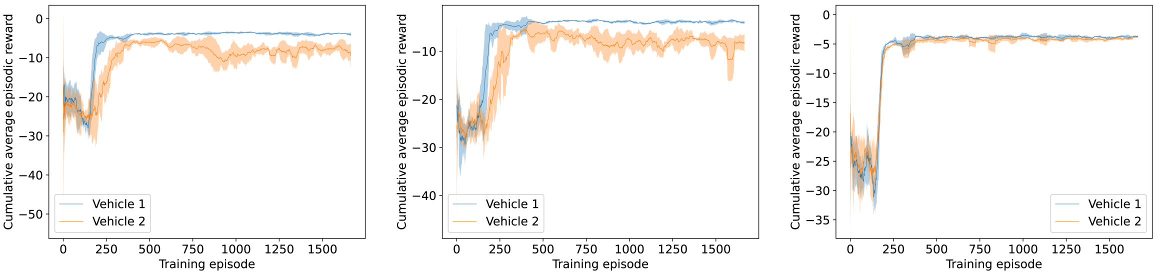
Figure 7. Results for a specific 60s test episode using the 2 vehicle 2 platoon environment trained using Inter-FRL with weight aggregation.

3.3.1. Performance across 4 random seeds

The performance for the platoon is calculated by averaging the cumulative reward generated by the simulation for each of the 4 random seeds and is summarized in Table 5. The results in Table 5 summarize the performance for no-FRL, Intra-FRLGA, and lastly Intra-FRLWA. It is observed that Intra-FRLWA performs most favourably, followed by no-FRL and lastly Intra-FRLGA.

Table 5. Performance after training across 4 random seeds. Each simulation result contains 600 time steps

Training method	Seed 1	Seed 2	Seed 3	Seed 4	Average system reward	Standard deviation
No-FRL	-3.84	-3.40	-3.29	-3.21	-3.44	0.24
Intra-FRLGA	-2.85	-8.05	-4.23	-2.99	-4.53	2.10
Intra-FRLWA	-2.56	-2.60	-2.68	-2.75	-2.65	0.07



(a) No-FRL **(b)** Intra-FRLGA **(c)** Intra-FRLWA

Figure 8. Average performance across 4 random seeds for 1 platoon 2 vehicle scenario trained without FRL (Figure 6a), Intra-FRLGA (Figure 6b), and with Intra-FRLWA (Figure 6c). The shaded areas represent the standard deviation across the 4 seeds.

3.3.2. Convergence properties

The cumulative reward is calculated over each training episode, and a moving average is computed over 40 episodes to generate Figure 8. Similar to the Inter-FRL experiments, Intra-FRLWA shows the most favourable training results. In addition, the rate of convergence increases with Intra-FRLWA over no-FRL and Intra-FRLGA. Lastly, the stability during training is also shown to be improved as the standard deviation across the four random seeds is much smaller than the other two cases (as evident in the shaded regions of Figure 8).

3.3.3. Test results for one episode

A single simulation is performed on an episode plotting the jerk, along with the control input $u_{i,k}$, acceleration $a_{i,k}$, velocity error $e_{vi,k}$, and position error $e_{pi,k}$. Figure 9 shows the precise control of Intra-FRLWA on the environment. The environment is initialized to the same conditions as that of the Inter-FRLWA scenario ($e_{pi} = 1.0m$, $e_{vi} = 1.0m/s$, $a_i = 0.03m/s^2$), and each DDPG agent in the platoon quickly and precisely tracks the Gaussian random input $u_{i,k}$ from the leader while minimizing position error, velocity error, acceleration , and jerk. Much like the Inter-FRLWA scenario, it is observed that a strong optimization of the reward function (Equation 10) has occurred. This is an indication of a good design of the reward function in addition to a good balance of parameters a, b, c and d in the reward function.

3.4. Comparison between inter and intra-FRL

The results for both Inter-FRL and Intra-FRL are summarized in Table 6 below.

It is clear that using weight aggregation in both Inter-FRL and Intra-FRL is favourable to gradient aggregation. In addition, Intra-FRLWA provides the overall best result. Intra-FRL likely converges to the best model due to conditions each agent experiences during training. For Inter-FRL, the environment is independent and identically distributed. For Intra-FRL, each follower's training depends on the policy of the preceding vehicle. For the 2 vehicle scenario studied, vehicle 1 will converge prior to vehicle 2 as vehicle 1 learns based on the

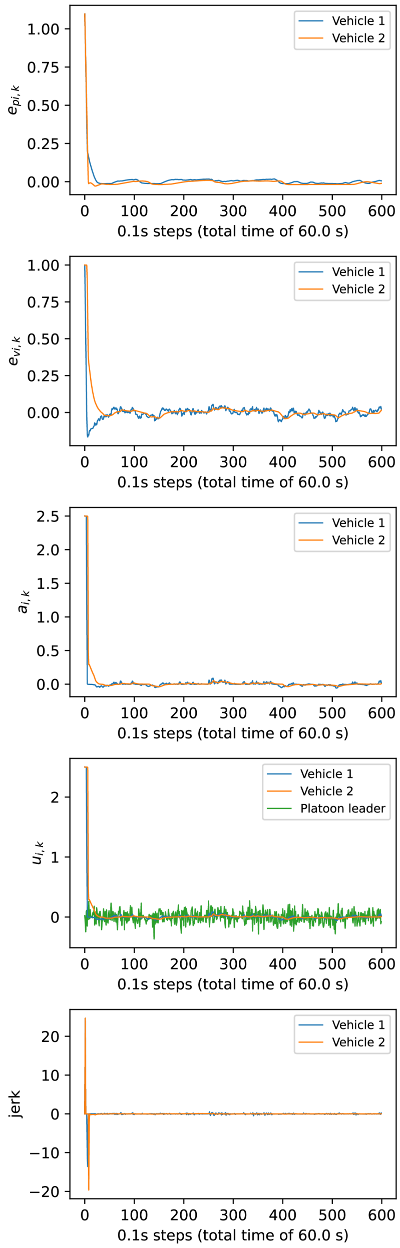


Figure 9. Results for a specific 60s test episode using the 2 vehicle 1 platoon environment trained using Intra-FRLWA.

Table 6. Performance after training across 4 random seeds for both Inter and Intra FRL. Each simulation result contains 600 time steps.

Training Method	Seed 1	Seed 2	Seed 3	Seed 4	Average system reward	Standard deviation
Inter-FRLGA	-2.79	-2.81	-3.05	-2.76	-2.85	0.11
Inter-FRLWA	-2.64	-2.88	-2.92	-2.93	-2.84	0.12
Intra-FRLGA	-2.85	-8.05	-4.23	-2.99	-4.53	2.10
Intra-FRLWA	-2.56	-2.60	-2.68	-2.75	-2.65	0.07

stochastic random input generated by the platoon leader. As vehicle 1 is training, vehicle 2 trains based off the policy of vehicle 1. As previously stated, Inter-FRL shares parameters amongst vehicles in the same index across platoons, where-as Intra-FRL provides the advantage of sharing parameters from preceding vehicles to following vehicles. Our implementation of Intra-FRL includes a directional parameter averaging. For exam-

Table 7. Performance after training across 4 random seeds with varying platoon lengths. Each simulation result contains 600 time steps.

Training Method	No. Vehicles	Seed 1	Seed 2	Seed 3	Seed 4	Avg. System Reward	Std. Dev.
No-FRL	3	-3.64	-3.28	-3.76	-3.52	-3.55	0.20
No-FRL	4	-123.58	-4.59	-7.39	-4.51	-35.02	59.06
No-FRL	5	-4.90	-5.94	-6.76	-6.11	-5.93	0.77
Intra-FRLWA	3	-3.44	-3.16	-3.43	-4.14	-3.54	0.42
Intra-FRLWA	4	-3.67	-3.56	-4.10	-3.60	-3.73	0.25
Intra-FRLWA	5	-3.92	-4.11	-4.33	-3.97	-4.08	0.18

ple, vehicle 1 does not train with averaged parameters from the followers, but vehicle 2 has the advantage of including vehicle 1's model in its averaging. This directional averaging provides an advantage to vehicle 2, as evidenced by the increased performance in Table 6.

3.5. Intra-FRL with variant number of vehicles

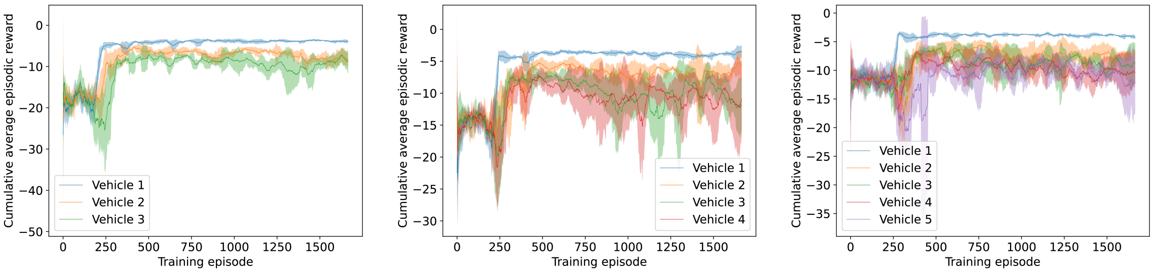
An additional factor to consider when evaluating FRL in relation to the no-FRL base scenario is how FRL performs with increasing agents relative to no-FRL. In this section, 12 experiments are conducted with no-FRL, and 12 with Intra-FRLWA. Each set of 12 experiments for no-FRL and Intra-FRLWA are broken up by number of vehicles and random seed. The random seed is selected to be a value between 1 and 4, inclusive. In addition, the platoons under study contain either 3, 4, or 5 vehicles. Once training has been completed for all experiments, the cumulative reward for each experiment is evaluated using a single simulation episode in which the seed is kept constant. Intra-FRLWA is used as the FRL training strategy since Intra-FRLWA was identified to be the highest performing FRL strategy in the previous section.

3.5.1. Performance with varying number of vehicles

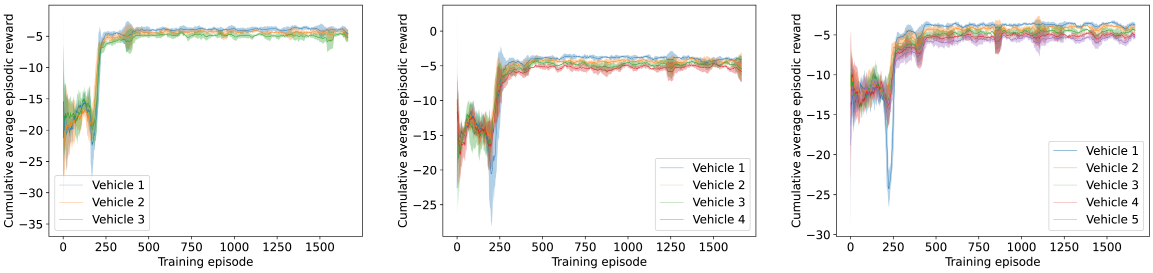
The performance for each experiment is calculated by taking the average cumulative episodic reward across each vehicle in the platoon at the end of the simulation episode. Table 7 presents the results for no-FRL and Intra-FRLWA for platoons with 3, 4, and 5 follower vehicles. Table 7 shows that Intra-FRLWA provides favourable performance in all platoon lengths. A notable example of Intra-FRLWA's success is highlighted when considering the poor performance of the 4 vehicle platoon trained with no-FRL using seed 1. The Intra-FRLWA training strategy was able to overcome the performance challenges, correcting the poor performance entirely.

3.5.2. Convergence properties

The cumulative reward is calculated over each training episode, and a moving average is computed over 40 episodes to generate Figure 10. Intra-FRLWA shows favourable training performance to that of the no-FRL scenario for all platoon lengths. In addition, the rate of convergence is increased using Intra-FRLWA versus no-FRL. Furthermore, the shaded areas corresponding to standard deviation across the seeds are reduced significantly, indicating better stability across the seeds for Intra-FRLWA than no-FRL. Last, the overall stability is improved as shown by the large noise reduction during training in Figure 10d, 10e, 10f when compared with no-FRL's Figure 10a, 10b, 10c.



(a) No-FRL: 3 Vehicles **(b)** No-FRL: 4 Vehicles **(c)** No-FRL: 5 Vehicles



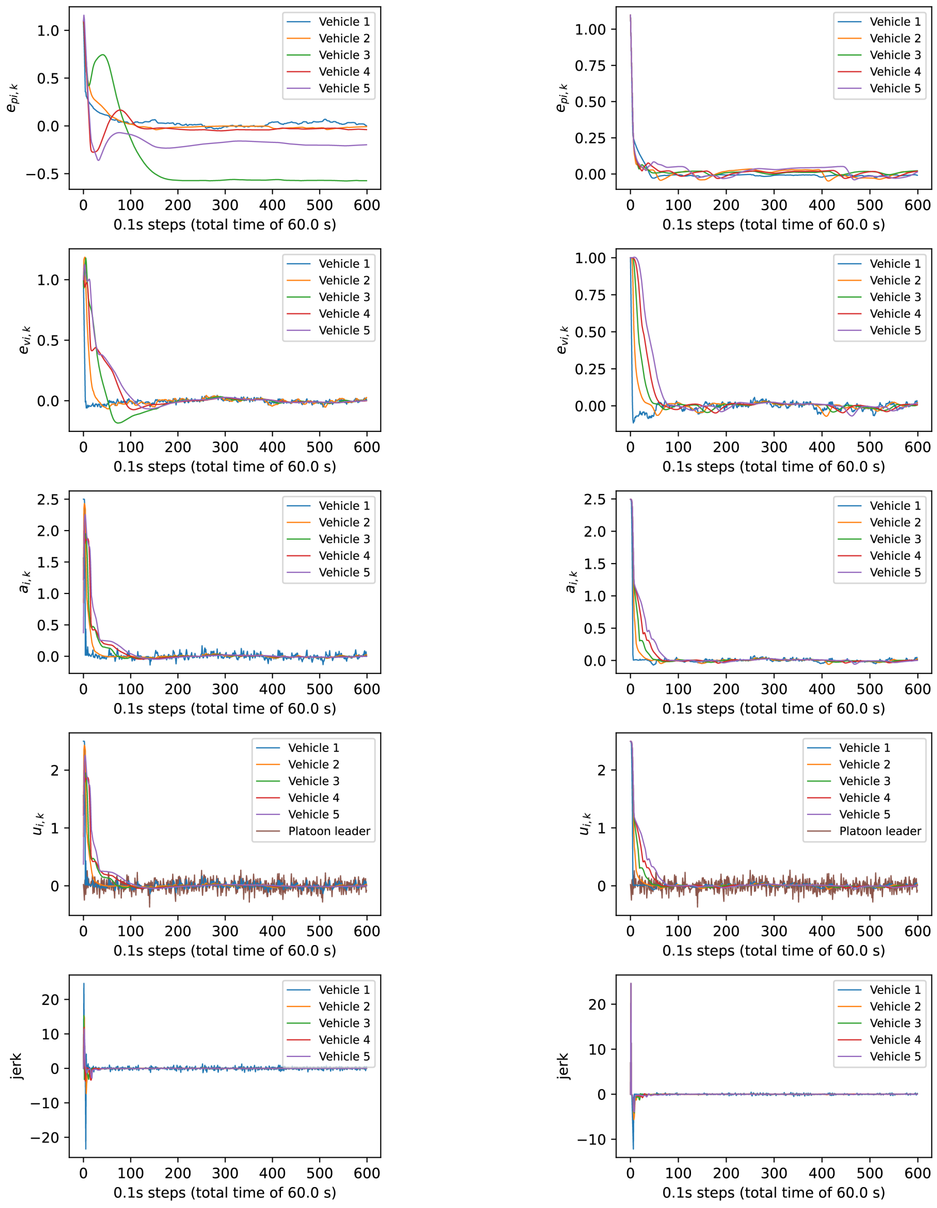
(d) Intra-FRLWA: 3 Vehicles **(e)** Intra-FRLWA: 4 Vehicles **(f)** Intra-FRLWA: 5 Vehicles

Figure 10. Average performance across 4 random seeds for 3 platoons with 3, 4 and 5 followers trained without FRL (Figures 10a, 10b, 10c), and with Intra-FRLWA (Figure 10d, 10e, 10f). The shaded areas represent the standard deviation across the four seeds.

3.5.3. *Test results for one episode*

As with all previous sections, a single simulation is performed on a 60 second episode plotting the jerk along with the control input $u_{i,k}$, acceleration $a_{i,k}$, velocity error $e_{vi,k}$, and position error $e_{pi,k}$. Figure 11 showcases the ability of Intra-FRLWA to control a 5 platoon environment precisely when compared to a platoon trained without Intra-FRLWA. The environment for Intra-FRLWA is initialized with the same values as no-FRL, just like all previous experiments: ($e_{pi} = 1.0m$, $e_{vi} = 1.0m/s$, $a_i = 0.03m/s^2$). Each DDPG agent trained with Intra-FRLWA quickly and precisely tracks the Gaussian random control input $u_{i,k}$ from the leader minimizing $e_{pi,k}$, $e_{vi,k}$, $a_{i,k}$ and jerk. In particular, the response for $e_{pi,k}$ and $e_{vi,k}$ in the platoon trained using Intra-FRLWA (Figure 11b) appears to respond to the platoon leader's input quicker and in a much smoother manner than that of the no-FRL scenario (Figure 11a).

The large difference in performance for no-FRL versus Intra-FRL can be explained by understanding how Intra-FRLWA works. With no-FRL, each agent trains independently, and the inputs to the following vehicles are directly outputted from the predecessors. Thus, the followers farther back in the platoon take longer to train as their predecessors' outputs can be highly variable while training. As the policies of the predecessors converge, the policy of each follower can then begin to converge. This sequential convergence from predecessor to follower can be seen in Figure 10, where the convergence during training is slower for vehicles 4 and 5 than it is for 3, 2 and 1. Intra-FRLWA helps to resolve this challenge by allowing vehicles to average their model weights, thus distributing an aggregation of more mature predecessor parameters amongst the platoon.



(a) No-FRL **(b)** Intra-FRLWA

Figure 11. Results for a specific 60s test episode using the 5 vehicle 1 platoon environment trained using no-FRL (Figure 11a), and with Intra-FRLWA (Figure 11b).

4. CONCLUSION

In this paper, we have formulated an AV platooning problem and successfully applied FRL in a variety of methods to AV platooning. In addition, we proposed new approaches for applying FRL to AV platoons: Inter-FRL and Intra-FRL. By comparing FRL performance with both gradient and weight averaging in the AV platooning scenario, it has been shown that weight averaging was the optimal aggregation method regardless of using Inter-FRL or Intra-FRL. Furthermore, it was found that the Intra-FRLWA strategy was most advantageous for applying FRL to AV platooning. Finally, it was proven that applying Intra-FRLWA to AV platoons up to 5 vehicles in length provided large performance advantages during and after training when compared to AV platoons that were controlled by DDPG agents trained without FRL. These results are backed by simulations performed using models trained across four random seeds, and an additional simulation set with variable platoon sizes. The focus of this paper was on decentralized platoon control, where each follower in the platoon trains locally with respect to their individual reward.

In the future, improvements to the system could be made by implementing weighted averaging in the FRL aggregation method. Moreover, in AV platooning, communication delays can be considered in the model to give a more concrete real life example.

DECLARATIONS

Authors' contributions

Made substantial contributions to the research, idea generation, testing, and software development. Solely programmed the Python AVDDPG application for conducting the DRL experiments, simulating and aggregating experiment results. Wrote and edited the original draft: Boin C
Performed oversight and leadership responsibility for the research activity planning and execution, as well as developed ideas and evolution of overarching research aims. Assisted editing the original draft: Lei L
Performed critical review, commentary and revision, as well as provided administrative, technical, and material support: Yang S

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 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.

Copyright

REFERENCES

1. McMahan HB, Moore E, Ramage D, Arcas BA. Federated learning of deep networks using model averaging. *ArXiv*, vol. abs/1602.05629, 2016.
2. Konecný J, McMahan HB, Ramage D. Federated optimization: Distributed optimization beyond the datacenter. *ArXiv*, vol. abs/1511.03575, 2015.
3. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, vol. 54, 2017.
4. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and Applications. *arXiv*, vol. 10, no. 2, pp. 1–19, 2019.
5. Lim HK, Kim JB, Heo JS, Han YH. Federated reinforcement learning for training control policies on multiple iot devices. *Sensors (Basel)* 2020;20:1359. Available: https://doi.org/10.3390/s20051359
6. Ye Y, Zhang X, Sun J. Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. *Transportation Research Part C: Emerging Technologies* 2019;107:155-70. Available: https://doi.org/10.1016/j.trc.2019.08.011
7. Zhu M, Wang X, Wang Y. Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies* 2018;97:348-68. Available: https://doi.org/10.1016/j.trc.2018.10.024
8. Song X, Chen L, Wang K, He D. Robust time-delay feedback control of vehicular cacc systems with uncertain dynamics. *Sensors (Basel)* 2020;20:1775. Available: https://doi.org/10.3390/s20061775
9. Chu T, Kalabic U. Model-based deep reinforcement learning for CACC in mixed-autonomy vehicle platoon. Proceedings of the IEEE Conference on Decision and Control. vol. 2019-December, pp. 4079-84. [Online]. Available: https://doi.org/10.1109/CDC40024.2019.9030110
10. Nadiger C, Kumar A, Abdelhak S. Federated reinforcement learning for fast personalization. 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 123-127. Available: https://doi.org/10.1109/AIKE.2019.00031
11. Qi J, Zhou Q, Lei L, Zheng K. Federated reinforcement learning: Techniques, applications, and open challenges. *Intell Robot*2021;1:18-57. https://doi.org/10.20517/ir.2021.02
12. Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press, 2018.
13. Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 2019;13:1-207. Available: https://doi.org/10.2200/S00960ED2V01Y201910AIM043
14. Liu B, Wang L, Liu M. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *arXiv*, vol. 4, no. 4, pp. 4555–4562, 2019.
15. Liang X, Liu Y, Chen T, Liu M, Yang Q. Federated transfer reinforcement learning for autonomous driving. *arXiv*, 2019.
16. Zhang X, Peng M, Yan S, Sun Y. Deep-reinforcement-learning-based mode selection and resource allocation for cellular v2x communications. *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6380–6391, 2020.
17. Lim H, Kim J, Ullah I, Heo J, Han Y. Federated reinforcement learning acceleration method for precise control of multiple devices. *IEEE Access*, vol. 9, pp. 76 296–76 306, 2021.
18. Wang X, Li R, Wang R, Li X, Taleb T, Leung VCM. Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 154–169, 2021.
19. Huang H, Zeng C, Zhao Y, Min G, Zhu Y, Miao W, Hu J. Scalable orchestration of service function chains in nfv-enabled networks: A federated reinforcement learning approach. *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2558–2571, 2021.
20. Makantasis K, Kontorinaki M, Nikolos I. Deep reinforcement-learning-based driving policy for autonomous road vehicles. *IET Intelligent Transport Systems* 2020;14:13-24. [Online]. Available: https://doi.org/10.1049/iet-its.2019.0249
21. Sallab AE, Abdou M, Perot E, Yogamani S. Deep reinforcement learning framework for autonomous driving. *IS and T International Symposium on Electronic Imaging Science and Technology* 2017;29:70-6. [Online]. Available: https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023
22. Lin Y, McPhee J, Azad NL. Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019. pp. 1504–1510.
23. Peake A, McCalmon J, Raiford B, Liu T, Alqahtani S. Multi-agent reinforcement learning for cooperative adaptive cruise control. 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 2020, pp. 15-22 [Online]. Available: https://doi.org/10.1109/ICTAI50040.2020.00013
24. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 2016.

25. Lin Y, Mcphee J, Azad NL. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. *IEEE Trans Intell Veh* 2021;6:221-31. Available: http://dx.doi.org/10.1109/TIV.2020.3012947
26. Yan R, Jiang R, Jia B, Yang D, Huang J. Hybrid car-following strategy based on deep deterministic policy gradient and cooperative adaptive cruise control, 2021.
27. Zhu M, Wang Y, Hu J, Wang X, Ke R. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *CoRR*, vol. abs/1902.00089, 2019. [Online]. Available: http://arxiv.org/abs/1902.00089
28. Lei L, Liu T, Zheng K, Hanzo L. Deep reinforcement learning aided platoon control relying on V2X information. *IEEE Transactions on Vehicular Technolog* 2022. Available: http://dx.doi.org/10.1109/TVT.2022.3161585
29. Lei L, Tan Y, Zheng K, Liu S, Zhang K, Shen X. Deep reinforcement learning for autonomous internet of things: Model, applications and challenges. *IEEE Communications Surveys Tutorials* 2020;22:1722-60.

Lv *et al*. *Intell Robot* 2022;2(2):168-79
DOI: 10.20517/ir.2022.09

Intelligence & Robotics

Research Article

Open Access

Opponent modeling with trajectory representation clustering

Yongliang Lv, Yan Zheng, Jianye Hao

College of Intelligence and Computing, Tianjin University, Tianjin 300000, China.

Correspondence to: Yongliang Lv, College of Intelligence and Computing, Tianjin University, No. 135 Yaguan Road, Jinnan District, Tianjin 300000, China. E-mail: Lvyongliang@tju.edu.cn

How to cite this article: Lv Y, Zheng Y, Hao J. Opponent modeling with trajectory representation clustering. *Intell Robot* 2022;2(2):168-79. http://dx.doi.org/10.20517/ir.2022.09

Received: 9 Mar 2022 **First Decision:** 25 Apr 2022 **Revised:** 16 May 2022 **Accepted:** 1 Jun 2022 **Published:** 16 Jun 2022

Academic Editor: Simon X. Yang **Copy Editor:** Fanglin Lan **Production Editor:** Fanglin Lan

Abstract

For a non-stationary opponent in a multi-agent environment, traditional methods model the opponent through its complex information to learn one or more optimal response policies. However, the response policy learned earlier is prone to catastrophic forgetting due to data imbalance in the online-updated replay buffer for non-stationary changes of opponent policies. This paper focuses on how to learn new response policies without forgetting old policies that have been learned when the opponent policy is constantly changing. We extract the representation of opponent policies and make explicit clustering distinctions through the contrastive learning autoencoder. With the idea of balancing the replay buffer, we maintain continuous learning of the trajectory data of various opponent policies that have appeared to avoid policy forgetting. Finally, we demonstrate the effectiveness of the method under a classical opponent modeling environment (soccer) and show the clustering effect of different opponent policies.

Keywords: Non-stationary, opponent modeling, contrastive learning, trajectory representation, data balance

1. INTRODUCTION

In the field of multi-agent reinforcement learning (MARL)[1–3], the non-stationary problem[4,5] caused by policy changes of other agents has always been challenging. Since the policy and behavior of other agents are generally unknown when the policies of other agents change, the environment is no longer considermed to be a stationary arkov decision process (MDP), and it cannot be solved by simply using a single-agent reinforcement learning algorithm[6–8]. A common class of ideas is to introduce additional information to aid training by modeling other agents i.e. opponent modeling[4,9].

Opponent modeling is a common idea in the MARL domain, which has many works of different points of view, such as explicitly representing the opponent's policies through neural networks to train some optimal response policies[10–12] or implicitly learning the opponent policy's representation to assist training[13–16]. Since the goal of the agent under our control is to maximize its local reward, other agents are viewed collectively as an opponent, although "opponent" does not always imply a fully competitive environment. However, existing opponent modeling methods, whether explicitly or implicitly, set the opponent to use a fixed policy or switch between fixed policies, which is not suitable for most real-world situations. Therefore, we further set the opponent policy in the form of a probability distribution, so as to learn a general policy that can deal with all kinds of opponents, which requires additional consideration of policy forgetting.

Specifically, when the opponent policy changes, the data in the replay buffer[17] are constantly replaced by the interactive trajectory with the new opponent policy so that the agent's response policy converges to deal with the new opponent policy. However, at the same time, the agent may forget the response policy it has learned before because of the loss of previous interaction data; therefore, it still needs to re-learn when some opponent policies appear again, which greatly reduces the response efficiency.

We believe that the main reason for this type of policy forgetting problem is that there are not enough trajectories of interactions with various opponent policies saved in the replay buffer. Thus, this paper uses the idea of data balancing[18,19] to ensure the diversity of trajectories interacting with various opponent policies in the replay buffer as much as possible. Data balancing is widely used in continuous learning[20] to solve catastrophic forgetting problems. In contrast, in most continuous learning settings, task IDs are given to distinguish between different tasks, but we do not know the types of opponent policies. Thus, to distinguish various trajectories, we self-supervise extracted policy representations from interactive trajectories by contrastive learning[21–24] and clustering at the representational level. Our proposed method, trajectory representation clustering (TRC), can be combined with any existing reinforcement learning (RL) algorithm, to avoid policy forgetting in non-stationary multi-agent environments.

The contributions of this paper can be summarized as follows: (1) Interaction trajectories are self-supervised encoded through a contrastive learning algorithm so that different opponent policies can be more accurately represented and distinguished in the representation space. No additional information is required except the opponent observation; (2) From the perspective of balancing data types, we artificially retain the types of data that account for a small proportion in the replay buffer to avoid catastrophic policy forgetting.

The rest of this paper is organized as follows. The related work on opponent modeling and contrastive learning is discussed in Section 2. Section 3 details the used network architecture, loss function, and algorithm flow. Then, some experiments based on the classic environment of soccer are presented to verify the performance of our method in Section 4. Finally, the conclusions and future work are introduced in Section 5.

2. RELATED WORK

2.1. Opponent modeling

Opponent modeling stems from a naive motivation that infers the opponent's policy and behavior through the information about the opponent to obtain a higher reward for itself. Early opponent modeling work[25,26] mainly focused on simple game scenarios where the opponent policy is fixed. With the development of deep reinforcement learning, scholars have begun to apply the idea of opponent modeling in more complex environments and settings. The following introduces the opponent modeling work in recent years in terms of explicit modeling and implicit modeling.

2.1.1. Implicit opponent modeling

Implicit opponent modeling generally refers to extracting representations from opponent information to aid training. He *et al.* first used the opponent's observation and agent's observation as merged input in a deep network to train the agent end-to-end. They also pointed out that information such as the opponent's policy type can be used to assist the training of RL[13]. Subsequently, Hong *et al.* additionally used the information of opponent action, fitted the opponent policy through the neural network, and then multiplied the output of the hidden layer of the opponent's policy network with the output of the hidden layer of the Q network to calculate the Q value[14]. Considering that the opponent may also have learning behaviors, Foerster *et al.* maximized the agent's reward by estimating the parameters of the opponent policy network based on the idea of recurrent reasoning[16]. Raileanu *et al.* considered the parameters of the opponent policy network from another perspective and used the agent policy to make decisions based on the opponent observation, so as to infer the opponent's goal and achieve better performance[15]. Due to the different assumptions about the opponent, the effects of different algorithms are also difficult to compare.

2.1.2. Explicit opponent modeling

Explicit opponent modeling generally refers to explicitly modeling opponent policies, dividing opponent types, and detecting and responding online during the interaction process. Rosman *et al.* first proposed Bayes policy reuse (BPR) to be used in multi-task learning, maintaining a belief for each task through Bayesian formula, judging the task type, and choosing the optimal response policy for unknown tasks[27]. Since then, Hernandez-Leal *et al.* extended the environment to a multi-agent system, used MDP to model opponents, and added a detection mechanism for unknown opponent policies[10] . In the face of more complex environments, Zheng *et al.* used neural networks to model opponents and the rectified belief model (RBM) to make opponent detection more accurate and rapid, as well as policy distillation technology to reduce the scale of the network[11]. On this basis, Yang *et al.* introduced the theory of mind[28] to defeat opponents with higher-level decision-making methods for opponents who also use opponent modeling method[12].

2.2. Contrastive learning

Contrastive learning, as the most popular self-supervised learning algorithm in recent years, is different from generative encoding algorithms. Contrastive learning focuses on learning common features between similar instances and distinguishing differences between non-similar instances. van den Oord *et al.* first proposed InfoNCE loss, which encodes time-series data. By separating positive and negative samples, it can extract data-specific representations[21]. Based on similar ideas, He *et al.* achieved high performance in the field of image classification, by improving the similarity between the query vector and its corresponding key vector while reducing the similarity with the key vector of other images[23]. From the perspective of data augmentation, Chen *et al.* performed random cropping, inversion, grayscale, and other transformations on the image and extracted the invariant representation behind the image through contrastive learning[22]. The subsequent series of works[29–31] continued with a series of improvements, and the performance on some tasks is close to that of supervised learning algorithms.

From the above works, we can see that most of the previous opponent modeling work is to additionally input representations into neural networks for policy training. This paper provides another perspective on training a general policy to respond to various opponents by balancing the data in the replay buffer interacting with different opponent policies. Through the powerful representation extraction ability of contrastive learning, we distinguish various opponent policies at the representation level. It is worth noting that we only additionally use opponent observations, which is a looser setting compared to other work in multi-agent settings.

3. METHOD

3.1. Problem formulation

We describe the problem as a partially-observable stochastic game (POSG)[32] composed of a finite set $\mathcal{I} = \{1, 2, \dots, N\}$, a state space $\mathcal{S}$, the joint action space $\mathcal{A} = \mathcal{A}^1 \times \dots \times \mathcal{A}^N$, the joint observation space $\mathcal{O} = \mathcal{O}^1 \times \dots \times \mathcal{O}^N$, a transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denoting the transition probabilities between two states when given a joint action, and a reward function $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ for each agent $i \in \mathcal{I}$. Since we only focus on the performance of the agent under our control, we denote this agent by 1 and other agents are denoted by -1 with joint observation o^{-1} and joint action a^{-1}.

We design a set of K fixed policies for agent -1 $\Pi = \left\{\pi^{-1,k} \mid k = 1, \dots, K\right\}$, which can be rule-based (artificial) or network (pre-trained). Assume that an opponent policy π^w is a probability distribution on Π, $w \in \Delta(\Pi)$, which means at the beginning of each episode, the agent -1 samples a policy from Π with probability distribution w and executes this policy throughout the episode. Different from the setting of only switching between fixed policies in previous work, we allow more complex opponent policy changes, making the setting looser and more general.

Ideally, our goal is to find a general response policy π_θ parameterized by θ, which can maximize the reward of agent 1 against each opponent policy π^w. However, considering the reality, we maximize the minimum reward of π_θ against π^w:

$$\max_{\theta} \min_{w \in \Delta(\Pi)} \mathbb{E}_{\pi_\theta, \pi^w} \left[\sum_{t=0}^{H-1} \gamma^t r_t^1\right] \tag{1}$$

where r_t^1 is the reward of agent 1 at step t after performing the action a_t^1 determined by π_θ, H is the horizon of episode, and $\gamma \in (0, 1)$ is a discount factor. It is also important to note that at any time the agent 1 knows neither the policy type k of opponent nor the distribution w. This gives us the motivation to infer the type of opponent policy.

3.2. Representation extraction module

Different from previous opponent modeling methods that model opponent policies, we use a contrastive learning approach to self-supervised distinguish trajectories against different opponent policies, so that we only use the opponent's observations. We denote trajectory as $\tau = \{o_t^1, o_t^{-1}\}_{t=0}^{t=H-1}$ where o_t^1 and o_t^{-1} are the observations of agent 1 and agent -1 at step t, respectively. Given a set of trajectories $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_M\}$, the representation of each trajectory is self-supervised extracted by the CPC[21] algorithm.

Figure 1 shows the architecture of the contrastive predictive coding algorithm. First, we encode the observations by a multi-layer perceptron (MLP) to get a sequence of latent representation z_t:

$$z_t = f_{\mathrm{MLP}}(o_t^1, o_t^{-1}) \tag{2}$$

Then, use a gated recurrent unit (GRU) to extract the context information for the first t steps:

$$c_t = f_{\mathrm{GRU}}(z_{:t}) \tag{3}$$

In addition, we also need to define the similarity function f_k. To unify the dimensions, we use a bilinear product function:

$$f_k(z_{t+k}, c_t) = z_{t+k}^T W_k c_t \tag{4}$$

where $k = 1, \dots, H - t - 1$ and W_k is different for each k. For given set of trajectory $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_M\}$, c_t^i and z_{t+k}^i are calculated from τ_i. Since representations extracted from the same trajectory have similarities, we

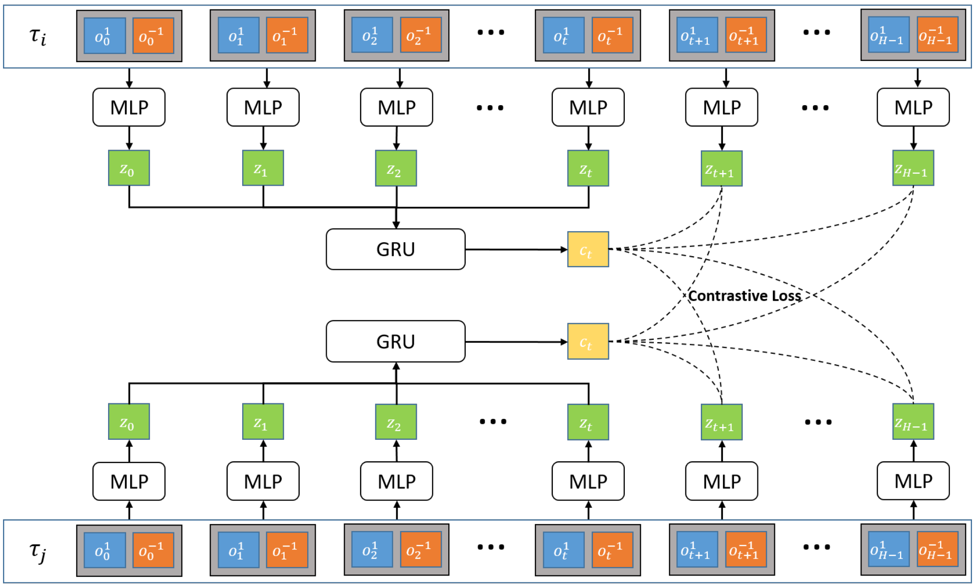


Figure 1. Overview of contrastive predictive coding (CPC), a representation extraction algorithm by contrasting positive and negative samples.The context c_t and subsequent state embeddings $\{z_{t+1}, z_{t+2}, \ldots, z_{H-1}\}$ are regarded as positive samples when they come from the same trajectory; otherwise, they are regarded as negative samples. By increasing the similarity between positive samples and reducing the similarity between negative samples, we obtain trajectory representations to distinguish different opponent policies.

maximize $f_k(z_{t+k}^i, c_t^j)$ when $i = j$ and minimize $f_k(z_{t+k}^i, c_t^j)$ when $i \neq j$. The InfoNCE loss is:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{M(H-t-1)} \sum_{k=1}^{H-t-1} \sum_{i=1}^{M} \left[\log \frac{exp(f_k(z_{t+k}^i, c_t^i))}{\sum_{j=1}^{M} exp(f_k(z_{t+k}^j, c_t^i))} \right] \tag{5}$$

where t is random sampling within a suitable range, M is the size of the trajectory set (batch size), and H is the horizon. Optimizing this loss will extract a unique representation of each trajectory that is different from others.

As described above, we self-supervise the extraction of policy representations from trajectories through contrastive learning, which can discriminate different opponent policies in representation space. Especially the contrast between positive and negative samples makes the representation highlight the differences between trajectories, which is beneficial for subsequent clustering operations.

3.3. Experience replay module

In Section 3.2, we introduce how to extract the representations of opponent policies from the trajectories that interact with opponents. Different from the previous approaches of directly using representations to assist training, we focus on another aspect, that is, the impact of non-stationary opponents on the experience replay. Experience replay is a commonly used method in reinforcement learning whose purpose is to improve the sample efficiency. When the replay buffer is full, the data are usually processed in a first-in, first-out (FIFO) manner. When the opponent uses a fixed policy, the environment can be treated as a deterministic MDP, and FIFO is feasible. When the opponent is non-stationary, the replay buffer will pass through data that interacts with different types of opponent policies. The decrease in the proportion of certain types of data will affect the effectiveness against such an opponent, and the loss of old data may lead to the forgetting of learned strategies. Therefore, we design new data in and out, a mechanism to keep as many types of trajectory data as possible in the replay buffer.

We cluster the trajectory data in the replay buffer in the representation space, and, for the representation, $z^i_{:H}$ and $z^j_{:H}$ of the trajectories τ_i and τ_j, we use the average Euclidean distance to measure the distance between them:

$$d_{i,j} = \frac{1}{H} \sum_{h=0}^{H-1} ||z^i_h, z^j_h|| \tag{6}$$

For all trajectories in the replay buffer, we can calculate the representation distance matrix D by Equation (6). Additionally, the truncation method can be used for trajectory representations of different lengths, or the dynamic time warping (DTW) can be used instead of the Euclidean distance.

Since the number of opponent policies is unknown, some clustering methods such as K-means are not suitable for use. We use agglomerative clustering to distinguish trajectory representations in the replay buffer, which is implemented in the standard algorithm library scikit-learn, and the clustering threshold is set as the average distance of all trajectory representations. Then, the labels of the trajectories that interact with the opponents are obtained in a self-supervised manner.

To balance the proportion of different types of data in the replay buffer, we no longer pop the oldest data when the replay buffer is full, but pop the oldest data from the largest class based on the clustering results. This ensures the dynamic balance of various types of data to a certain extent. Even if a certain type of opponent policy has a very low probability of appearing in a period, the data interacting with it can maintain a certain proportion in the replay buffer, thereby avoiding policy forgetting. However, this approach will lead to some useless old data existing in the replay buffer for a long time, reducing the training effect of reinforcement learning. We introduce a probability threshold ρ, where the replay buffer pops the oldest data from the largest class with the probability of ρ and pops the oldest data from the entire replay buffer with the probability of $1 - \rho$. This allows the data that hinder training to be popped. In this paper, we set $\rho = 0.9$.

3.4. Combine with reinforcement learning

This section describes the overall algorithm flow in combination with the classic reinforcement learning algorithm soft actor–critic (SAC) whose optimization goal is:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot \mid \mathbf{s}_t)) \right] \tag{7}$$

where $\mathcal{H}(\pi(\cdot \mid \mathbf{s}_t))$ is the additional policy entropy added to encourage exploration and α is the temperature parameter determines the relative importance of the entropy term. However, our method can be combined with any off-policy reinforcement learning algorithm.

Since the training speed of representation learning is much faster than that of reinforcement learning, we set the training frequency F_c for it to balance their learning rate. In addition, this also considers that the flow of data in the replay buffer in the short term will not change the data distribution in it. Considering that the introduction of the clustering requires a large amount of extra computation, and the data that newly entered replay buffer will not be popped in the short term, we update the labels of trajectory representations by clustering every F_l episode.

The complete algorithm is described in Algorithm 1. The training of representation learning and reinforcement learning process alternately. Since the FIFO rule is still followed in the class, our method will not have too much influence on the training of reinforcement learning; at the same time, the diversity of the data in the replay buffer is guaranteed as much as possible, so that the policy forgetting caused by the non-stationary of the opponent policy is avoided.

Algorithm 1 SAC with TRC.

```
Require: Initialize SAC parameter vector $\theta$, CPC parameter vector $\varphi$, total episode $T$, episode horizon $H$,
    batch size $M$, CPC training frequency $F_c$, labels update frequency $F_l$, and threshold $\rho$.
for episode $i = 0 \dots T - 1$ do
    opponent choose policy $\pi^{-1}$
    for step $t = 0 \dots H - 1$ do
        $a^1_t \sim \pi_\theta\left(a^1_t \mid o^1_t, o^{-1}_t\right)$
        $a^{-1}_t \sim \pi^{-1}\left(a^{-1}_t \mid o^1_t, o^{-1}_t\right)$
        $o^1_{t+1}, o^{-1}_{t+1} \sim p\left(o^1_{t+1}, o^{-1}_{t+1}, \mid o^1_t, o^{-1}_t, a^1_t, a^{-1}_t\right)$
        $\tau_i \leftarrow \tau_i \cup \{(o^1_t, o^{-1}_t, a^1_t, r(o^1_t, o^{-1}_t, a^1_t, a^{-1}_t), o^1_{t+1}, o^{-1}_{t+1})\}$
    end for
    $D \leftarrow D \cup \tau_i$
    if $i$ mod $F_c$ == 0 then
        Sample trajectory batch $\mathcal{T}$ from $D$
        Update $\varphi$ by Equation (5)
    end if
    if $|D|$ == $M$ then
        if random sample a probability value greater than $\rho$ then
            Pop the oldest trajectory from $D$
        else
            if $i$ mod $F_l$ == 0 then
                Compute $z_{:H} = f_\varphi(\tau)$ for each $\tau$ in $D$
                Compute distance matrix of trajectory representations by Equation (6)
                Cluster trajectory representations by agglomerative clustering
            end if
            Pop the oldest trajectory from the largest class
        end if
    end if
    Update $\theta$ by SAC algorithm.
end for
```

4. RESULTS

We evaluate our approach in a more complex soccer environment and compare the average returns during RL training against three baselines. We also discuss the impact of the proportion of data in the replay buffer on reinforcement learning training and the improvement of our approach to the diversity of trajectories in the replay buffer. In addition, we analyze representational clustering by t-distributed stochastic neighbor embedding (t-SNE) to analyze the properties of different adversary policies at the representational level.

4.1. Game description

Soccer is a classic competitive environment that has been used by many opponent modeling approaches[11,13] to verify their performance. We extend the rules based on the classic soccer environment and design more complex rule-based opponent policies based on this. As shown in Figure 2, the environment is a 15 × 15 grid world, and there are two goals on each end line. At the beginning of the episode, the two agents are in the center of their respective end lines with 0 energy, and one random agent holds the ball. Each agent has 13 optional actions, moving to any of the 12 grid points within a two-grid range around itself or staying in place, but moving 2 grids requires 2 energy. The agent with the ball recovers 0.5 energy per step, while the agent without the ball recovers 1 energy per step, and the upper limit of energy is 2. When both agents are about to enter the same grid, they stop in place and exchange the ball possession. When the agent dribbles the ball

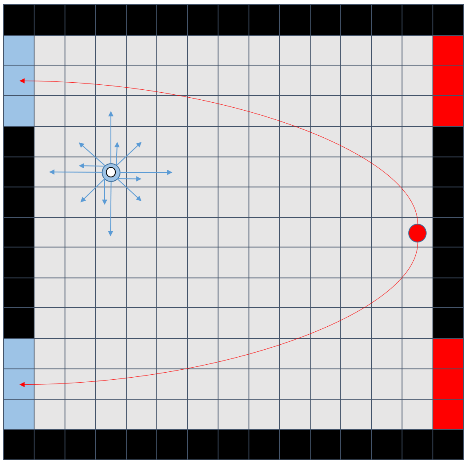
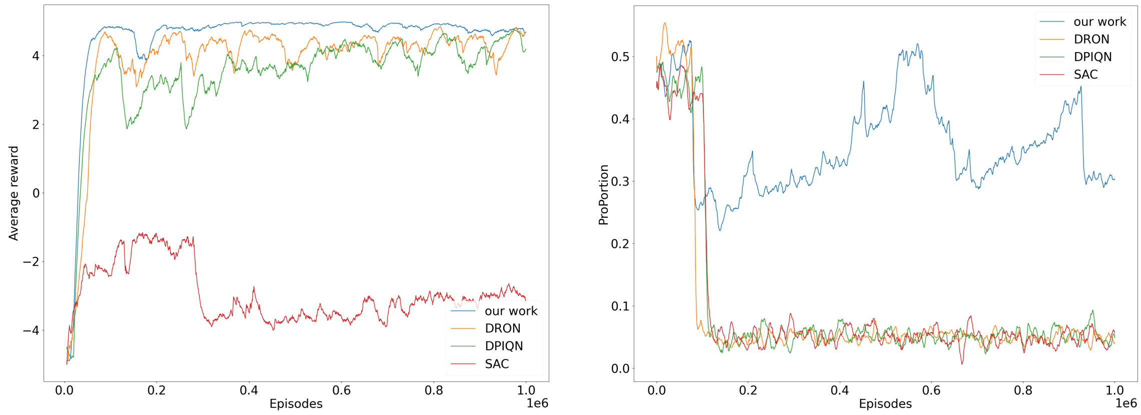


Figure 2. The configuration of soccer. The goal of each agent is to drive the ball into the opponent's goal.



(a)

(b)

Figure 3. (a) The average reward curve of interacting with opponent policy π_1^{-1}; and (b) the proportion change curve of opponent π_1^{-1} trajectory in replay buffer.

into the opponent's goal, it gets a +5 reward, while the opponent gets a −5 reward, and then ends this episode. If the interaction exceeds 50 steps, the episode will also be terminated and each agent will get 0 rewards. The position, energy, and ball possession are fed back to the agent as observation.

The opponent policies are designed to be random policies based on given rules, which makes it more complex. Specifically, we design two base opponent policies π_1^{-1} and π_2^{-1} with different styles. π_1^{-1}: Keep away from the opponent while attacking the upper goal when holding the ball, and get close to the opponent when not holding the ball. π_2^{-1}: Keep away from the opponent while attacking the lower goal when holding the ball, and defend near its end line when not holding the ball. As described in Section 3.1, we define $w \in [0, 1]$ as a class of opponent policies, and, at the beginning of the episode, the opponent chooses a policy from $\{\pi_1^{-1}, \pi_2^{-1}\}$ with probability distribution $\{w, 1 - w\}$.

4.2. Non-stationary opponent

We make the opponent policy switch from $w = 0.5$ to $w = 0.05$ at step 100k to observe the performance of the agent training by different algorithms in a non-stationary environment. Figure 3a shows the comparison of the reward curves of our algorithm and three baselines against opponent policy π_1^{-1}. In these baselines, vanilla

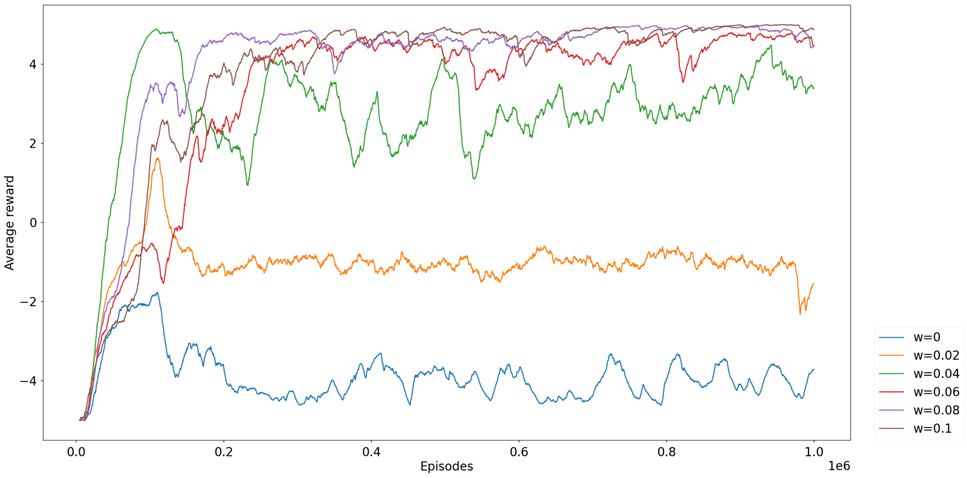


Figure 4. The average reward curve of interacting with opponent policy π_1^{-1} when w change from 0.5 to 0, 0.02, 0.04, 0.06, 0.08, and 0.1.

SAC uses no opponent information and performs the worst. DRON uses the opponent's observation as an additional input, while DPIQN further uses the opponent's actions to obtain the representations of opponent policy to aid training. However, they both perform worse than our work due to a lack of consideration of data balance. Figure 3b shows the change in the proportion of interaction trajectories with opponent strategy π_1^{-1} in the replay buffer. It can be seen that, when the probability of an opponent policy decreases, only our method can maintain a relatively high proportion of the data obtained by interacting with it in the replay buffer. This improves responsiveness to such an opponent policy and avoids forgetting the learned policy.

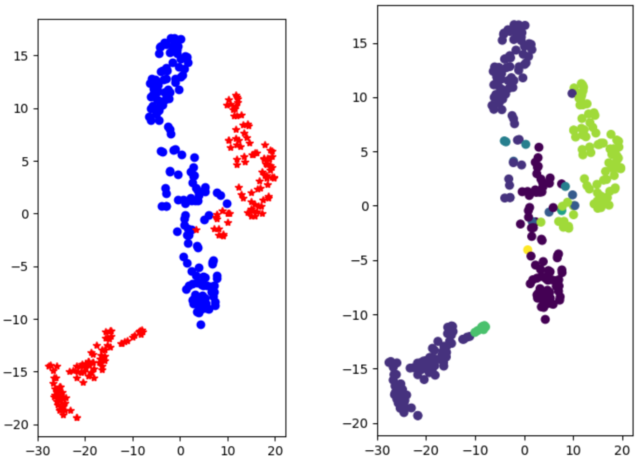
To explain the impact of data ratio on policy forgetting in more detail, we make the opponent policy change from $w = 0.5$ to $w = 0, 0.02, 0.04, 0.06, 0.08, 0.1$ at step 100k and use SAC for training with the other conditions remaining the same. As shown in Figure 4, when a certain opponent policy appears very infrequently, a small proportional increase in the replay buffer can bring about higher performance improvement, but, for data that already exist in significant proportions, the impact of adjusting the data ratio is minimal. This also explains our motivation to balance the proportion of various data.

4.3. Analysis of clustering

In Section 4.2, we show the performance of the algorithm and analyze the rationale behind data balancing. In this section, from the perspective of policy representation, we analyze the clustering properties of the policy representations obtained by contrastive learning in the representation space. Figure 5 shows the visualization of trajectory encoding after dimensionality reduction by t-SNE. Self-supervised contrastive learning is not very accurate in distinguishing two types of opponent policies. Because policies may have similar parts, a type of policy can also be decomposed into several more refined sub-policies. Self-supervised learning of policy representations only with trajectory information can only be used for coarse clustering. However, our algorithm does not rely on extremely accurate trajectory clustering and strategy identification but balances the proportion of various trajectory data generally. This also makes the algorithm have certain robustness.

5. CONCLUSION

This paper constructs a general sampling algorithm based on data balance for multi-agent non-stationary problems. The trajectory representation of the interaction with the opponent is extracted by comparative learning, and then the representation is distinguished by hierarchical clustering. Finally, the data balance in the replay buffer is realized by changing the order of in and out of the replay buffer. We get better performance against



(a) (b)

Figure 5. t-SNE projection of the embeddings in the soccer environment: (a) the two colors represent the two base opponent policies π_1^{-1} and π_2^{-1}; and (b) the different colors represent the classes of trajectory representations encoded by the contrastive learning.

a non-stationary opponent. In particular, we only use the observation information of the opponent, and the setting is looser than other opponent modeling algorithms. In the future, we would like to combine multi-task learning algorithms to learn different opponent policies as different tasks and explore more efficient ways to distinguish opponent policies.

DECLARATIONS

Authors' contributions

Designed and run experiments: Lv Y
Made substantial contributions to conception and design of the study: Zheng Y
Provided administrative, technical, and material support: Hao J

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that they have no conflicts of interest to this work.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Littman ML. Markov games as a framework for multi-agent reinforcement learning. Machine Learning Proceedings 1994. Elsevier; 1994. pp. 157-63. DOI
2. Busoniu L, Babuska R, De Schutter B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans Syst, Man, Cybern C* 2008;38:156-72. DOI
3. Vinyals O, Babuschkin I, Czarnecki WM, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 2019;575:350-4. DOI
4. Hernandez-Leal P, Kaisers M, Baarslag T, de Cote EM. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017. DOI
5. Papoudakis G, Christianos F, Rahman A, Albrecht SV. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019. DOI
6. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. DOI
7. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. DOI
8. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning, PMLR, 2018. p. 1861-70.
9. Hernandez-leal P, Kartal B, Taylor ME. A survey and critique of multiagent deep reinforcement learning. *Auton Agent Multi-Agent Syst* 2019;33:750-97. DOI
10. Hernandez-Leal P, Taylor ME, Rosman B, et al. Identifying and tracking switching, non-stationary opponents: A bayesian approach. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence, 2016.
11. Zheng Y, Meng Z, Hao J, et al. A deep bayesian policy reuse approach against non-stationary agents. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018. p. 962–972.
12. Yang T, Hao J, Meng Z, et al. Towards efficient detection and optimal response against sophisticated opponents. In: Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019.
13. He H, Boyd-Graber J, Kwok K, Daumé III H. Opponent modeling in deep reinforcement learning. In: International Conference on Machine Learning, PMLR, 2016. p. 1804-13. DOI
14. Hong ZW, Su SY, Shann TY, Chang YH, Lee CY. A deep policy inference q-network for multi-agent systems. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018. p. 1388-.96.
15. Raileanu R, Denton E, Szlam A, Fergus R. Modeling others using oneself in multi-agent reinforcement learning. In: International Conference on Machine Learning, PMLR, 2018. p. 4257-66.
16. Foerster J, Chen RY, Al-Shedivat M, Whiteson S, Abbeel P, Mordatch I. Learning with opponent-learning awareness. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018. p. 122-30.
17. Lin LJ. Reinforcement learning for robots using neural networks. Carnegie Mellon University, 1992. Available from: https://dl.acm.org/doi/book/10.5555/168871 [Last accessed on 7 Jun 2022]
18. Chaudhry A, Rohrbach M, Elhoseiny M, et al. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. DOI
19. Chrysakis A, Moens MF. Online continual learning from imbalanced data. In: International Conference on Machine Learning, PMLR, 2020. p. 1952-61.
20. Khetarpal K, Riemer M, Rish I, Precup D. Towards continual reinforcement learning: a review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020. DOI
21. van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. DOI
22. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International conference on machine learning, PMLR, 2020. p. 1597–1607.
23. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. p. 9729-38.
24. Laskin M, Srinivas A, Abbeel P. Curl: Contrastive unsupervised representations for reinforcement learning. In: International Conference on Machine Learning, PMLR, 2020. p. 5639-50.
25. Koopmans T. Activity analysis of production and allocation. 1951. Available from: https://cowles.yale.edu/sites/default/files/files/pub/mon/m13-all.pdf [Last accessed on 7 Jun 2022]
26. Carmel D, Markovitch S. Model-based learning of interaction strategies in multi-agent systems. *Journal of Experimental & Theoretical Artificial Intelligence*1998;10:309-32. DOI
27. Rosman B, Hawasly M, Ramamoorthy S. Bayesian policy reuse. *Mach Learn* 2016;104:99-127. DOI
28. de Weerd H, Verbrugge R, Verheij B. How much does it help to know what she knows you know? an agent-based simulation study. *Artificial Intelligence* 2013;199-200:67-92. DOI
29. Chen X, Fan H, Girshick R, He K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. DOI
30. Chen T, Kornblith S, Swersky K, Norouzi M, Hinton GE. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems* 2020;33:22243-55.
31. Grill JB, Strub F, Altché F, et al. Bootstrap your own latent - a new approach to self-supervised learning. *Advances in Neural Information*

Processing Systems 2020;33:21271-84.

32. Hansen EA, Bernstein DS, Zilberstein S. Dynamic programming for partially observable stochastic games. *AAAI* 2004;4:709-15. Available from: https://www.aaai.org/Papers/AAAI/2004/AAAI04-112.pdf [Last accessed on 7 Jun 2022]

Yu *et al*. *Intell Robot* 2022;2:180-99
DOI: 10.20517/ir.2022.10

Intelligence & Robotics

Review

Open Access

Check for updates

Designs, motion mechanism, motion coordination, and communication of bionic robot fishes: a survey

Zhiwei Yu[1], Kai Li[1], Yu Ji[1], Simon X. Yang[2]

[1]College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, Jiangsu, China.
[2]School of Engineering, University of Guelph, 50 Stone Road East, Guelph, ON N1G 2W1, Canada.

Correspondence to: Dr. Zhiwei Yu, College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, Jiangsu, China. E-mail: yuzhiwei@nuaa.edu.cn

How to cite this article: Yu Z, Li K, Ji Y, Yang SX. Designs, motion mechanism, motion coordination, and communication of bionic robot fishes: a survey. *Intell Robot* 2022;2:180-99. https://dx.doi.org/10.20517/ir.2022.10

Received: 22 Mar 2022 **First Decision:** 19 May 2022 **Revised:** 2 Jun 2022 **Accepted:** 27 Jun 2022 **Published:** 29 Jun 2022

Academic Editors: Jianjun Ni, Daqi Zhu **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

Abstract

In the last few years, there have been many new developments and significant accomplishments in the research of bionic robot fishes. However, in terms of swimming performance, existing bionic robot fishes lag far behind fish, prompting researchers to constantly develop innovative designs of various bionic robot fishes. In this paper, the latest designs of robot fishes are presented in detail, distinguished by the propulsion mode. New robot fishes mainly include soft robot fishes and rigid-soft coupled robot fishes. The latest progress in the study of the swimming mechanism is analyzed on the basis of summarizing the main swimming theories of fish. The current state-of-the-art research in the new field of motion coordination and communication of multiple robot fishes is summarized. The general research trend in robot fishes is to utilize more efficient and robust methods to best mimic real fish while exhibiting superior swimming performance. The current challenges and potential future research directions are discussed. Various methods are needed to narrow the gap in swimming performance between robot fishes and fish. This paper is a first step to bring together roboticists and marine biologists interested in learning state-of-the-art research on bionic robot fishes.

Keywords: Bionic robot fish, motion mechanism, motion coordination, group communication

1. INTRODUCTION

Propellers are frequently used as actuators in conventional underwater robots, and their propulsion efficiency is only 40%-50%. Furthermore, their shapes employ a non-bionic structure that cannot be integrated into the underwater environment, making close observation of underwater organisms difficult. Fish have undergone extensive natural selection and can swim with an efficiency of more than 90%[1]. Fish also have distinct advantages in terms of speed, maneuverability, and stealth[2-6]. For example, swordfish can reach a speed up to 30 $\mathrm{m \cdot s^{-1}}$[3]. Bionic robot fishes, which treat fish as bionic objects, can effectively absorb these advantages to overcome the defects of traditional underwater robots and become more effective tools for ocean exploration.

The propulsion modes of fish are usually classified into two categories according to the body parts used for propulsion, namely body and/or caudal fin (BCF) propulsion and median and/or paired fin (MPF) propulsion[7,8]. It is worth noting that the median fin refers to the dorsal or anal fin, while the paired fin refers to the pectoral or pelvic fin. Taking tilapia as an example, the structure and position of each fin are shown in Figure 1. The BCF propulsion mode, in which the body and/or caudal fin acts as a propeller, is the most common in fish and first discovered by researchers. This propulsion mode has the advantages of high swimming speed and quick start performance, making it suitable for applications requiring high speed or instantaneous acceleration[9]. The median and/or paired fin acts as a propeller in the MPF propulsion mode. This propulsion mode has the advantages of high maneuverability, high propulsion efficiency, and good stability, making it suitable for applications requiring maneuvering to turn or long-term swimming, as well as scenes with rapid water flow[10]. After summarizing recent research results, we show that existing robot fishes already have the BCF and MPF combined (BCF-MPF) propulsion mode. This propulsion mode is based on the cooperation of the caudal and pectoral fins. With proper design, it is capable of balancing swimming speed and propulsion efficiency, which has a wider application than either individually. Furthermore, it is a promising research topic. The basic elements of the three propulsion modes are summarized in Table 1.

Some review papers focus on the motion control of robot fishes[11-14], while others focus on the design, fabrication, and propulsion methods of robot fishes[15-17]. There is also a review paper that focuses on the perception of robot fishes[18]. However, the majority of them were published more than five years ago. In these years, unprecedented attention has been paid to the study of bionic robot fishes. Related achievements have proliferated and enriched the research in the field of robot fishes. As a result, this paper provides a new survey on various major fields of robot fishes, addressing some gaps in related fields. Figure 2 depicts the paper's framework, which includes three objectives. Firstly, we provide a comprehensive survey of the most recent designs of robot fishes, as well as the most recent progress in the study of motion mechanism. A new field of study, namely motion coordination and communication of multiple robot fishes, is discussed. Second, based on the survey, the challenges of current research and potential future research directions are summarized. Three aspects are included: the gap between robot fishes and fish in terms of swimming performance, methods to study the swimming mechanism of robot fishes, and the motion coordination and communication of multiple robot fishes. Finally, a summary of the paper is provided.

The rest of the article is organized as follows. Section 2 elaborates on the latest designs of robot fishes. Section 3 analyzes the motion mechanism of robot fishes. The motion coordination and communication of multiple robot fishes are discussed in Section 4. Section 5 provides a comprehensive discussion on the challenges and future works. Finally, in Section 6, some concluding remarks are made.

2. DESIGNS OF ROBOT FISHES

According to their body structure, robot fishes are classified into three types: rigid, soft, and rigid-soft coupled. The strengths and weaknesses of different body structures are summarized in Table 2. The rigid

Table 1. Propulsion modes of robot fishes

Propulsion modes	Propellers	Strengths	Applications
BCF	Body and/or caudal fin	1. High swimming speed 2. High quick start performance	Requiring high speed Requiring instantaneous acceleration
MPF	Median and/or paired fin	High maneuverability High propulsion efficiency Good stability	Requiring maneuvering to turn Requiring long-term swimming Rapid water flow
BCF-MPF	Cooperation of the caudal and pectoral fins	Balancing swimming speed and propulsion efficiency	Broader than either individually

Table 2. The body structures of robot fishes

Body structures	Strengths	Weaknesses
Rigid	High swimming speed	Poor maneuverability
Soft	Great maneuverability	Low swimming speed
Rigid-soft coupled	Achieving great maneuverability while generating high swimming speed with a reasonable design	

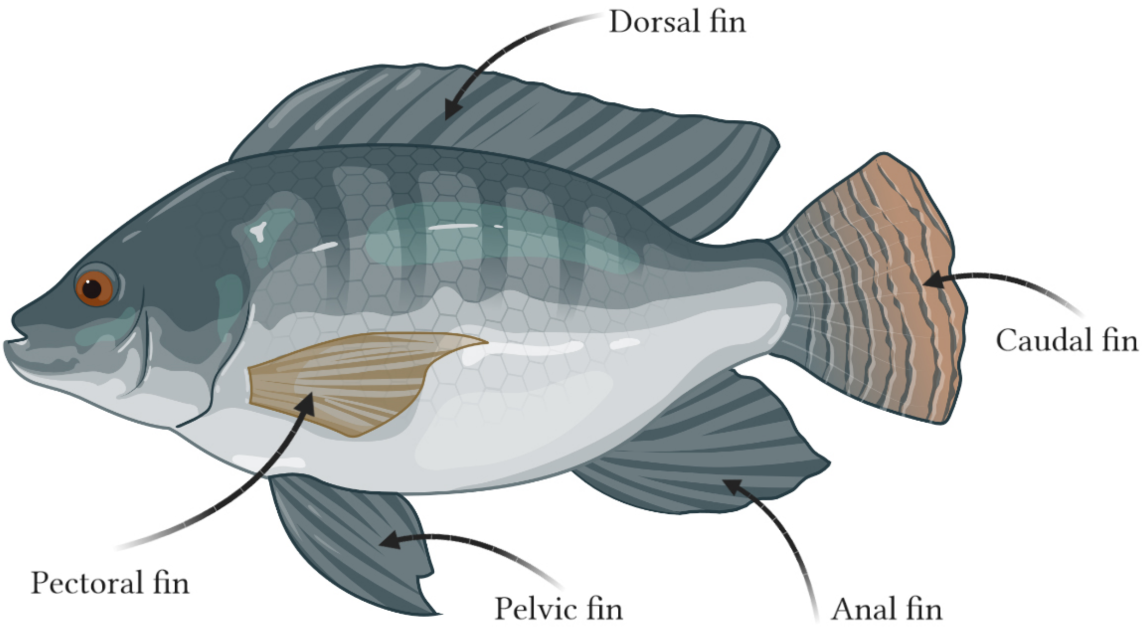


Figure 1. Types of fins in tilapia.

robot fish has high swimming speed, but its maneuverability is poor. In contrast, the soft robot fish has great maneuverability, but its swimming speed is low. The rigid-soft coupled robot fish lies between the two. Through reasonable design, it can achieve great maneuverability while generating high swimming speed. The rigid robot fish has received little attention in recent years. This is primarily due to the fact that the rigid structure of the rigid robot fish is far from the elastic skin and muscles of fish. As a result, we only discuss soft and rigid-soft coupled robot fishes in this paper.

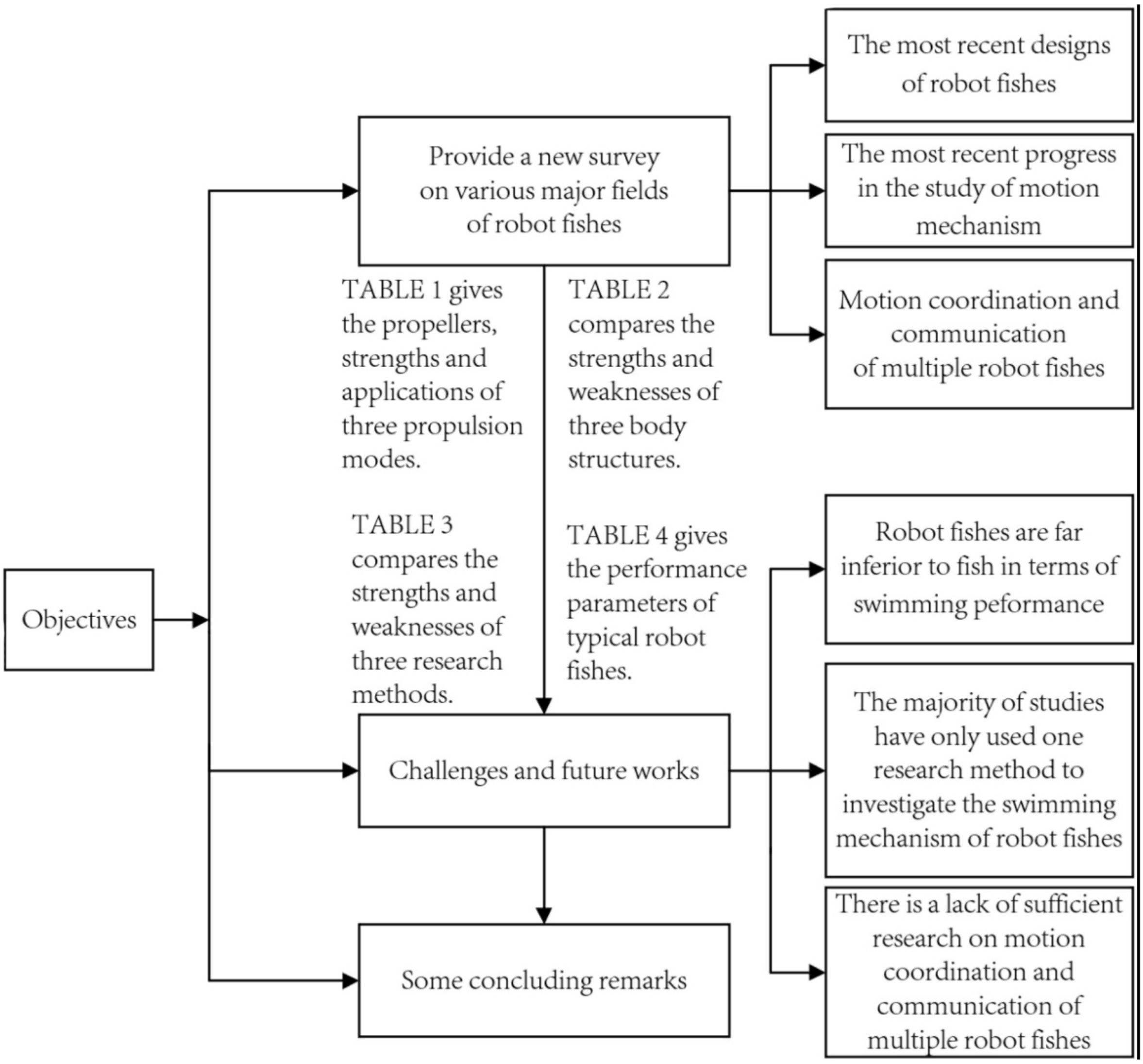

Objectives
Provide a new survey on various major fields of robot fishes
TABLE 1 gives the propellers, strengths and applications of three propulsion modes.
TABLE 2 compares the strengths and weaknesses of three body structures.
TABLE 3 compares the strengths and weaknesses of three research methods.
TABLE 4 gives the performance parameters of typical robot fishes.
Challenges and future works
Some concluding remarks
The most recent designs of robot fishes
The most recent progress in the study of motion mechanism
Motion coordination and communication of multiple robot fishes
Robot fishes are far inferior to fish in terms of swimming peformance
The majority of studies have only used one research method to investigate the swimming mechanism of robot fishes
There is a lack of sufficient research on motion coordination and communication of multiple robot fishes

Figure 2. The review framework.

2.1. Robot fishes in BCF propulsion mode

2.1.1. Soft robot fishes

This robot fish generally uses intelligent materials or other special devices to simulate the muscles of fish, expecting to significantly improve the swimming performance. The first attempt was made by Katzschmann *et al.*[19]. The robot fish SoFi was designed by them, which used a soft fluid actuator to simulate muscle tissue. The soft caudal fin had two lateral chambers symmetrical along the central axis. A gear pump drove fluid flow from one side of the chamber to another, causing the caudal fin to bend. SoFi successfully swam around aquatic life at depths of 0-18 m and effectively integrated into the marine environment. However, SoFi still has room for improvement, such as optimizing the geometry of the tail section. Dielectric elastomer actuators (DEAs), a type of smart material, are also used in robot fishes. Shintake *et al.* attached two DEAs to both sides of the robot fish's body, as shown in Figure 3A[20]. The DEAs were stretched by the same length so that the initial state of the robot fish was straight. The voltage was applied to each side of the body in turn, so that one side of the DEA was elongated while the other side was contracted. As a result, the robot fish's body oscillated from one side to another side, causing the caudal fin to oscillate. The maximum

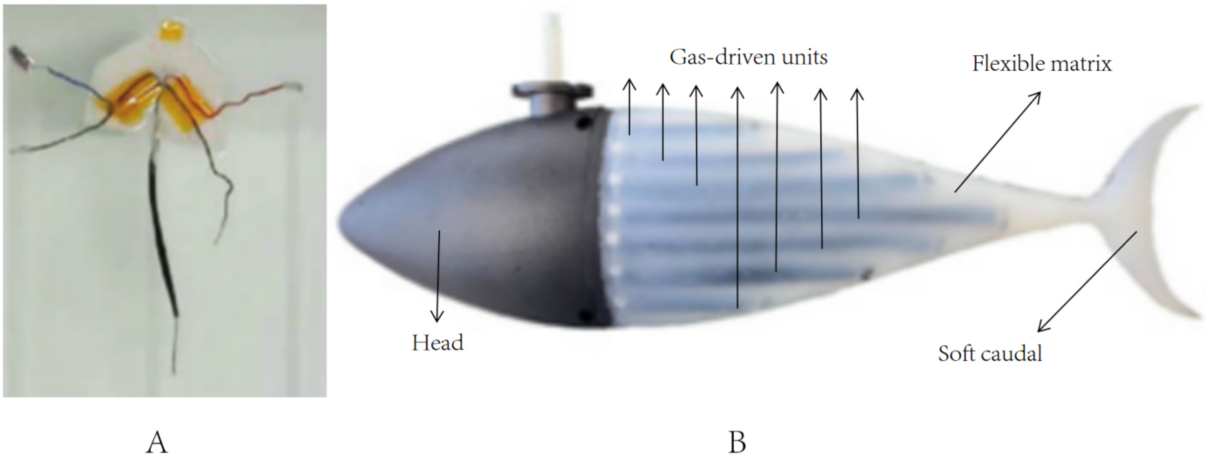


Figure 3. Soft robot fishes in BCF propulsion mode: (A) a robot fish with DEAs[20]; and (B) Flexi-Tuna[21]. BCF: Body and/or caudal fin; DEAs: dielectric elastomer actuators.

swimming speed of the robot fish was 0.25 BL·s^{-1} at 0.75 hertz oscillation frequency. However, the authors needed to test the fish in different sizes and swimming types (e.g., turning) to figure out how much swimming ability the fish had. Liu *et al*. proposed using gas-driven units to simulate muscle fibers of fish and successfully designed the robot fish Flexi-Tuna[21]. As shown in Figure 3B, 14 drive units were symmetrically distributed on both sides of the robot fish's body. Then, alternating pressure was applied to the drive units to make the tail oscillate back and forth. According to the results, under the optimal frequency of 3.5 Hz, the maximum swing angle of Flexi-Tuna was 20° and the maximum thrust was 0.185 N. This research realized the application of artificial muscles in robot fishes and provided new ideas for the design of soft robot fishes. However, some optimizations, such as variable stiffness design of caudal fin, are still needed to achieve better swimming performance of robot fishes.

2.1.2. Rigid-soft coupled robot fishes

In recent years, researchers have come up with some new ideas to improve the swimming performance of this type of robot fish.

The headshaking of robot fishes leads to an increase of water resistance, which in turn reduces their swimming speed. To address this issue, Liao *et al*. proposed using two caudal fins rather than a single caudal fin[22]. Caudal fins were mounted symmetrically on the tail of the robot fish, as shown in Figure 4A. They were designed to flap oppositely to offset lateral forces, which in turn prevented the headshaking. The robot fish had three motions: oscillatory motion, jet motion, and oscillatory and jet cooperative motion. A suitable motion type could be chosen based on the distance between two caudal fins. This indicated that the robot fish had great flexibility. According to the experimental results, the robot fish could reach the speed of 2.5 body lengths per second (BL·s^{-1}), demonstrating excellent swimming speed.

Researchers have made great progress in mimicking the body structure of fish. Coral *et al*. created a robot fish using actuators made of shape memory alloys (SMAs)[23]. As shown in Figure 4B, these actuators were bent into a continuous structure to resemble the fish backbone. Bio-inspired synthetic skin was used to mimic the skin of fish. Nevertheless, the authors only verified the feasibility of this scheme. Zhu *et al*. created Tunabot by mimicking the body structure of tuna and mackerel and discussed the influence of oscillation frequency in depth[24]. The robot fish had a streamlined shape with an elastic skin overlaid on the actuator system. Tunabot swam at a maximum tail-beat frequency of 15 hertz, reaching 4 BL·s^{-1} according to

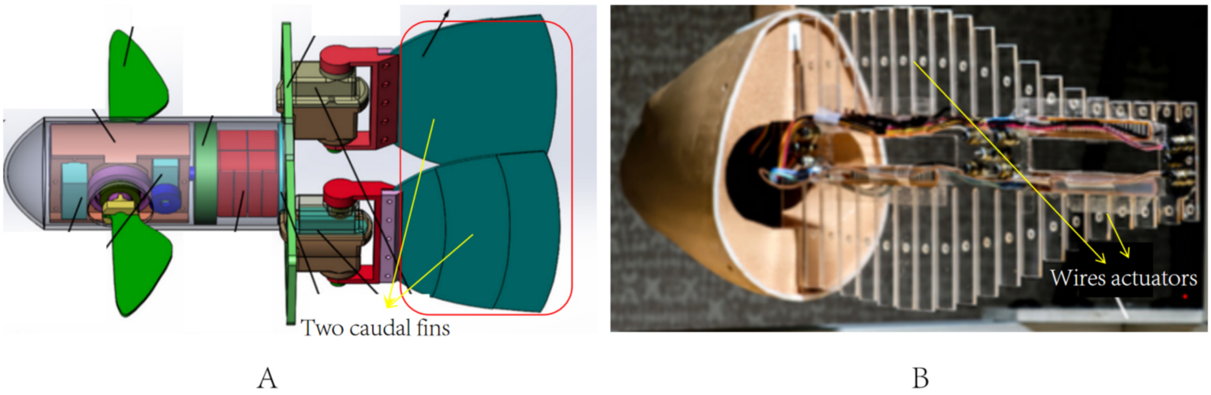


Figure 4. Rigid-soft coupled robot fishes in BCF propulsion mode: (A) a robot fish with two caudal fins[22]; and (B) wires actuators of the robot fish[23]. BCF: Body and/or caudal fin.

experiments. Tunabot could swim 9.1 km if it swam at 0.4 m·s^{-1} or 4.2 km if it swam at 1 m·s^{-1} while powered by a 10 Wh battery pack. This highlighted the capabilities of high-frequency swimming. This provided new ideas to improve the swimming performance of robot fishes. The variable stiffness design of the robot fish is also an imitation of fish. TenFiBot, a robot fish with variable stiffness, was designed by Chen and Jiang[25]. The whole structure of TenFiBot was a tandem structure with multiple variable-stiffness tensegrity joints (VSTJs). The preload of the springs on the VSTJs could be adjusted to change the stiffness distribution on the TenFiBot's body. Experiments demonstrated that the change of stiffness distribution directly affected the swimming performance (such as swimming speed) of the robot fish. By changing the stiffness distribution of the robot fish, its swimming performance could be greatly improved.

2.2. Robot fishes in MPF propulsion mode

2.2.1. Soft robot fishes

This robot fish tends to be designed with smart materials and is smaller in size. As MPF propulsion mode is adopted, it has greater maneuverability. Therefore, it is ideal for applications in special environments, such as fine pipes, deep sea, etc. Inspired by the hadal snail-fish, which lives at 8000 m water depth, Li *et al*. designed an untethered soft robot fish that could withstand extreme hydrostatic pressure[26]. The robot fish was driven by DEAs. The electronic components of the robot fish were decentralized on several smaller printed circuit boards, which could effectively reduce the shear stress between components. This ensured that the robot fish could withstand extreme water pressure. The robot fish successfully swam at a depth of 10,900 m in the Mariana Trench, showing great potential for application in deep-sea exploration.

2.2.2. Rigid-soft coupled robot fishes

A key condition to achieving high swimming performance is to adjust the distribution of soft and hard structures in robot fishes. As shown in Figure 5A, a robot fish with cartilages and soft tissues was designed by Yurugi *et al*.[27]. Experiments revealed that adding cartilages to the fins of the robot fish could improve swimming efficiency. The researchers also investigated the fish's swimming behavior. As shown in Figure 5B, Ma *et al*. designed a robot fish driven by the oscillating and twisting of the pectoral fins after studying the pectoral fin movement of the cownose ray[28]. The pectoral fins simultaneously realized oscillating motion and chordwise twisting motion. The maximum swimming speed of the robot fish was 0.94 BL·s^{-1}, and the turning radius was nearly zero. This reflected the excellent turning performance and high swimming speed of the robot fish. These authors should conduct additional research into the effect of pectoral fin flexibility on swimming performance.

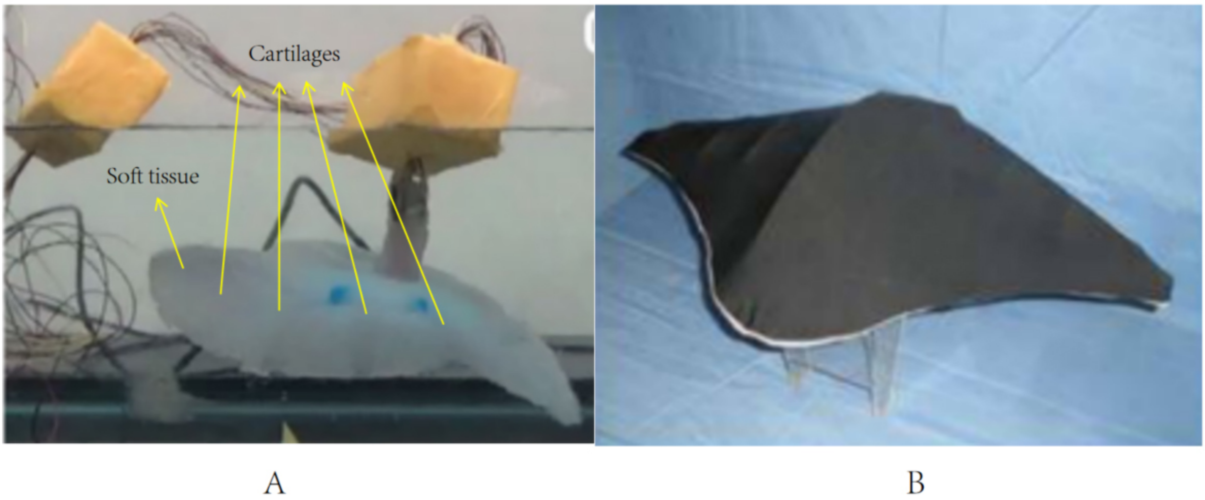


Figure 5. Rigid-soft coupled robot fishes in MPF propulsion mode: (A) a robot fish with soft tissue and cartilages.[27]; and (B) bionic cownose ray robot fish[28]. MPF: Median and/or paired fin.

2.3. Robot fishes in BCF-MPF propulsion mode

2.3.1. Soft robot fishes

The caudal fin of this robot fish mainly serves a steering function, and the pectoral fins mainly provide propulsion. Zhang *et al*. first tried to build a robot fish[29]. The dielectric elastomers (DEs) were attached to the elastic frame, and the variable voltage was applied to drive the pectoral fins up and down to generate forward thrust. A steering electrical servo drove the caudal fin deflection angle for turning. Figure 6A depicts the position of the pectoral and caudal fins. Unfortunately, the authors did not test the performance of this robot fish. Li *et al*., inspired by manta rays, designed a soft electronic robot fish driven by DEAs, as shown in Figure 6B[30]. The speed of the robot fish was 0.69 BL·s^{-1}. It could use the surrounding water as an electric ground and swim for up to 3 h on a single charge. This thoroughly illustrated the robustness of this robot fish.

2.3.2. Rigid-soft coupled robot fishes

The caudal fin of this robot fish is able to oscillate significantly and rapidly, allowing for high propulsion power. Simultaneously, the pectoral fins have multiple degrees of freedom, allowing for great maneuverability. As a result, the excellent swimming performance of these robot fishes has attracted the interest of many researchers. This was attempted by Li *et al*., who created the robot fish shown in Figure 7[31]. The caudal fin of the robot fish had three rigid joints, which ensured its high flexibility. The pectoral fins could perform rotary motion and forward-backward motion, and the two motions were completely independent. This robot fish could reach a turning speed of 0.6 radians per second (rad·s^{-1}) with the coordinated propulsion of the caudal and pectoral fins. This provided the robot fish with more turning options and higher maneuverability. Zhong *et al*. designed a new type of robot fish[32]. The caudal fin of the robot fish was driven by wires, which could be deformed along a chordwise direction or both chordwise and spanwise directions. Flapping and rowing motions were possible with the pectoral fins. The results show that, without using the pectoral fins, the turning radius of the robot fish was 0.6 BL; with the pectoral fins, the turning radius was reduced to 0.25 BL. This clearly had higher maneuverability. These experiments only tested the turning performance of these robot fishes and did not test their performance in other swimming types (e.g., straight swimming).

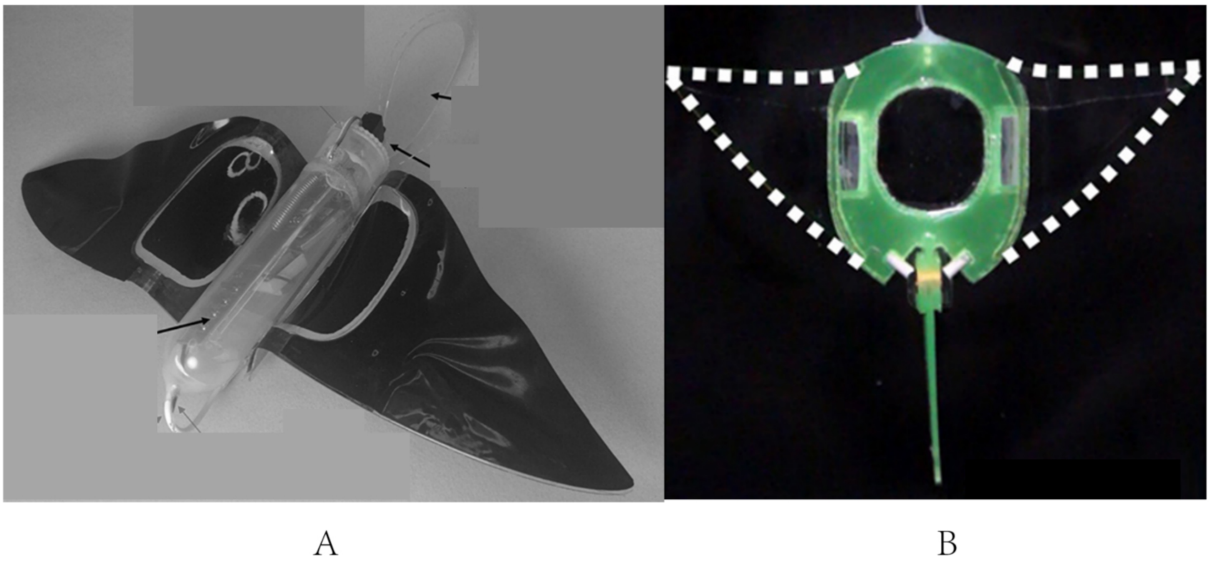


Figure 6. Soft robot fishes in BCF-MPF propulsion mode: (A) a soft robot fish[29]; and (B) a soft electronic fish[30]. BCF: Body and/or caudal fin; MPF: median and/or paired fin.

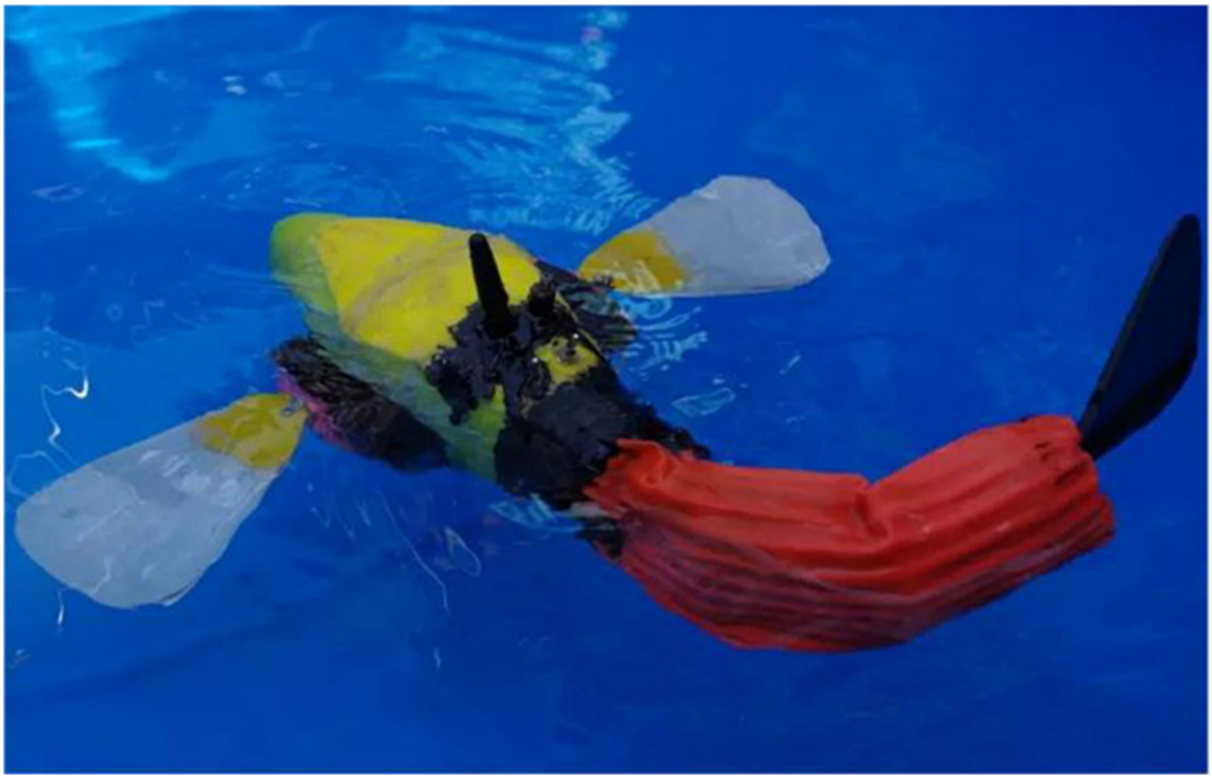


Figure 7. A rigid-soft coupled robot fish in BCF-MPF propulsion mode[31]. BCF: Body and/or caudal fin; MPF: median and/or paired fin.

3. THE MOTION MECHANISM OF ROBOT FISHES

The study of the motion mechanism of robot fishes provides an in-depth understanding of the process by which robot fishes obtain thrust. The results of this study can be utilized to improve the designs of robot fishes. This enables robot fishes to achieve higher propulsion power and efficiency, bridging the swimming performance gap between robot fishes and fish.

Currently, there are three main research methods to study the swimming mechanism of robot fishes. The strengths and weaknesses of the three research methods are summarized in Table 3. The first method is theoretical analysis. In this method, the swimming equations of robot fishes are established by mathematical and physical models. The method is very adaptable, but it is mathematically challenging. Further, the difficulty lies in the need to establish equations that can be solved and correctly describe the complex swimming of robot fishes. The second method is experimental observation. This method uses particle image velocimetry (PIV) or other special equipment to observe robot fishes or fish. The conclusions of the research are highly accurate due to real-world observations, but they have poor universality due to the experimental setting's restrictions. The third method is numerical simulation, which uses computers to numerically solve existing models to predict the swimming characteristics of robot fishes. The method is low cost and accurate, but it cannot solve some complex swimming problems that lack a perfect mathematical model. We can see that each of the three research methods has strengths and weaknesses, and combining these methods can yield complementary benefits.

3.1. Theoretical analysis

The swimming of robot fishes mimics that of fish. A better understanding of the fish motion mechanism aids in the design of robot fishes. There are numerous theories about fish swimming, but only a few widely accepted ones are discussed here. In 1970, Lighthill proposed the "elongated-body theory"[33]. This theory only investigates the role of the fish's caudal cross-section in swimming, ignoring the effect of the caudal vortex. As a result, the swimming performance obtained by this theory is only related to the flow parameters in the cross-section of the fish's caudal. Furthermore, the theory is only applicable to analyzing the swimming of fish with small amplitude. One year later, the "large-amplitude elongated-body theory" was further proposed by Lighthill[34]. In 1991, Tong *et al*. developed the "three-dimensional waving plate theory" based on the "two-dimensional waving plate theory" of Wu[35,36]. This theory simplifies the swimming of a fish to a flexible deformed plate oscillating in a wave-like motion. It is worth noting that the tail vortex effect is considered, which makes the calculation results closer to the real swimming of the fish. This theory is applicable to fish swimming with small amplitude. It can be extended to the accelerated swimming of fish and large-amplitude non-linear swimming.

In recent years, there have been new developments in the theory of robot fishes' swimming. They are mainly a supplement to the previous theories and thus solve some practical problems. Wang *et al.* incorporated the robot fish's head oscillation equation into the kinematic model based on the elongated-body theory[33,37]. The improved kinematic model was established successfully. The results show that the maximum swing angle of the head was reduced to 86% of its original value, while the swimming speed was increased by 17%. Kirchhoff's equations of motion were utilized by Kopman *et al.* to show the dynamics of frontal link[38]. Caudal fin oscillation was modeled by Euler-Bernoulli beam theory. The influence of the fluid around the robot fish was described by the Morison equation. Finally, the dynamic equation of the robot fish propelled by soft fin was established.

3.2. Experimental observation

With the emergence of new experimental equipment, experimental observation has become more popular. PIV is the most effective experimental method. It is a method of measuring flow velocity that involves recording the position of particles in the flow field with multiple cameras and analyzing the images captured. The basic idea is to spread tracer particles in the flow field and then inject a pulsed laser into the measured flow field area. The images of the particles are recorded by two or more consecutive exposures. Zhu *et al*. visualized the flow field by PIV and obtained the flow field image of Tunabot during the caudal

Table 3. Research methods to study the swimming mechanism of robot fishes

Research methods	Strengths	Weaknesses
Theoretical analysis	Very adaptable	Mathematically challenging
Experimental observation	Highly accurate	Poor universality
Numerical simulation	Low cost Accurate	Solving a limited number of problems

fin oscillation[24]. It is frequently necessary to construct special experimental platforms in order to meet the measurement of specific physical quantities. As shown in Figure 8, the robot fish was immersed in a tank and the swimming speed was measured[27].

3.3. Numerical simulation

In recent years, computer technology, computational fluid dynamics (CFD), and other disciplines have advanced rapidly. New iterations of computers have led to a significant increase in computing power, allowing some complex swimming problems to be solved. The calculation model is continuously improved in practice, resulting in increasing accuracy of the calculation. Thus, numerical simulation has made it possible to acquire accurate answers to some complex swimming problems. Currently, many research results are available. The hydrodynamic performance of fish of different shapes near the water surface using CFD was studied by Zhan *et al.*[39]. Using an incompressible Navier-Stokes flow solver based on the immersion boundary method, Liu *et al*. studied the body-fin and fin-fin interactions[40]. Han *et al*. used the same solver as Liu *et al.* to simulate the swimming of the fish on the static cartesian grid[40,41]. The interactions between the intermediate fins were analyzed in detail. The CFD method was used by Macias *et al.* to simulate the swimming process of the fish in undisturbed water flow[42]. Zhu *et al.* combined the immersed boundary-lattice Boltzmann method in numerical simulation with a deep recurrent Q-network to simulate the behavior of fish[43]. It provided an effective method for researching fish adaptation behaviors in complex environments. All of the above swimming problems require a massive amount of computation, which was previously extremely difficult to achieve. From the results of the calculations, all of the authors considered that the accuracy of the calculations met the requirements. We believe that numerical simulation as a method will have considerable potential in the future.

3.4. Multiple research methods

Using multiple research methods to analyze a problem, each research method can not only complement each other's strengths but also verify the results of the others, which increases the convincingness of the research. Korkmaz *et al*. established kinematic and dynamic models of the robot fish using the Denavit-Hartenberg method and Lagrange method, respectively[2]. The swimming of the robot fish was simulated using MATLAB/Simulink. Experiments in the pool validated the simulation results. Behbahani *et al.* established the dynamic model of robot fishes using the rigid body dynamics theory[44]. The hydrodynamic force acting on the pectoral fin was solved by the blade element theory. The kinetic model was evaluated experimentally. The dynamic equation of the fish in autonomous swimming was established by Xin *et al.*[45]. The steering motion of fish was simulated using three-dimensional (3D) CFD software. Liu *et al*. established a kinematic model by simplifying the caudal fin to a rigid hydrofoil and the caudal peduncle to a rigid plate[46]. The caudal fin propulsion mechanism was analyzed using CFD to determine the principle of generating propulsive power. It can be anticipated that this method will be used by more and more researchers and become a new research trend.

4. MOTION COORDINATION AND COMMUNICATION OF MULTIPLE ROBOT FISHES

The research of multiple robot fishes emerged in recent years and is now a hot research field. When

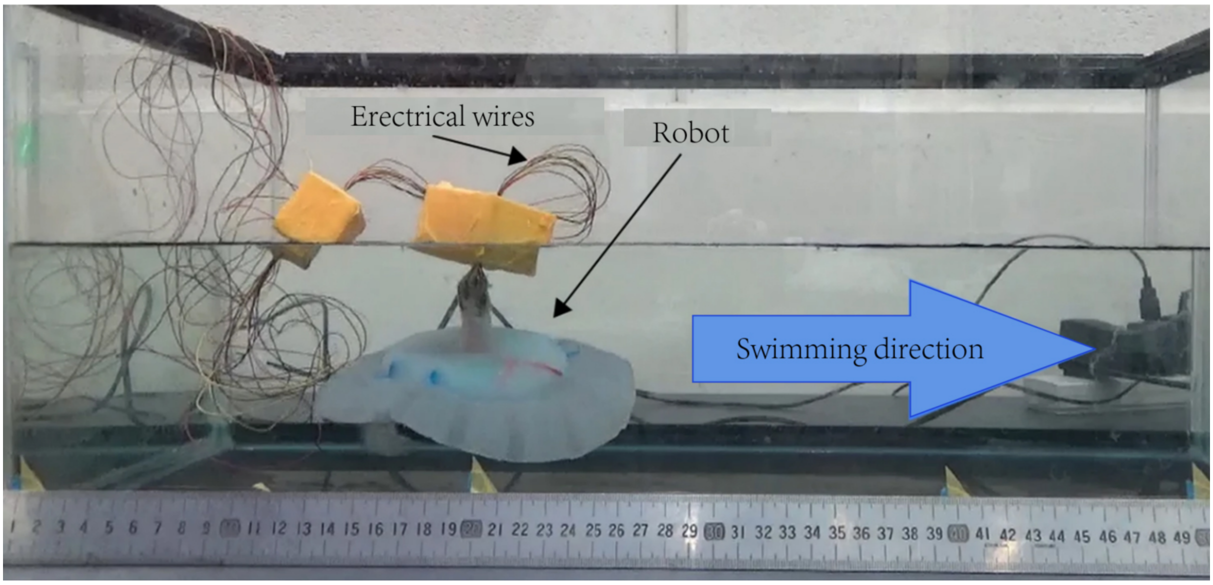


Figure 8. Experimental setup for measuring the swimming speed[27].

discussing the problem of multiple robot fishes, we are most concerned with the problems of motion coordination and communication of multiple robot fishes. As a result, we review the latest research on these two issues in depth.

4.1. Motion coordination of multiple robot fishes

Fish frequently congregate in schools. Fish schools can not only effectively fight against natural enemies but also save energy and help them survive in harsh environments. Researchers believe that schools of multiple robot fishes can reap the same benefits. Therefore, we focus on coordinated swimming of multiple robot fishes and related discussions. The current study is mainly concerned with tandem formation and parallel formation. However, there have been studies on other planar formations.

Tandem formation refers to the connection of the heads and tails of two or more fish in a straight line, as shown in Figure 9. The fish at the front of the line is known as the leading fish, and the fish behind it is known as the following fish. The most basic formation of this is two fish swimming in tandem formation. Tandem swimming of two 3D bionic fish was studied by Wu *et al.*[47]. The results show that, in the absence of any control by the two fish, the vortex generated by the leading fish deflected the path of the following fish. Khalid *et al.* found that the undulating frequency of the following fish does not affect the vortex and time-averaged drag of the leading fish at a certain Strouhal number[48]. Furthermore, it appeared to be more favorable for the leading fish when both fish kept swimming in tandem formation.

Parallel formation refers to two or more fish lining up in a row, as shown in Figure 10. Similarly, the fish at the front of the line is called the leading fish, and the fish behind it is called the following fish. The most basic form of this is two fish swimming in parallel formation. The efficiency of two fish when swimming in parallel was analyzed by Doi *et al.*[49]. The results show that the highest swimming efficiency was achieved when the distance between the two fish (K1) was 0.4 BL under the premise of $L_1 = 0$. A vortex phase matching strategy for robot fishes was found by Li *et al.*[50]. The following robot fish could conserve energy when the front-back distance between two robot fishes was linearly connected to the tailbeat phase difference. As shown in Figure 11, the following robot fish could save energy by vortex phase matching. By

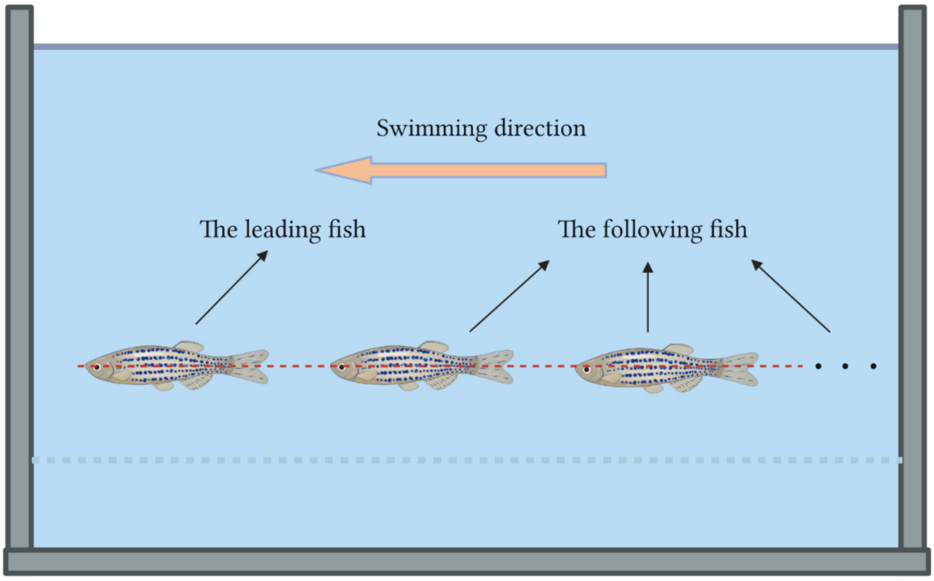


Figure 9. Tandem formation of fish.

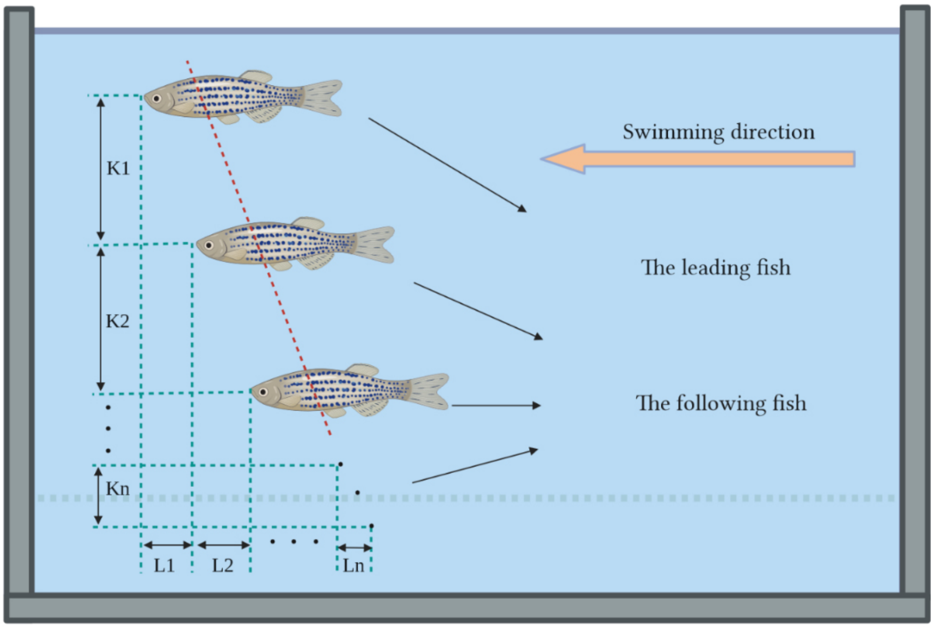


Figure 10. Parallel formation of fish ($Li \geq$ and $Ki \geq 0$, $i = 1, 2, \ldots, n$)

fitting, the phase difference was found to be linearly related to the phase difference, as shown in Figure 11A and B. Subsequent experiments confirmed that fish also exhibit this swimming strategy. Without a complicated vision system and artificial lateral line system (ALLS), this swimming strategy can reduce energy consumption and enhance swimming efficiency. This is quite crucial. The swimming speed and energy consumption of a single robot fish and two parallel robot fishes were investigated by Li *et al.*[51]. It was discovered that, regardless of the tail-beat phase difference, maintaining a parallel formation always increased their swimming speed and decreased their energy consumption. Furthermore, the authors hypothesized that fish can balance their consumption with the benefits they receive from their neighbors by

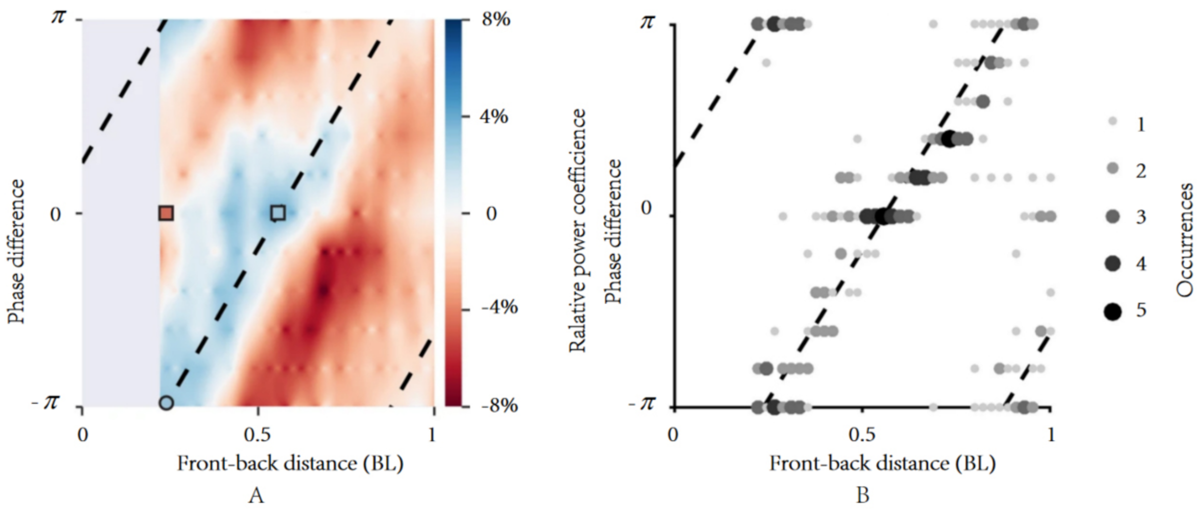


Figure 11. The following robot fish saves energy by vortex phase matching. (A) Relative power coefficient: Positive and negative values, respectively, represent energy saving and energy cost relative to swimming alone. The dashed line represents the function between phase difference and front-back distance, as shown in (B). (B) Location of energy saving: The size and darkness of the dots represent the number of times that the energy saving state occurs[50].

adjusting the tail-beat phase difference as they swim. This suggested that individuals in swimming schools might engage in competitive games.

The discussion of various planar formations aids in determining the best formation. The average swimming efficiency of robot fish formations formed in tandem, square, diamond, and rectangular shapes was investigated by Li *et al.*[52]. It was found that the average swimming efficiency of the tandem formation was highest when the spacing of robot fishes was less than 1.25 BL. The average swimming efficiency of the rectangular formation was highest when the spacing was greater than 1.25 BL. In addition, the wake and pressure generated by the oscillation of the robot fish had an important effect on the Froude efficiency. The wake primarily influenced propulsive force, while pressure primarily influenced the lateral power loss. In this study, the phase difference of each robot fish's oscillation was constant, and the situation when the phase difference changed was not discussed.

The 3D formation is closer to a natural school of fish, and therefore it has more practical application. 3D is mainly reflected by having the height difference as a variable. The energy consumption of two robot fishes when they formed a 3D formation was studied by Li *et al.*[53]. The results show that the following robot fish could save energy consumption when there was a linear relationship between the height difference and phase difference of the two robot fishes. This research result is significant because it provided ideas for the future 3D formation of robot fishes.

4.2. Communication of multiple robot fishes

When multiple robot fishes form a formation, they must communicate with others in order to maintain the formation and avoid a collision. Since the distance between each robot fish is short, this is a problem for underwater close communication. Relevant studies have been conducted to date, and some solutions have been proposed. Among them, Xie Guangming's team from Peking University conducted extensive research and produced impressive results.

A proper electronic communication system facilitates the communication of multiple robot fishes. Since the robot fish's electronic communication system frequently uses the same channel, collisions always occur during communication. To solve this problem, based on carrier sense multiple access with collision avoidance (CSMA/CA), an electronic communication system was proposed by Zhang *et al*.[54]. This system incorporated a communication channel detection circuit and employed a CSMA/CA-based protocol. The simulation and experimental results validate the system's effectiveness. Nevertheless, this communication system suffered from effective bandwidth loss.

Fish can perceive information from the surrounding fluid using the lateral line system (LLS)[55]. This has serious implications for their underwater survival. Inspired by the excellent performance of the fish's LLS, the artificial lateral line system (ALLS) was designed and applied to the robot fish. Predictably, ALLS plays an important role in improving the interaction and collaboration capabilities between adjacent robot fishes. Zheng *et al*. established ALLS by composing an array of pressure sensors[56]. This ALLS could detect vortex streets generated by adjacent robot fish. According to the experimental results, it allowed the robot fish to perceive the relative vertical distance and yaw/pitch/roll angle with the adjacent robot fish. Furthermore, the oscillation amplitude/frequency/offset of the adjacent robot fish could also be sensed. However, the study was limited to the perception of the outside world by only one robot fish applying ALLS. Therefore, Zheng *et al*. further investigated ALLS on the perception of longitudinal separation sensing of two robot fishes[57]. Longitudinal separation implies that the two robot fishes maintain constant lateral spacing, change the longitudinal spacing, and keep the robot fish within the influence of the vortex produced by another robot fish. The meaning of longitudinal spacing and lateral spacing is clearly shown in Figure 12. The authors experimentally obtained a qualitative relationship between the longitudinal separation of two robot fishes and the ALLS-measured hydrodynamic pressure variations. The effectiveness of ALLS in relative state awareness applications was also verified. Unfortunately, the study was limited to qualitative analysis, with no quantitative analysis.

Using vision for communication is the most straightforward method. Berlinger *et al*. devised a new method of communication in schools of robot fishes that was inspired by the fact that fish could use vision to coordinate their motions[58]. The vision system of the robot fish was comprised of two cameras and LEDs. Through the algorithm, the robot fish could quickly determine the location of the adjacent robot fish after recognizing the light. The experimental results demonstrate that the robot fish could perform a variety of school behaviors using visual information. However, it is unclear whether the communication technology is still effective in environments that may hinder vision, such as murky waters.

We find that communication can be established between the robot fish and the fish school. When the robot fish swims in the water, it attracts the fish school to move closer to it. Eventually, the robot fish becomes the leader, leading the whole school of fish to swim forward, as shown in Figure 13. It is worth noting that the robot fish does not have smell, sound, or light to attract the fish. We hypothesize that the tail vortices created by the robot fish when it swims are the cause of this phenomenon. The swimming performance of robot fish is much inferior to that of fish. One of the reasons for this is that the tail vortices are not fully utilized. As is known, creatures have always tended to be profit-oriented. When fish perceive tail vortices, they tend to take advantage of them. In turn, it follows the robot fish, which eventually leads to this phenomenon. This brings the robot fish into communication with the fish school. We are confident that finding out how to exploit this communication would be meaningful research.

5. CHALLENGES AND FUTURE WORKS

Thanks to a lot of research on bionic robot fishes in recent years, significant progress has been made.

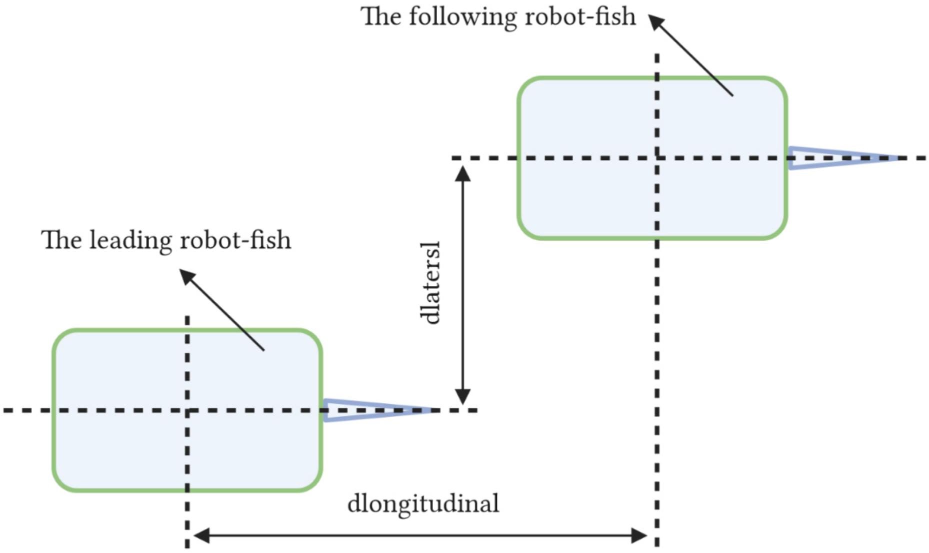


Figure 12. The meaning of longitudinal spacing and lateral spacing.

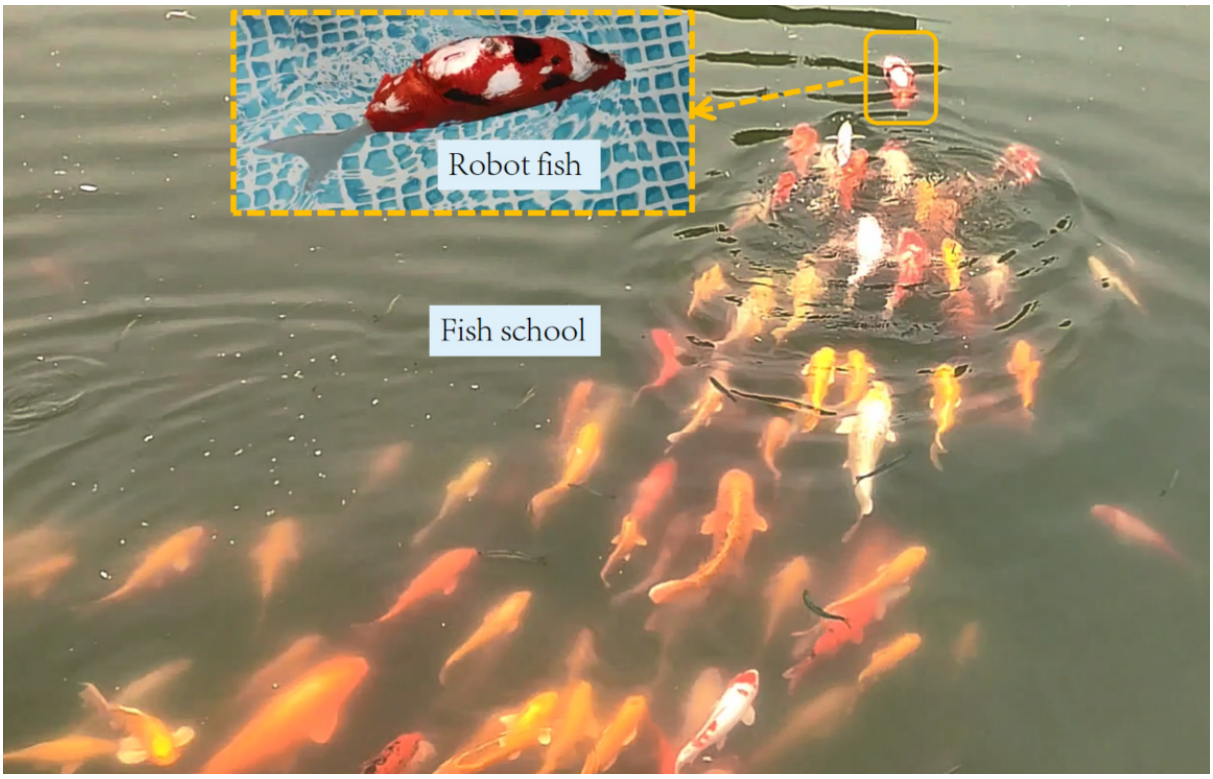


Figure 13. The robot fish and the school of fish that follows it.

However, there are numerous challenges that need further work.

- Robot fishes are far inferior to fish in terms of swimming performance. Table 4 displays the performance parameters of typical robot fishes over the last five years. The maximum swimming speed of robot fishes in Table 4 is currently only 4 BL·s^{-1}, whereas a fish can easily reach 8 BL·s^{-1} with regular swimming[59]. This

Table 4. Typical robot fishes and their performance parameters

Reference	Maximum swimming speed		Minimum turning radius (m)	Frequency (hertz)		Swimming type	Structural type
	$BL \cdot s^{-1}$	$m \cdot s^{-1}$		Caudal fin	Pectoral fins		
Ref. [19]	0.5(Ave)	0.23(Ave)	0.78(Ave)	NA	---	BCF	Soft
Ref. [20]	0.25	0.037	NA	0.75	---	BCF	Soft
Ref. [24]	4	1.02	NA	15	---	BCF	Rigid-soft
Ref. [22]	2.5	0.25	NA	NA	---	BCF	Rigid-soft
Ref. [25]	0.87	0.31	NA	2.9	---	BCF	Rigid-soft
Ref. [26]	0.45	0.052	NA	---	1	MPF	Soft
Ref. [28]	0.94	0.43	≈0	---	NA	MPF	Rigid-soft
Ref. [27]	NA	0.013	NA	---	4	MPF	Rigid-soft
Ref. [30]	0.69	0.064	0.085	NA	NA	BCF-MPF	Soft
Ref. [29]	NA	0.062	0.234	NA	NA	BCF-MPF	Soft
Ref. [32]	0.66	0.365	0.139	NA	NA	BCF-MPF	Rigid-soft

Frequency (hertz) indicates the value at the maximum (or Ave) swimming speed. The ranking of the references is based on the magnitude of the maximum swimming speed ($BL \cdot s^{-1}$) of the robot fish and is classified by swimming type and structural type. NA: Not available; Ave: average; BCF: body and/or caudal fin; MPF: median and/or paired fin.

demonstrates the gap in swimming performance between robot fishes and fish, which is an urgent problem to be solved. We believe there are several approaches to solve this problem. The first approach is to investigate the effect of the vortices on the swimming efficiency of robot fishes. We believe that the high propulsion efficiency of fish is closely related to the vortices they generate when they swim. It is possible to improve the swimming efficiency of robot fishes by measuring the vortices generated when fish swim and reproducing them in robot fishes. The second approach is to narrow the gap between the drive systems of robot fishes and the muscles and skin of fish. Robot fishes simulate the swimming of fish by using multiple rigid connecting rods. Fish have a flexible body made up of muscles and skin that allows them to swim continuously and supplely. However, due to the rigidity of the connecting rod and the limitation of the number of rods, the motion of robot fishes exhibits a discrete and unnatural movement. Attempts can be made to flex the connecting rod to achieve continuous motion of robot fishes, thus improving maneuverability. The third approach is to further reduce the water resistance of robot fishes when swimming. Water resistance is currently decreased mostly by designing the shape of the robot fish to be streamlined. Fish, on the other hand, have fish scales and mucous on their bodies, which can considerably reduce resistance. However, the relevant design is rarely observed in the current robot fishes. The fourth approach is to conduct an in-depth investigation of robot fishes in the BCF-MPF propulsion mode. Robot fishes in BCF propulsion mode swim fast but have poor maneuverability. In contrast, robot fishes in MPF propulsion mode have great maneuverability but slow swimming speed. The BCF-MPF propulsion mode combines the above two propulsion modes, which can accurately imitate the swimming of fish. With a reasonable design, it can achieve high swimming speed and great maneuverability and has wider application prospects. This is a promising research direction. The final approach is to use sensor technology to create close connections between robot fishes and fish. Replicating the swimming process of fish can improve the swimming performance of robot fishes. Through the sensors, we obtain real-time feedback data (body deformation, etc.) when fish swim, further completing the monitoring of the entire swimming process. Finally, the collected data are applied to robot fishes. This allows robot fishes to make rhythmic movements similar to fish, improving their swimming performance.

- The majority of studies have only used one research method to investigate the swimming mechanism of robot fishes. Actually, each research method has its own strengths and weaknesses. Because of the weaknesses, using only one method may provide unconvincing results. The combined use of multiple research methods not only achieves the complementary benefits of each method but also allows each method to verify the others to ensure the accuracy of the results.

- There is a lack of sufficient research on motion coordination and communication of multiple robot fishes. Multiple robot fishes in an appropriate formation have been shown to reduce energy consumption[47-53]. In nature, the number of fish in a school is usually greater than three, and the school is in a three-dimensional formation. However, the current study has limitations in terms of the number and formation of robot fishes. Specifically, the number of robot fishes is generally two, and the formation of robot fishes is mostly flat. The research of three or more robot fishes and the research of three-dimensional formation of robot fishes will be future research trends. The communication of multiple robot fishes is an intriguing research topic. Robot fishes need to communicate with each other to form formations, thus reducing energy consumption. The research on communication among multiple robot fishes has only recently received adequate attention. The related technology is not yet fully mature and should be tested in the actual environment. In addition, the research on communication between robot fish and fish schools is interesting content. We can imagine a future where schools of robot fishes swim together with schools of fish to form a larger school, achieving energy savings as well as harmony between robot fishes and fish.

6. CONCLUSION

This paper provides a comprehensive review of recent advances in several important fields of bionic robot fishes. The latest achievements in the development of robot fishes are presented. Based on the discussion of the main swimming theories of fish, the latest progress in the study of the swimming mechanism is summarized. The current state of research in the new field of motion coordination and communication of multiple robot fishes is analyzed.

Based on the survey, the data show that robot fishes in BCF propulsion mode can obtain high propulsion speed. This reflects the speed advantage of BCF propulsion mode. Robot fishes in MPF propulsion mode realize a small radius or even in situ turning. The turning radius is an important indicator of maneuverability. This reflects the high maneuverability of the MPF propulsion mode. The maneuverability of robot fishes in BCF-MPF propulsion mode is improved compared to robot fishes in BCF propulsion mode. However, in terms of swimming speed, compared with robot fishes in MPF propulsion mode, they fail to demonstrate the expected superiority. As a result, robot fishes in this propulsion mode have more room for advancement. The high-frequency oscillation of the caudal fin can significantly increase the propulsive speed. The soft robot fish has a low speed of propulsion compared with the rigid-soft coupled robot fish.

This paper primarily summarizes research results on robot fishes from the last five years, and the reader should be aware of the paper’s time constraints. In the future, we will make efforts to improve the swimming performance of robot fishes and continue to track new advances in the research of robot fishes.

DECLARATIONS

Authors’ contributions

Made substantial contributions to the research and investigation process, reviewed and summarized the literature, wrote and edited the original draft: Li K, Ji Y

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

Performed critical review, commentary and revision, as well as provided technical guidance: Yang S

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by the National Key Research and Development Program of China (Grant No. 2019YFB1309600), the National Natural Science Foundation of China (Grant Nos. 52075248, 51475230).

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.

Copyright

REFERENCES

1. Triantafyllou MS, Triantafyllou GS. An efficient swimming machine. *Sci Am* 1995;272:64-70. DOI
2. Korkmaz D, Akpolat ZH, Soygüder S, Alli H. Dynamic simulation model of a biomimetic robotic fish with multi-joint propulsion mechanism. *Transactions of the Institute of Measurement and Control* 2015;37:684-95. DOI
3. Videler JJ. Fish swimming. 1st ed. London: Chapman and Hall Ltd; 1993. p. 1-226
4. Lauder GV, Anderson EJ, Tangorra J, Madden PG. Fish biorobotics: kinematics and hydrodynamics of self-propulsion. *J Exp Biol* 2007;210:2767-80. DOI PubMed
5. Fish FE. Advantages of natural propulsive systems. *Mar Technol Soc J* 2013;47:37-44. DOI
6. Lauder GV. Fish locomotion: recent advances and new directions. *Ann Rev Mar Sci* 2015;7:521-45. DOI PubMed
7. Sfakiotakis M, Lane D, Davies J. Review of fish swimming modes for aquatic locomotion. *IEEE J Oceanic Eng* 1999;24:237-52. DOI
8. Breder CM. The locomotion of fishes. *Zoologica* 1926;4:159-297.
9. Wang A. Development and analysis of body and/or caudal fin biomimetic robot fish. *J Mech Eng* 2016;52:137. DOI
10. Cai Y. Research advances of bionic fish propelled by oscillating paired pectoral foils. *JME* 2011;47:30. DOI
11. Colgate J, Lynch K. Mechanics and control of swimming: a review. *IEEE J Oceanic Eng* 2004;29:660-73. DOI
12. Yu J, Wen L, Ren Z. A survey on fabrication, control, and hydrodynamic function of biomimetic robotic fish. *Sci China Technol Sci* 2017;60:1365-80. DOI
13. Scaradozzi D, Palmieri G, Costa D, Pinelli A. BCF swimming locomotion for autonomous underwater robots: a review and a novel solution to improve control and efficiency. *Ocean Engineering* 2017;130:437-53. DOI
14. Yu J, Wang M, Dong H, Zhang Y, Wu Z. Motion control and motion coordination of bionic robotic fish: a review. *J Bionic Eng* 2018;15:579-98. DOI
15. Chu W, Lee K, Song S, et al. Review of biomimetic underwater robots using smart actuators. *Int J Precis Eng Manuf* 2012;13:1281-92. DOI
16. Liu H, Tang Y, Zhu Q, Xie G. Present research situations and future prospects on biomimetic robot fish. *Inter J Smart Sens Intell Syst* 2022;7:458-80. DOI
17. Liu G, Wang A, Wang X, Liu P. A review of artificial lateral line in sensor fabrication and bionic applications for robot fish. *Appl Bionics Biomech* 2016;2016:4732703. DOI PubMed PMC
18. Raj A, Thakur A. Fish-inspired robots: design, sensing, actuation, and autonomy--a review of research. *Bioinspir Biomim* 2016;11:031001. DOI
19. Katzschmann RK, DelPreto J, MacCurdy R, Rus D. Exploration of underwater life with an acoustically controlled soft robotic fish. *Sci Robot* 2018;3:eaar3449. DOI PubMed
20. Shintake J, Cacucciolo V, Shea H, Floreano D. Soft biomimetic fish robot made of dielectric elastomer actuators. *Soft Robot* 2018;5:466-74. DOI PubMed PMC
21. Liu S, Wang Y, Li Z, Jin M, Ren L, Liu C. A fluid-driven soft robotic fish inspired by fish muscle architecture. *Bioinspir Biomim*

2022;17:026009. DOI PubMed
22. Liao P, Zhang S, Sun D. A dual caudal-fin miniature robotic fish with an integrated oscillation and jet propulsive mechanism. *Bioinspir Biomim* 2018;13:036007. DOI PubMed
23. Coral W, Rossi C, Curet OM, Castro D. Design and assessment of a flexible fish robot actuated by shape memory alloys. *Bioinspir Biomim* 2018;13:056009. DOI PubMed
24. Zhu J, White C, Wainwright DK, Di Santo V, Lauder GV, Bart-Smith H. Tuna robotics: a high-frequency experimental platform exploring the performance space of swimming fishes. *Sci Robot* 2019;4:eaax4615. DOI PubMed
25. Chen B, Jiang H. Body stiffness variation of a tensegrity robotic fish using antagonistic stiffness in a kinematically singular configuration. *IEEE Trans Robot* 2021;37:1712-27. DOI
26. Li G, Chen X, Zhou F, et al. Self-powered soft robot in the mariana trench. *Nature* 2021;591:66-71. DOI PubMed
27. Yurugi M, Shimanokami M, Nagai T, Shintake J, Ikemoto Y. Cartilage structure increases swimming efficiency of underwater robots. *Sci Rep* 2021;11:11288. DOI PubMed PMC
28. Ma H, Cai Y, Wang Y, Bi S, Gong Z. A biomimetic cownose ray robot fish with oscillating and chordwise twisting flexible pectoral fins. *IR* 2015;42:214-21. DOI
29. Zhang Z, Yang T, Zhang T, et al. Global vision-based formation control of soft robotic fish swarm. *Soft Robot* 2021;8:310-8. DOI PubMed
30. Li T, Li G, Liang Y, et al. Fast-moving soft electronic fish. *Sci Adv* 2017;3:e1602045. DOI PubMed PMC
31. Li Z, Ge L, Xu W, Du Y. Turning characteristics of biomimetic robotic fish driven by two degrees of freedom of pectoral fins and flexible body/caudal fin. *InterJ Adv Rob Syst* 2018;15:172988141774995. DOI
32. Zhong Y, Li Z, Du R. Robot fish with two-DOF pectoral fins and a wire-driven caudal fin. *Advanced Robotics* 2018;32:25-36. DOI
33. Lighthill MJ. Aquatic animal propulsion of high hydromechanical efficiency. *J Fluid Mech* 1970;44:265. DOI
34. Lighthill MJ. Large-amplitude elongated-body theory of fish locomotion. *Proc R Soc Lond B* 1971;179:125-38. DOI
35. Tong BG, Zhuang LX. Hydrodynamic Model for Fish's Undulatory Motion and Its Applications. *Chin J Nat* 1998;1:1-7.
36. Wu TY. Swimming of a waving plate. *J Fluid Mech* 1961;10:321-44. DOI
37. Wang P, Xu BZ, Lou BD, et al. Optimization and experimentation on the kinematic model of bionic robotic fish. *CAAI Trans Intell Syst* 2017;12:196-201. DOI
38. Kopman V, Laut J, Acquaviva F, Rizzo A, Porfiri M. Dynamic modeling of a robotic fish propelled by a compliant tail. *IEEE J Oceanic Eng* 2015;40:209-21. DOI
39. Zhan JM, Gong YJ, Li TZ. Gliding locomotion of manta rays, killer whales and swordfish near the water surface. *Sci Rep* 2017;7:406. DOI PubMed PMC
40. Liu G, Ren Y, Dong H, Akanyeti O, Liao JC, Lauder GV. Computational analysis of vortex dynamics and performance enhancement due to body-fin and fin-fin interactions in fish-like locomotion. *J Fluid Mech* 2017;829:65-88. DOI
41. Han P, Lauder GV, Dong HB. Hydrodynamics of median-fin interactions in fish-like locomotion: effects of fin shape and movement. *Physics Fluids* 2020;32:011902. DOI
42. Macias MM, Souza IF, Brasil Junior AC, Oliveira TF. Three-dimensional viscous wake flow in fish swimming - A CFD study. *Mechanics Research Communications* 2020;107:103547. DOI
43. Zhu Y, Tian FB, Young J, Liao JC, Lai JCS. A numerical study of fish adaption behaviors in complex environments with a deep reinforcement learning and immersed boundary-lattice Boltzmann method. *Sci Rep* 2021;11:1691. DOI PubMed PMC
44. Behbahani SB, Tan X. Design and modeling of flexible passive rowing joint for robotic fish pectoral fins. *IEEE Trans Robot* 2016;32:1119-32. DOI
45. Xin Z, Wu C. Vorticity dynamics and control of the turning locomotion of 3D bionic fish. *SAGE* 2018;232:2524-35. DOI
46. Liu G, Liu S, Xie Y, Leng D, Li G. The analysis of biomimetic caudal fin propulsion mechanism with CFD. *Appl Bionics Biomech* 2020;2020:7839049. DOI PubMed PMC
47. Wu C, Wang L. Numerical simulations of self-propelled swimming of 3D bionic fish school. *Sci China Ser E-Technol Sci* 2009;52:658-69. DOI
48. Khalid MSU, Akhtar I, Dong H. Hydrodynamics of a tandem fish school with asynchronous undulation of individuals. *Journal of Fluids and Structures* 2016;66:19-35. DOI
49. Doi K, Takagi T, Mitsunaga Y, Torisawa S. Hydrodynamical effect of parallelly swimming fish using computational fluid dynamics method. *PLoS One* 2021;16:e0250837. DOI PubMed PMC
50. Li L, Nagy M, Graving JM, Bak-Coleman J, Xie G, Couzin ID. Vortex phase matching as a strategy for schooling in robots and in fish. *Nat Commun* 2020;11:5408. DOI PubMed PMC
51. Li L, Ravi S, Xie G, Couzin ID. Using a robotic platform to study the influence of relative tailbeat phase on the energetic costs of side-by-side swimming in fish. *Proc Math Phys Eng Sci* 2021;477:20200810. DOI PubMed PMC
52. Li S, Li C, Xu L, Yang W, Chen X. Numerical simulation and analysis of fish-like robots swarm. *Applied Sciences* 2019;9:1652. DOI
53. Li L, Zheng X, Mao R, Xie G. Energy saving of schooling robotic fish in three-dimensional formations. *IEEE Robot Autom Lett* 2021;6:1694-9. DOI
54. Zhang H, Wang W, Zhou Y, et al. CSMA/CA-based electrocommunication system design for underwater robot groups. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2017:2415-20. DOI
55. Zhai Y, Zheng X, Xie G. Fish lateral line inspired flow sensors and flow-aided control: a review. *J Bionic Eng* 2021;18:264-91. DOI

56. Zheng X, Wang C, Fan R, Xie G. Artificial lateral line based local sensing between two adjacent robotic fish. *Bioinspir Biomim* 2017;13:016002. DOI PubMed
57. Zheng XW, Wang MY, Zheng JZ, et al. Artificial lateral line based longitudinal separation sensing for two swimming robotic fish with leader-follower formation. *IEEE/RSJ International Conference on Intelligent Robots and Systems* 2019:2539-44. DOI
58. Berlinger F, Gauci M, Nagpal R. Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm. *Sci Robot* 2021;6:eabd8668. DOI PubMed
59. Spierts IL, Leeuwen JL. Kinematics and muscle dynamics of C- and S-starts of carp (Cyprinus carpio L.). *J Exp Biol* 1999;202:393-406. DOI PubMed

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) 2022." 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

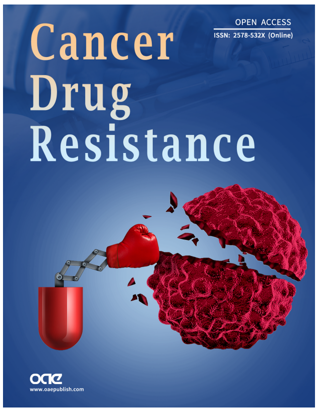
Intelligence & Robotics Editorial Office
Suite 1504, Plaza A, Xi’an National Digital Publishing Base,
No. 996 Tiangu 7th Road, Gaoxin District, Xi’an 710077, Shaanxi, China.

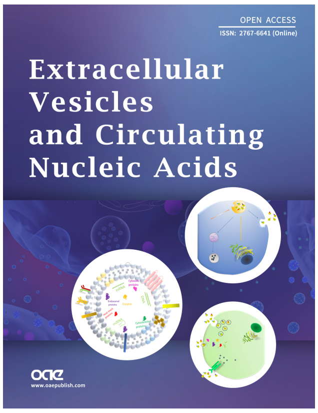
Managing Editor

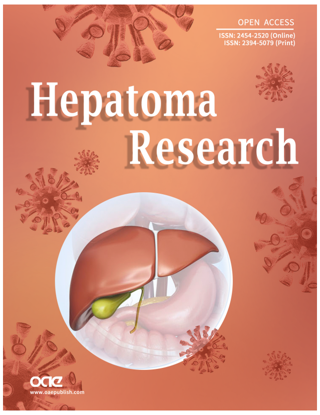
Lijun Jin
Email: editorial@intellrobot.com

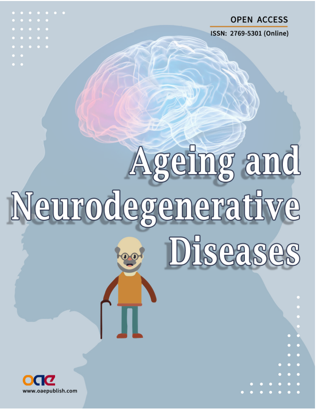
OAE Publishing Inc. (https:oaepublish.com) is a multidisciplinary open-access publishing company, founded in Los Angeles in 2015. Until now, OAE has been recognized by authoritative organizations in publishing industries,such as the *ORCID, COPE*, Scientific, Technical and Medical Publishers (STM), Crossref,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

As of June 2022, 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 June 2022, 3,154 articles have been published online, with 10,944,568 hits and 2,285,864 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