

# Diffusion-based Multi-agent Reinforcement Learning for Semantic Vehicular Edge Computing

Yi Yang, *Student Member, IEEE*, Wenqiang Ma, *Student Member, IEEE*, Wen Sun, *Senior Member, IEEE*, Jianhua He, *Senior Member, IEEE*, Yaru Fu, *Member, IEEE*, Chau Yuen, *Fellow, IEEE*, and Yan Zhang, *Fellow, IEEE*

**Abstract**—Vehicular edge computing (VEC) is critical for the safe and efficient driving of intelligent vehicles, by which they can offload computation-intensive tasks (such as driving environment perception) to edge servers to overcome the limitations of onboard computational resources and cooperate with others. One of the major challenges faced by VEC is that the offloaded intelligent driving tasks generally generate large amounts of data, which can easily stretch and congest the vehicle communication channels. To address the above challenges, we first propose a novel semantic VEC (SVEC) architecture, which can extract the semantic information of tasks and offload them to edge servers, thereby achieving reliable and efficient offloaded task communication and computation adaptively. Considering the scarce channel resources of vehicles and the intelligent tasks with different priorities and modalities, we define a novel user utility model for SVEC and transform the problem of maximizing user utility into a joint optimization problem of semantic feature extraction, task offloading and resource allocation. Furthermore, to cope with the complexity of the solution space of the optimization problem, we propose a diffusion-based multi-agent reinforcement learning algorithm, which improves the ability of agents to explore the solution space through the diffusion process, thereby achieving optimal decisions for semantic feature extraction, task offloading and resource allocation. Simulation results show that the proposed scheme improves the overall performance of SVEC while reducing offload latency and average system cost.

**Index Terms**—Vehicular edge computing, semantic communication, task offloading, resource allocation, diffusion model, deep reinforcement learning.

## I. INTRODUCTION

WITH advances of autonomous driving and driving demand of computation-intensive intelligent vehicle applications, Vehicular Edge Computing (VEC) is emerging as a transformative distributed paradigm that effectively reduces the computational and communication burden of vehicles by offloading tasks to proximal edge servers [1]. By avoiding the propagation delay caused by transmitting data from intelligent

vehicles with various levels of driving automation to cloud data centers, VEC can effectively improve the performance of compute-intensive and delay-sensitive intelligent vehicle applications [2], [3]. In addition, VEC can also promote more vehicles to participate in intelligent tasks, as multiple resource-constrained vehicles can collaboratively process environmental and driving data, thereby supporting intelligent vehicle applications such as intelligent warning, autonomous driving, and real-time environmental perception [4].

However, the rapid growth in the number of intelligent vehicles has intensified the conflict between the stringent requirements of intelligent vehicle applications and the limited resources of communication and edge infrastructure [5]. The traditional VEC communication paradigm relies on the continuous transmission of raw sensor data over resource-constrained wireless channels. This results in excessive traffic load, severe network congestion, and inefficient use of edge computing resources, which collectively fail to satisfy the low-latency, high-reliability, and privacy-sensitive requirements of safety-critical vehicular applications. To address the above research problems, in this paper, we first propose a semantic VEC (SVEC) framework for reliable and efficient communication and computing of the offloaded tasks. The proposed SVEC architecture has two major building blocks, semantic vehicle communication and context aware task oriented intelligent offloading. Semantic communication refers to a communication paradigm that goes beyond the traditional approach of simply transmitting raw sensor data. It focuses on analyzing and extracting the key semantic features from multimodal data, such as images, text, and sensor readings, for efficient representation and communication over the vehicle channels [6]. By leveraging large deep learning models, semantic communication enables the transmission of essential semantic features rather than the entire raw data, which helps to alleviate the scarcity of spectrum resources while maintaining the fidelity of vehicle perception data [7]. A new variable of compression factor is proposed to facilitate a trade-off between the communication efficiency and application performance. It is noted that while semantic communication has been widely studied for bandwidth-constrained scenarios and shows great potential, such as in satellite communications [8]–[12], there has been little investigation of semantic vehicle communication dealing with multimodal data and challenging safety-critical intelligent vehicle applications. In realistic VEC networks, vehicle tasks often present multimodal characteristics, and wrong decisions caused by semantic errors may lead to fatal consequences.

Y. Yang, W. Ma and W. Sun are with the School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: yiyang@nwpu.edu.cn; mawenqiang520@mail.nwpu.edu.cn; sunwen@nwpu.edu.cn). (Corresponding author: Wenqiang Ma)

J. He is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK (e-mail: j.he@essex.ac.uk).

Y. Fu is with the School of Science and Technology, Hong Kong Metropolitan University, Hong Kong 999077, China (e-mail: yfu@hkmu.edu.hk).

C. Yuen is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: chau.yuen@ntu.edu.sg).

Y. Zhang is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway (e-mail: yanzhang@ieee.org).

The second major semantic building block of the SVEC framework is context aware task oriented intelligent offloading, tasked to adaptively and efficiently prioritize data which are important for communication and allocate computing resources under changing system contexts and constraints. While SVEC holds great potential for intelligent vehicle applications, it also introduces new challenges arising from the diversity of intelligent tasks with different priorities and data modalities, and complex decisions to be made on semantic extraction, computing offloading, and resource allocation. In addition, the additional computational overhead caused by the semantic extraction and reconstruction process, as well as the trade-off between the extraction, transmission, and offloading latency of semantic information and energy efficiency, needs to be considered [13]. To address these problems, we formulate an optimization problem for the communication and intelligent offloading tasks with a new system utility model. As the solution space of the optimization problem is a mixture of discrete variables and continuous variables, we develop a diffusion-based multi-agent deep reinforcement learning scheme for the intelligent vehicles to cooperate and compete under SVEC. The diffusion-based deep reinforcement learning model is exploited to explore the environment through the diffusion process and efficiently determines the optimal semantic feature extraction and task offloading strategies. While deep reinforcement learning has been widely studied for collaborative task offloading [14]–[19], existing solutions still face challenges such as low exploration efficiency and insufficient policy generalization when dealing with multi-agent policy coupling VEC scenarios. It is worth further investigating how to enhance the agent’s exploration of the complex solution space and achieve optimal decisions for semantic extraction, task offloading, and resource allocation.

The major contributions of this paper can be summarized as follows.

- We propose a novel SVEC architecture for semantic feature extraction and task offloading. The proposed architecture enables connected vehicles to quickly extract semantic information from different tasks, significantly reducing the amount of offloaded data while retaining the core information of tasks. Meanwhile, by leveraging the real-time perception of local and environmental status, the connected vehicles in the SVEC can adaptively determine the offloading strategy, further reducing the computing pressure of the vehicle and the overhead of transmitting semantic information during task offloading.
- To further improve the performance of SVEC, we design a novel diffusion-based multi-agent reinforcement learning scheme to solve the optimization problem for semantic communication and computing. By leveraging the generative and exploratory capabilities of the diffusion process, connected vehicles can better model the uncertainty in state transitions and generate diverse actions, thereby achieving optimal decisions for semantic feature extraction and task offloading. Moreover, the proposed scheme can efficiently manage and allocate resources, ultimately achieving the best balance between semantic

fidelity, system latency, and resource consumption.

- We conduct extensive simulations to verify the reliability of the proposed scheme. Considering the multimodal characteristics of intelligent tasks, we construct a multimodal SVEC system based on Transformer and convolutional neural networks (CNN), which can extract and encode semantic features from text, image, and speech data. Simulation results show that the proposed scheme improves the overall performance of SVEC while reducing offload latency and average system cost. In addition, the proposed scheme outperforms the benchmark with an overall reward of 1.21x higher, showcasing its superior performance in resource-constrained SVEC.

The structure of the paper is as follows: Section II discusses related work, while Section III presents the system model and defines the optimization problem. In Section IV, we introduce our diffusion-based multi-agent deep reinforcement learning algorithm. Section V provides a detailed performance evaluation, and Section VI concludes the paper.

## II. RELATED WORKS

### A. Semantic Communication

In recent years, semantic communication has attracted extensive attention from academia and industry. Semantic communication aims to improve communication efficiency and reduce bandwidth requirements by extracting and transmitting the semantic information of data rather than the raw data. Xie *et al.* [20] proposed a deep learning-based semantic communication scheme that can effectively extract semantic information from text and outperforms traditional communication methods under different signal-to-noise ratio (SNR) environments. Yang *et al.* [21] studied how to enhance edge intelligence through semantic communication and emphasized the potential of semantic communication to achieve real-time intelligence on edge devices. Qing *et al.* [22] proposed a semantic communication architecture supporting cloud-edge-device computing to achieve distributed and collaborative semantic services. However, in real scenarios, the data generated by connected vehicles usually contains multimodality, including images, lidar, GPS, and text or numerical sensor data [23]. Although semantic communication has been explored in various contexts, most existing approaches focus on single-modality inputs and fixed communication environments. Furthermore, the computational and storage capabilities of individual vehicles are typically constrained, posing new challenges for adaptive allocation of limited resources during semantic extraction and task offloading.

### B. Semantic-aware Edge Computing System

Many works have explored integrating semantic communication into VEC to enhance task offloading and resource allocation efficiency. Wang *et al.* [24] proposed an adaptive semantic resource allocation paradigm based on semantic bit quantization, which can dynamically provide resource allocation strategies for users according to the perceived semantic tasks and channel characteristics. Zheng *et al.* [25] studied the dynamic resource allocation problem of semantic extraction

tasks in edge networks, achieving a flexible trade-off between system benefits and costs while ensuring the performance of semantic extraction tasks. Cang et al. [26] introduced a semantic-aware framework for joint communication and computing resource allocation in edge computing networks. This work jointly optimizes semantic-aware partitioning factors and resource management to minimize system energy consumption under long-term latency and processing rate constraints. In SVEC, the challenges become more complex due to the highly dynamic nature of communication conditions, varying levels of semantic relevance across tasks, and the heterogeneity of vehicles in terms of communication resources, task preferences, and semantic processing capabilities [27]. Existing schemes do not fully consider the impact of vehicle heterogeneity and dynamic changes in the SVEC environment on semantic extraction, task offloading, and resource allocation strategies, which hinder their performance in safety-critical and latency-sensitive applications.

### C. Multi-agent Reinforcement Learning for Decision Making

Multi-agent reinforcement learning, as a collaborative optimization technology, is suitable for joint offloading decisions and resource management of different connected vehicles in SVEC [28]. Hoa et al. [29] used the advantage actor-critic and proximal policy optimization methods to optimize the computing resource allocation and task offloading in the semantic communication system, and effectively reduced the latency of the entire semantic communication system. Hoa et al. [29] utilized the advantage actor-critic and proximal policy optimization methods to optimize the computing resource allocation and task offloading strategies in semantic communication, effectively reducing the latency of semantic communication services. Shao et al. [30] proposed a soft actor-critic (SAC) based spectrum decision optimization method in semantic communication, which improved the performance of vehicle semantic communication and alleviated the problems of spectrum scarcity and network traffic. However, SAC algorithms often rely on single-agent reinforcement learning frameworks, which fall short in capturing the collaborative dynamics and heterogeneous interactions among multiple vehicles in SVEC systems [31]. Ji et al. [32] proposed a semantic-aware task offloading system based on Multi-Agent Proximal Policy Optimization (MAPPO) and designed a unified quality of experience standard for different tasks, which can extract the semantic information of the task and offload it to the edge server. However, their framework lacks adaptive mechanisms to handle the heterogeneity and semantic relevance fluctuations in realistic vehicular environments. Therefore, there is a pressing need for a solution that dynamically balances semantic communication efficiency, latency, and resource consumption while accommodating the diverse characteristics of multimodal tasks.

Note that most of the current work does not fully consider the complexity of multimodal data in SVEC, and how to dynamically provide vehicles with different modal semantic feature extraction, task offloading and resource allocation strategies. In this paper, we propose a novel SVEC architecture

for semantic feature extraction, task offloading, which can quickly extract key semantic features in different modal tasks and efficiently offload them to edge servers. Meanwhile, we propose a diffusion-based multi-agent reinforcement learning framework to adaptively adjust semantic extraction, task offloading and resource allocation strategies in complex SVEC environments.

TABLE I  
MAIN SYMBOLS

| Notation          | Definition   |
|-------------------|--|
| $N$               | The number of vehicles in SVEC                           |
| $d_n^m$           | Data size of the task of vehicle $n$                     |
| $\rho_n(t)$       | The task offloading decision of vehicle $n$              |
| $\zeta_n^m$       | Priority of the different modal task by vehicle $n$      |
| $r_n(t)$          | Transmission rate from vehicle $n$ to the edge server    |
| $t_n^m(t)$        | Total processing latency of connected vehicle $n$        |
| $E_n^m(t)$        | Total energy consumption for vehicle $n$ 's current task |
| $\lambda_n(t)$    | Semantic extraction factor of vehicle $n$                |
| $p^{max}$         | Maximum transmit power constraint                        |
| $T_{n,m}^{max}$   | Maximum execute latency constraint                       |
| $F_n^{local,max}$ | Maximum computing capability of vehicle $n$              |
| $F^{max}$         | Maximum computing capability of edge server              |
| $E_n^{max}$       | Maximum battery capacity of vehicle $n$                  |
| $B^{max}$         | The upper limitation of total bandwidth                  |
| $\pi_{\theta_n}$  | The parameter of the behavior actor network              |
| $\pi_{\phi_n}$    | The parameter of the behavior critic network             |

## III. SYSTEM MODEL

### A. SVEC Architecture

Fig. 1 depicts a SVEC architecture for semantic feature extraction and task offloading. In SVEC, connected vehicles need to perceive the surrounding environment in real-time to improve safety, including obstacles, pedestrians, lane lines, etc., and achieve collaborative perception and decision-making by sharing environmental perception information among multiple vehicles. The core of SVEC lies in its ability to extract meaningful semantic information from multimodal sensor data, such as that collected from cameras, microphones, and other sensors. This extracted semantic information enables vehicles to offload tasks to the edge, utilizing the computing power of the edge server to process complex tasks like object recognition, lane detection, and traffic sign recognition. The semantic feature extraction and task offloading process in SVEC is as follows: 1) Semantic Extraction: Each vehicle extracts task-relevant semantic information using different semantic encoders deployed within the vehicle system. These encoders analyze the sensor data to identify and extract the key features required for the task. 2) Task Offloading: The extracted semantic information is sent to the edge server through the uplink channel. This offloading process is adaptive, depending on the vehicle's task preferences, priority, and network conditions. 3) Processing and Result Return: The edge server decodes the received semantic information, processes the task, and returns the processed results to the vehicle, enabling it to act on the information for further decision-making.

Considering the multimodal nature of tasks in SVEC, we focus on computational tasks involving text, image, and audio

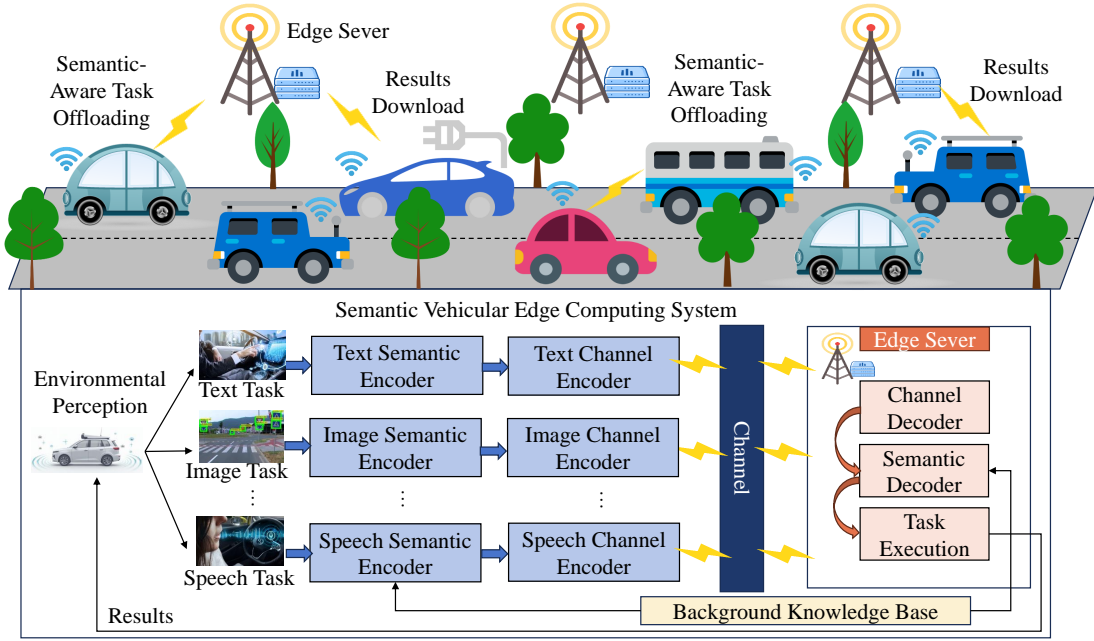


Fig. 1. The proposed SVEC architecture for semantic feature extraction and task offloading. This architecture allows vehicles to extract semantic features from the environment, offload tasks according to priority and resource availability, and optimize resource allocation for efficient operation.

data, and showcase the overall process of semantic communication. The proposed SVEC consists of semantic transmitters deployed in connected vehicles and corresponding receivers deployed on edge servers. The semantic transmitter includes both a semantic and a dynamic channel encoder, while the receiver is comprised of a channel and a semantic decoder. We utilize the Transformer and CNN to implement the transmitter and receiver. Specifically, for text data, the Transformer is employed to extract semantic information in natural language processing. For image data, we utilize CNN to capture key features and represent them as semantic vectors. For speech data, preprocessing converts the audio into a mel-spectrogram, which is then fed into the CNN to extract speech semantics.

### B. System Model

We define the set of connected vehicles as  $\mathcal{V} = \{1, 2, \dots, n, \dots, N\}$ . The data size of the task currently being processed by vehicle  $n$  is denoted as  $d_n^m$ , where  $m \in \{\mathcal{T} : \text{text}, \mathcal{I} : \text{image}, \mathcal{S} : \text{speech}\}$ . Each input is then processed to extract the corresponding encoded features. The task data  $d_n^m$  undergoes feature extraction using a semantic encoder  $f_m$  specific to the modality. After extracting the semantic features, the next step is to utilize the channel encoder  $z_m$  to encode the semantic information into a form suitable for transmission over the wireless channel. Therefore, the encoded semantic signal is represented as:

$$s_n^m = z_m(f_m(d_n^m; \theta_m)), \quad (1)$$

where  $z_m$  and  $f_m$  are the channel encoder and semantic encoder, respectively.  $d_n^m$  is the task data generated by vehicle  $n$ .  $\theta_m$  is the parameter of the semantic encoder of modality  $m$ . The transfer process involves transmitting the encoded

semantic signal  $s_n^m$  from the connected vehicle  $n$  to the edge server. The transmitted signal  $y_n^m$  can be represented as:

$$y_n^m = h_n s_n^m + n_m, \quad (2)$$

where  $h_n$  denotes the channel gain from vehicle  $n$  to the edge server, while  $n_m$  represents the noise power of zero-mean additive white Gaussian noise. The edge server then decodes the received semantic information through the decoder, where the decoded output is given by:

$$\widehat{s}_n^m = f_m^{-1}(z_m^{-1}(y_n^m; \theta_m)), \quad (3)$$

To improve communication quality and mitigate co-channel interference, we introduce orthogonal frequency division multiple access to support high-efficiency communication systems, which allows for multiple users to transmit simultaneously over different frequency subchannels, thus improving overall communication performance. Therefore, the transmission rate from connected vehicle  $n$  to the edge server at time slot  $t$  can be denoted as:

$$r_n(t) = b_n(t) \log_2 \left( 1 + \frac{p_n(t) h_n(t)}{\sigma^2} \right), \quad (4)$$

where  $b_n(t)$  is the uplink bandwidth resource obtained by vehicle  $n$ .  $p_n(t)$  represents the transmit power of vehicle  $n$ , and  $\sigma^2$  denotes the noise power. The channel gain  $h_n(t)$  is given by  $h_n(t) = g_n(t) |u_n(t)|^2$ , where  $g_n(t)$  is the large-scale fading and  $u_n(t)$  is the small-scale fading. We define a semantic extraction factor  $\lambda_n \in (0, 1]$  for connected vehicle  $n$ . The larger the  $\lambda_n$ , the lower the semantic compression and the more data needs to be transmitted. Conversely, the smaller the  $\lambda_n$ , the higher the semantic compression and the less data needs to be transmitted. When  $\lambda_n = 1$ , the vehicle will

transmit all the data for the task. Therefore, the transmission latency is given by:

$$t_n^{trans}(t) = \frac{\rho_n(t)d_n^m \lambda_n}{r_n(t)}, \quad (5)$$

where  $\rho_n(t) \in \{0, 1\}$ . Tasks are processed locally at vehicle  $n$  when  $\rho_n(t) = 0$ , or offloaded to the edge server when  $\rho_n(t) = 1$ .  $d_n^m(t)$  is the size of the raw task generated by vehicle  $n$ .

### C. Computing Model

The computation latency for connected vehicle  $n$  depends on whether the task is processed locally on the vehicle or offloaded to the edge server. The computation latency of connected vehicle  $n$  can be defined as:

$$t_n^{veh}(t) = \begin{cases} \frac{(1 - \rho_n(t))d_n^m(t)c_n^m}{F_n}, & \text{if } \rho_n(t) = 0, \\ \frac{\rho_n(t)d_n^m(t)c_n^m}{F_n \lambda_n}, & \text{if } \rho_n(t) = 1, \end{cases} \quad (6)$$

where  $c_n^m$  is the computing consumption of the current task computed by vehicle  $n$ .  $F_n$  is the computing capability of vehicle  $n$ . When tasks are offloaded to the edge server, the computation time of tasks on the edge server can be expressed as:

$$t_n^{edge}(t) = d_n^m(t)c_e^m \eta \sum_{n \in N} \rho_n(t)/F_e, \quad (7)$$

where  $c_e^m$  is the computing consumption of the current task computed by the edge server, and  $\eta$  is a positive compensation factor.  $F_e$  is the computing capability of the edge server. The total processing latency for vehicle  $n$ 's current task is the sum of the transmission latency, vehicle computation latency, and edge server computation latency. Therefore, we can define the total processing latency as:

$$\begin{aligned} t_n^m(t) &= \rho_n(t)(t_n^{trans}(t) + t_n^{veh}(t) + t_n^{edge}(t)) \\ &\quad + (1 - \rho_n(t))t_n^{veh}(t) \\ &= \rho_n(t)(t_n^{trans}(t) + t_n^{edge}(t)) + t_n^{veh}(t), \end{aligned} \quad (8)$$

### D. Energy Consumption Model

The energy consumption of connected vehicles and edge servers plays a significant role in optimizing the SVEC performance. The energy consumption of each vehicle depends on various factors, including the vehicle's operation time, computing power, and the energy required for communication with the edge server. Therefore, the energy consumption of connected vehicle  $n$  can be denoted as:

$$E_n^{veh}(t) = \kappa_n t_n^{veh}(t) F_n^3 + p_n(t) t_n^{trans}(t), \quad (9)$$

where  $\kappa_n$  is the energy coefficient of vehicle  $n$ . Similarly, the energy consumption of the edge server associated with vehicle  $n$  can be described as:

$$E_n^{edge}(t) = \kappa_e t_n^{edge}(t) (F_e / \sum_{n \in N} \rho_n(t))^3, \quad (10)$$

where  $\kappa_e$  is the energy coefficient of the edge server. Finally, the total energy consumption for vehicle  $n$ 's current task is the sum of the energy consumed by both the vehicle and the edge server, which can be expressed as:

$$E_n^m(t) = E_n^{veh}(t) + E_n^{edge}(t), \quad (11)$$

### E. Task Prioritization Model

In SVEC, connected vehicles often have different priorities for tasks in different modalities due to time-varying environments and differences in vehicle states. According to the allowed latency threshold, tasks of different modalities are classified into high-priority and low-priority tasks. High-priority tasks, such as navigation and vehicle road perception, are subject to stringent delay constraints. If these tasks cannot be completed within their maximum tolerable latency, they are considered a failure, which may lead to significant consequences for the vehicle's safety. In contrast, low-priority tasks, such as in-vehicle entertainment applications, have more flexible latency requirements. While latency in these tasks may degrade user experience, they do not affect the vehicle's overall operation. Therefore, the utility function for high-priority tasks can be defined as:

$$P_{n,m}^{high}(t) = \max(-C, \log_2(1 + T_{n,m}^{max}(t) - t_n^m(t))), \quad (12)$$

where  $T_{n,m}^{max}(t)$  represents the maximum latency limit, and  $C$  is a constant indicating the penalty for failing to complete a high-priority task. For low-priority tasks, the time constraint is more relaxed, and the low-priority task is still available even if it is not completed within the maximum latency. The difference between the latency of the task and the maximum delay latency affects the utility of the low-priority task, where the larger the delay difference, the faster the utility decreases. Therefore, the low-priority utility function is given by:

$$P_{n,m}^{low}(t) = C \cdot (1 + (T_{n,m}^{max}(t) - t_n^m(t)))^{-1}, \quad (13)$$

Therefore, we can define the priority-based latency utility function as:

$$P_n^m(t) = \zeta_n^m P_{n,m}^{high}(t) + (1 - \zeta_n^m) P_{n,m}^{low}(t), \quad (14)$$

where  $\zeta_n^m \in \{0, 1\}$  is the priority of the different modal task set by vehicle  $n$ ,  $\zeta_n^m = 0$  means the task is of low priority, otherwise, it is of high priority.

## IV. PROBLEM FORMULATION

### A. Problem Formulation

To assess the semantic communication performance in SVEC, we design the semantic fidelity of the data received and decoded by the edge server from the vehicle, which is an indicator that quantifies the similarity between the original vector data and the received vector data at the semantic level. The semantic fidelity is evaluated through a siamese network model, which compares the original encoded data with the decoded data to capture any distortions or losses in semantic information during transmission and decoding. By leveraging the siamese network model, we can effectively measure how well the semantic meaning is preserved after the communication process, thus offering a more accurate representation of the communication quality at the semantic level. The semantic fidelity can be mathematically expressed as:

$$S_n^m = \frac{\mathbf{s}_n^m \cdot \widehat{\mathbf{s}}_n^m}{\|\mathbf{s}_n^m\| \cdot \|\widehat{\mathbf{s}}_n^m\|}, \quad (15)$$

where  $\mathbf{s}_n^m$  and  $\widehat{\mathbf{s}}_n^m$  refer to the embedding vectors in the vehicle's encoder and edge server's decoder.

In order to accurately evaluate the overall efficiency and performance of the SVEC system, we introduce the system gain as a utility function, which serves as a comprehensive measure of the system's performance. This metric is defined as the weighted sum of three key performance indicators: semantic fidelity, task priority, and energy consumption. These performance metrics are crucial for optimizing the task offloading process as they jointly reflect the trade-offs among semantic accuracy, offloading efficiency, and resource consumption. The semantic fidelity is incorporated to ensure that the integrity of task-relevant information is preserved throughout the process, preventing the loss of important data that could impact decision-making, especially in safety-critical applications. The task priority is addressed by prioritizing critical tasks to ensure low-latency execution, while non-critical tasks are managed effectively with more flexible latency tolerances. Finally, energy consumption is a critical factor for the sustainability of intelligent vehicle systems, and we have incorporated it into the utility function to optimize resource usage while minimizing unnecessary energy expenditure. This composite metric, system gain, captures the interaction between these factors while accounting for the varying importance of each task based on its priority level. Therefore, the utility function for system gain is formally expressed as:

$$U_n(t) = \sum_{n=1}^N \psi_n^e E_n^m(t) + \psi_n^t P_n^m(t) + \psi_n^f S_n^m(t), \quad (16)$$

where  $\psi^e$ ,  $\psi^t$  and  $\psi^f$  are the degrees of preference for energy consumption, latency and semantic fidelity, respectively. The sum of  $\psi^e$ ,  $\psi^t$  and  $\psi^f$  are equal to 1. To determine the optimal offloading, semantic compression rate, offloading power, and bandwidth allocation strategy in the SVEC environment, it is essential to achieve the best balance between latency and resource consumption while staying within a given resource budget. To this end, we define a common optimization goal that seeks to minimize latency and resource consumption while improving semantic fidelity. The optimization problem can be formally expressed as:

$$\mathbf{P0} : \max_{\{\rho, \lambda, p, b\}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N U_n(t) \quad (17)$$

$$\text{s.t.} \quad \rho_n(t) \in \{0, 1\}, \quad \forall n \in \mathcal{V}, \quad (17a)$$

$$0 < \lambda_n(t) \leq 1, \quad \forall n \in \mathcal{V}, \quad (17b)$$

$$p_n(t) \leq p^{max}, \quad \forall n \in \mathcal{V}, \quad (17c)$$

$$t_n^m \leq T_{n,m}^{max}, \quad \forall n \in \mathcal{V}, \quad (17d)$$

$$F_n \leq F_n^{local\_max}, \quad \forall n \in \mathcal{V}, \quad (17e)$$

$$F_e + \sum_{n=1}^N F_n \leq F^{max}, \quad \forall n \in \mathcal{V}, \quad (17f)$$

$$S_n^m \geq S_{min}, \quad \forall n \in \mathcal{V}, \quad (17g)$$

$$\sum_{n=1}^N b_n(t) \leq B^{max}, \quad \forall n \in \mathcal{V}, \quad (17h)$$

$$\lim_{T \rightarrow \infty} \sum_{t=1}^T E_n^m(t) \leq E_n^{max}, \quad \forall n \in \mathcal{V}, \quad (17i)$$

where  $\{\rho, \lambda, p, b\} = \{\rho_n(t), \lambda_n(t), p_n(t), b_n(t)\}, \forall n \in N$ . (17a) is the offloading choice, (17b) is the semantic extraction factor,  $p^{max}$  in (17c) is the maximum transmit power constraint,  $T_{n,m}^{max}$  in (17d) is the maximum execute latency constraint,  $F_n^{local\_max}$  in (17e) is the maximum computing capability of vehicle  $n$ ,  $F_e$  in (17f) is the maximum computing capability of edge sever,  $S_{min}$  in (17g) is the range of semantic fidelity,  $B^{max}$  in (17h) is the maximum total bandwidth of edge sever,  $E_n^{max}$  in (17i) is the battery capacity of vehicle  $n$ .

### B. Problem Simplification

The difficulty in directly solving the **P0** problem arises from the long-term average energy constraint, which intertwines the strategies for semantic extraction, task offloading, and resource allocation across multiple time slots. To overcome this challenge, we employ a *Lyapunov* optimization approach that introduces a virtual energy-deficient queue. This queue helps manage the coupling between resource allocation decisions and task offloading decisions to ensure that the long-term energy constraint is met. *Lyapunov* optimization is a well-established technique for controlling dynamic systems that optimizes performance by balancing various constraints while ensuring system stability and efficiency [33]. In our problem, *Lyapunov* optimization can facilitate the decision-making process by managing the energy consumption dynamically, allowing for real-time adjustments to semantic extraction, task offloading, and resource allocation strategies. The evolution of the energy-deficient queue follows a specific dynamics, which can be expressed as:

$$Q_n(t+1) = \max \{Q_n(t) + E_n^m(t) - E_n^{max}, 0\}. \quad (18)$$

Regarding the optimization objective for all vehicles, we reformulate the original problem **P0** into:

$$\mathbf{P1} : \max_{\{\rho, \lambda, p, b\}} \sum_{t=1}^T \sum_{n=1}^N U_n(t) + Q_n(t) \cdot (E_n^m(t) - E_n^{max}) \quad (19)$$

s.t. Constraints (17a) - (17h) in **P0**,

Therefore, our subsequent objective is to solve the mixed-integer programming problem **P1** at each time step, which presents significant computational challenges due to its hybrid discrete-continuous decision variables  $\{\rho, \lambda, p, b\}$  and non-convex objective function. Conventional methods—including heuristic algorithms and decomposition-based techniques—often yield suboptimal solutions or suffer from prohibitive computational complexity, especially as the problem scales. To this end, we next introduce the diffusion model into the proposed SVEC framework to solve **P1**.

### C. The Forward Process of Probability Noising

In recent years, the Denoising Diffusion Probabilistic Model (DDPM), an emerging generative model, has gained prominence in the field due to its unique advantages. As shown in Fig. 2, DDPM typically involves two key processes: the forward process and the reverse process. The forward process involves progressively adding noise to the real data. This is

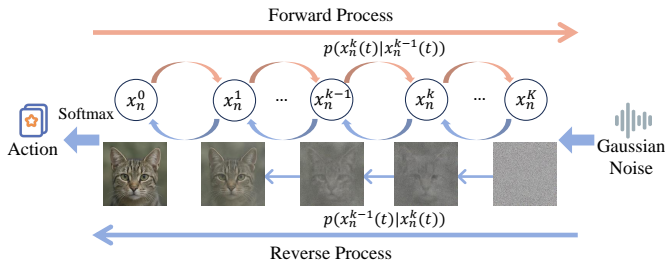


Fig. 2. The illustration of the diffusion model for generating optimal action. This model utilizes a diffusion process to iteratively refine the action selection in multi-agent systems.

done through a series of steps, where at each step, a small amount of Gaussian noise is added to the original data, slowly degrading the data until it becomes indistinguishable from pure noise. The reverse process is where the actual data generation takes place. Starting from random noise, the reverse process gradually removes the noise step-by-step. At each step, the model attempts to denoise the current noisy state by leveraging learned parameters, ultimately reconstructing data that resembles the original real data. In this paper, problem **P1** involves a hybrid solution space composed of both discrete and continuous variables, which leads to an exponential increase in complexity as the number of vehicles grows. To address this challenge, we incorporate the reverse diffusion process of the denoising diffusion probabilistic model (DDPM) into the multi-agent reinforcement learning (MARL) framework. Specifically, we utilize the multi-step denoising mechanism of DDPM to progressively refine random noise into high-quality action samples, thereby guiding the agents toward more optimal solutions during training. This not only enriches the diversity of policy exploration but also enhances training stability in complex, dynamic environments. In addition, benefiting from the implicit generation capability of DDPM, the agent is able to effectively capture the dynamic changes of the SVEC environment, thereby providing a fine-grained, time-aware state representation. Next, we introduce the forward process and the reverse process of DDPM, respectively.

During the forward process, starting from the initial distribution  $x_n^0$ , Gaussian noise is progressively added to generate the sequence  $\{x_n^1, x_n^2, \dots, x_n^K\}$ . At each step, the transition from  $x_n^{k-1}$  is modeled as a normal distribution with mean  $\sqrt{1 - \beta_k}x_n^{k-1}$  and variance  $\beta_k \mathbf{I}$ , where  $\beta_k \mathbf{I}$  adjusts the amount of noise. This process effectively introduces noise to the initial decision vector at each step, gradually transforming it into pure noise as  $k$  increases. Therefore, the forward process can be defined as:

$$p(x_n^k | x_n^{k-1}) = \mathcal{N}\left(x_n^k; \sqrt{1 - \beta_k}x_n^{k-1}, \beta_k \mathbf{I}\right), \quad (20)$$

where the variance changes over time, the edge server adjusts the amount of noise added in the forward process, this can be described as  $\beta_k = 1 - e^{-\frac{\beta_{min}}{k} - \frac{2k-1}{2K^2}(\beta_{max} - \beta_{min})}$ ,  $\beta_{min}$  and  $\beta_{max}$  represent the minimum variance and maximum variance, respectively, and  $K$  represents the diffusion step.

The forward process is a Markov process where each state  $x_n^k$  depends only on its previous state  $x_n^{k-1}$ . As a result, the

distribution of  $x_n^K$  given  $x_n^0$  can be expressed as the product of conditional transitions over all denoising steps, as follows:

$$p(x_n^K | x_n^0) = \prod_{k=1}^K p(x_n^k | x_n^{k-1}), \quad (21)$$

Then we can associate the initial state  $x_n^0$  with any intermediate state  $x_n^k$  in the diffusion sequence. This relationship can be expressed as:

$$x_n^k = \sqrt{\bar{\alpha}_k}x_n^0 + \sqrt{1 - \bar{\alpha}_k}\bar{\epsilon}_k, \quad (22)$$

where,  $\alpha_k = 1 - \beta_k$ , and  $\bar{\alpha}_k = \prod_{z=1}^k \alpha_z$  represents the cumulative product of  $\alpha_z$  over the previous denoising steps  $z$ . Additionally,  $\bar{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is a standard normal noise vector. However, in the SVEC system, the optimal decision  $x_n^0$  cannot usually be obtained directly in the process of solving the optimization problem **P1**, because  $x_n^0$  represents the ideal decision under the observation condition, and is usually unknown or unavailable. Therefore, in the proposed scheme, we do not consider the forward diffusion process of DDPM.

#### D. The Reverse Process of Probability Inference

The reverse process is a denoising process that infers the target  $x_n^0$  from pure Gaussian noise  $x_n^K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . It is implemented through a neural network trained to predict the noise component at each diffusion step, thereby enabling iterative refinement of the noisy sample towards the clean data distribution. In this paper, we utilize the reverse process to gradually remove noise and obtain the optimal semantic extraction, task offloading and resource management strategies. This reverse process can be described as:

$$p(x_n^{k-1} | x_n^k) = \mathcal{N}(x_n^{k-1}; \mu_n^k(x_n^k), \frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \beta_k \mathbf{I}), \quad (23)$$

where  $\mu_n^k$  represents the predicted mean value of  $x_n^{k-1}$  given the noisy sample  $x_n^k$ . Additionally, the variance  $\frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \beta_k$  controls the amount of noise to be removed at each step. Together, the learned mean and the deterministic variance guide the reverse process [34], enabling the model to progressively recover the target  $x_n^0$  from the noisy state  $x_n^K$ .

According to Bayes' theorem, the reverse diffusion process can be mathematically derived through the conditional probability distribution of the forward process, rather than directly calculating the reverse process itself, allowing us to express the reverse process as a function of the forward process. Specifically, the reverse process becomes a reparameterization of the forward process, whose goal is to gradually denoise the samples. In this way, we derive the mean of each step  $k$  of the reverse process by considering both the noisy data and the learned noise model. The mean can be expressed as:

$$\mu_n^k(x_n^k) = \frac{\sqrt{\alpha_k}(1 - \bar{\alpha}_{k-1})}{1 - \bar{\alpha}_k} x_n^k + \frac{\sqrt{\alpha_{k-1}}\beta_k}{1 - \bar{\alpha}_k} x_n^0, \quad (24)$$

The reconstructed sample  $x_n^0$  can be directly obtained by applying the reverse process iteratively based on Eq. (23). Specifically, at each time step, the mean is calculated using the

noisy sample  $x_n^K$  and the predicted noise. The reconstruction of  $x_n^0$  involves progressively refining the noisy sample by removing noise at each step using the computed mean, as shown in:

$$x_n^0 = \frac{1}{\sqrt{\alpha_k}} x_n^k - \frac{\sqrt{1 - \bar{\alpha}_k}}{\sqrt{\alpha_t}} \cdot \tilde{\varepsilon}_\theta, \quad (25)$$

where the denoising operation is implemented through a deep neural network  $\tilde{\varepsilon}_\theta(x_n^k, k, o_n)$ . This model is responsible for predicting the noise component at each step of the reverse process according to the observation  $o_n$ , helping to refine the sample  $x_n^k$  and recover the original data  $x_n^0$ . By carefully controlling the noise magnitude, the model can maintain stable learning while progressively denoising the sample towards the target distribution.

Since the reverse process introduces new independent noise  $\tilde{\varepsilon}_k$ , which is different from the forward noise  $\varepsilon$  at each denoising step  $k$ , we cannot directly obtain  $x_n^0$  using Eq. (25). Therefore, to estimate the mean more effectively, we substitute Eq. (25) into Eq. (24). This substitution allows us to approximate the denoised state  $x_n^{k-1}$  based on the predicted noise at each step, leading to the following expression:

$$\mu_{\theta_n}^k(x_n^k, k, o_n) = \frac{1}{\sqrt{\alpha_k}} \left( x_k - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_k}} \cdot \tilde{\varepsilon}_{\theta_n} \right), \quad (26)$$

Due to the non-differentiability of the sampling operation during the training of diffusion models, the model is prevented from learning efficiently via gradient descent. We overcome this problem through the reparameterization technique, expressing the reverse process sampling as:

$$x_n^{k-1} = \mu_{\theta_n}^k(x_n^k, k, o_n) + \frac{1 - \alpha_k}{\sqrt{1 - \bar{\alpha}_t}} \beta_t \cdot \varepsilon, \quad (27)$$

where  $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is a noise term independently sampled from the standard Gaussian distribution. By iteratively applying the reverse update rule, as defined in Eq. (28), we can progressively compute all the intermediate samples  $x_n^K$  for each step  $k$ . This iterative process gradually denoises the sample, starting from a randomly generated normal noise  $x_n^K$  and refining it towards the target output  $x_n^0$ . To convert the output sample  $x_n^0$  into a probability distribution, we apply the softmax function. The final output is normalized to obtain a valid probability distribution:

$$\pi_{\theta_n}(o_n(t)) = \left\{ \frac{e^{x_n^0}}{\sum_{n=1}^N e^{x_n^0}}, \forall n \in \mathcal{V} \right\}. \quad (28)$$

where  $x_n^0$  represents the generated sample. The resulting probability distribution  $\pi_{\theta_n}(o_n(t))$  represents the likelihood of each possible decision or outcome under the observed state  $o_n$ , enabling the model to make decisions based on the learned distribution.

## V. DIFFUSION-BASED MARL OPTIMIZATION SCHEME FOR DECISION MAKING

In this section, we first model the problem as a Markov Decision Process (MDP). We then present the architecture of our diffusion-based multi-agent deep reinforcement learning (DMADRL) algorithm. Finally, we provide an analysis of the computational complexity associated with the proposed method.

### A. MDP Model

Next, we model problem **P1** as a MDP to effectively capture the sequential decision-making process involving semantic extraction, task offloading, and resource allocation in SVEC systems. By considering the long-term impact of state transitions and real-time decisions, the MDP framework is able to derive the optimal strategy that dynamically balances immediate performance metrics with future resource availability, ultimately ensuring sustainable and efficient semantic extraction and task offloading across the SVEC system.

**State:** The state  $\mathcal{O}$  reflects the current environment and provides the necessary information to support decision-making for the vehicle  $n$ . Within the state space, each state  $o_n$  is represented as a vector that integrates all the relevant information required for determining the next action. This includes not only the characteristics of the current task, such as task priority, local resources, and deadlines. Thus, the state  $o_n(t)$  of agent  $n$  is formulated as:

$$o_n(t) = \{d_n^m, E_{n,m}^{max}, T_{n,m}^{max}, \zeta_n^m, F_n^{local-max}\}, \quad (29)$$

The joint state space  $\mathcal{O}(t)$  represents the combined set of all possible states for all agents in the SVEC system, and is given by:

$$\mathcal{O}(t) = \{o_1(t), \dots, o_n(t), \dots, o_N(t)\}. \quad (30)$$

**Action:** The action  $a_n$  is determined by the diffusion-based network, where  $o_n$  represents the input to the network. Specifically,  $a_n \sim \pi_\theta(o_n)$ , meaning that the action  $a_n$  is sampled from the probability distribution represented by  $\pi_\theta(o_n)$ . Therefore, the action  $a_n$  of agent  $n$  can be formally represented as:

$$a_n(t) = \{\lambda_n(t), \rho_n(t), p_n(t), b_n(t)\}, \quad (31)$$

The joint action space  $\mathcal{A}(t)$  represents the set of all possible actions that can be taken by all agents at time step  $t$ , and can be represented as:

$$\mathcal{A}(t) = \{a_1(t), \dots, a_n(t), \dots, a_N(t)\}. \quad (32)$$

**Reward:** The reward represents the return received by the vehicle  $n$  after it implements an action  $a_n(t)$ . This reward influences the vehicle's subsequent training strategy and contributes to its long-term performance. The reward function encourages efficient and accurate decisions that enhance the vehicle's overall performance, and can be defined as:

$$r_n(t) = V \cdot U_n(t) + Q_n(t) \cdot (E_n^m(t) - E_n^{max}), \quad (33)$$

where  $V$  is a positive control parameter. Each agent's reward  $r_n(t)$  is influenced by the actions taken by others, creating a collaborative SVEC environment that encourages semantic extraction, task offloading, and resource allocation, ultimately achieving optimal performance across the SVEC system. Therefore, the joint reward space can be defined as:

$$\mathcal{R}(t) = \{r_1(t), \dots, r_n(t), \dots, r_N(t)\}. \quad (34)$$

The primary objective of the vehicle in SVEC is to establish an optimal trade-off among semantic fidelity, execution latency, and energy expenditure. For the reinforcement learning

framework, this translates to maximizing the expected return, formally defined as the time-discounted summation of future rewards:

$$R(o_n(t), \pi_{\theta_n}) = \mathbb{E} \left[ \sum_{t=1}^{+\infty} \sum_{n=1}^N \gamma r_n(t) | o_n(t), \pi_{\theta_n} \right]. \quad (35)$$

where  $\gamma \in (0, 1]$  is the discount rate, determining the weight of future rewards relative to immediate rewards.

### B. Diffusion-based MARL for Decision Making in SVEC

In SVEC, connected vehicles not only generate tasks with different modalities and service requirements, but also have differences in storage, computing, communication capabilities, etc. Therefore, it becomes crucial to determine the semantic extraction, task offloading and resource allocation strategies of connected vehicles in a fine-grained manner. In the traditional reinforcement learning framework, the agent obtains feedback information by interacting with the environment, performing actions, and observing the state changes of the environment. However, in practical deployments, environmental stability is frequently compromised by the complex interdependencies among heterogeneous agents, impeding the rapid convergence of learning processes. To enable coordinated optimization of semantic extraction, task offloading, and resource allocation in SVEC systems, we devise a novel DMADRL algorithm. As shown in Fig. 3, we model each vehicle as an independent agent, where every vehicle deploys a DMADRL model. The DMADRL model is composed of several key components: a global replay memory, which stores experiences from all agents to facilitate efficient learning; a target network, used to stabilize training by providing fixed targets for the Q-values; and a behavior network, which guides the decision-making of each agent. To enhance collaboration among multiple vehicles, we employ the Centralized Training Decentralized Execution (CTDE) mechanism, which enables the agents to be trained in a centralized manner while executing their policies independently, allowing them to collaboratively explore the solution space. The behavior network employs an actor-critic framework. The actor network  $\pi_{\theta_n}(o_n(t))$  and the critic network  $Q(\mathcal{O}, \mathcal{A} | \phi_n)$  work together to guide the agent's decision-making process, where  $\theta_n$  and  $\phi_n$  denote the policy and value function parameters, respectively. The target network is designed to stabilize the performance of the model and exhibits the same framework as that of the behavior network. Furthermore, each vehicle combines its states to generate comprehensive information, which is then stored in the replay memory  $\mathcal{D}$ . The replay memory consists of the current state  $\mathcal{O}$ , the action  $\mathcal{A}$ , the next state  $\mathcal{O}'$ , and the reward function  $\mathcal{R}$ . The behavior network is trained by randomly sampling experiences from the global replay memory.

The actor network performs semantic extraction, task offloading, and resource allocation decisions, which are generated by mapping the current state  $o(t)$  to actions  $\pi_{\theta_n}$  based on the reverse process of DDPM. However, sampling from discrete distributions is not differentiable, which makes it difficult to train using standard backpropagation algorithms. To overcome this challenge, we employ a technique called the

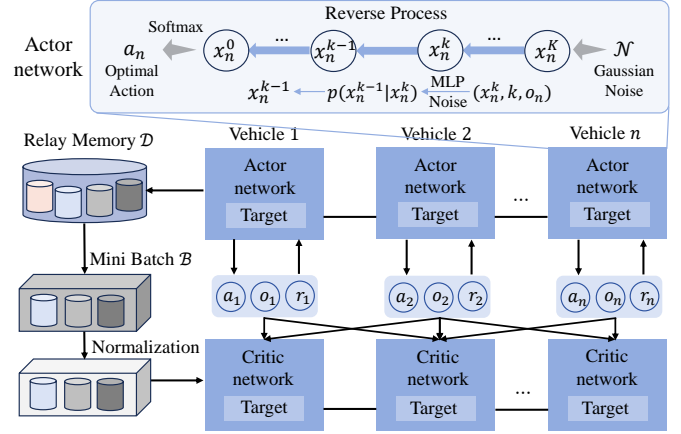


Fig. 3. The proposed DMADRL scheme for decision-making in SVEC. This scheme includes key components such as the actor, which generates actions based on the current state, the critic, which evaluates the selected actions, and the replay memory, which stores past experiences for improved learning.

**gumbel softmax trick** [35]. We assume the presence of  $N$  actor networks  $\pi_{\theta_n}$ . The training process employs experience replay with prioritized sampling from a centralized buffer  $\mathcal{D}$  storing historical transition tuples. During each training iteration, we uniformly sample a mini-batch  $\mathcal{B}$ , where each sample consists of a tuple  $(o, a, r, o')$ , and the deterministic policy gradient for each actor network  $\pi_{\theta_n}$  is computed, enabling the model to refine its decision-making process based on the feedback received from the environment. Therefore, the policy gradient of the actor network can be defined as:

$$\nabla_{\theta_n} \mathcal{J}(\theta_n) = \mathbb{E}_{o, a \sim \mathcal{D}} \left[ \nabla_{\theta_n} \pi_{\theta_n}(a_n | o_n) \nabla_{a_n} \mathcal{Q}_{\pi_{\theta_n}}(\mathcal{O}, a_1, \dots, a_N) \right], \quad (36)$$

where  $\mathcal{Q}_{\pi_{\theta_n}}(\mathcal{O}, a_1, \dots, a_N)$  denotes a centralized action-value function, which is computed by the corresponding critic network for all agents.

The objective of behavior critic network is to minimize the error between the current network's estimated  $Q$  value and the target  $Q$  value to optimize the critic network's parameters  $\phi_n$ . To improve the reliability of the critic network's evaluation of the action generated by the actor network,  $\mathcal{Q}_{\pi_{\phi_n}}$  is periodically updated based on the temporal difference error between the current action value estimate and the target value. For each agent  $n$ , the loss function of the critic network for agent  $n$  can be expressed as:

$$\mathcal{L}(\phi_n) = \mathbb{E}_{\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{O}'} \left[ (y - \mathcal{Q}_{\pi_{\phi_n}}(\mathcal{O}, a_1, \dots, a_N))^2 \right], \quad (37)$$

where

$$y = r_n + \gamma \mathcal{Q}_{\pi_{\phi_n}}(\mathcal{O}', a'_1, \dots, a'_N) \Big|_{a'_j \sim \pi_{\theta'_j}(o'_j)}, \quad (38)$$

where  $\mathcal{Q}_{\pi_{\phi_n}}$  is calculated by the target critic network for the next state  $\mathcal{O}'$  and the next action  $(a'_1, \dots, a'_N)$ .  $a'_j \sim \pi_{\theta'_j}(o'_j)$  is the action generated by the target actor network.

To improve the accuracy of the target values, we employ a soft update of the target networks at each training step. The soft update process of the target network can be defined as:

$$\begin{aligned}\theta'_n &\leftarrow \beta\theta_n + (1 - \beta)\theta'_n, \\ \phi'_n &\leftarrow \beta\phi_n + (1 - \beta)\phi'_n,\end{aligned}\tag{39}$$

where  $\beta$  is the soft update parameter used to control the update speed of the target network.

The decision-making process of vehicles is shown in Algorithm 1. In line 1-2, SVEC initializes the actor and critic network parameters to control and evaluate the vehicle actions. At the beginning of each agent’s decision process, a noise distribution  $x_n^K$  is initialized and sampled from a normal distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . In line 8-10, the algorithm applies a diffusion model to estimate  $\varepsilon_\theta$ , calculate the mean and distribution. After the denoising process is finished, the local action  $a_n(t)$  of agent  $n$  is determined according to the policy  $\pi_{\theta_n}$ , and then the action is executed, and the corresponding reward  $r_n(t)$  is calculated in line 11-12. In line 13-14, the transition  $(\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{O}')$  is stored in the replay buffer  $\mathcal{D}$ , and then the agent transitions to the next state  $o_n(t+1)$ . Once all actions have been executed and rewards have been observed, the algorithm proceeds to update the policy and evaluation network of each agent. In line 15-18, for each agent  $n$ , a mini-batch  $\mathcal{B}$  is sampled from the replay buffer  $\mathcal{D}$ , and the evaluation networks of the actor and the critic are updated. At the end of the episode, the target actor and critic networks are updated according to the soft update rule.

### C. Algorithm Complexity

We mainly consider the complexity of the proposed algorithm from two aspects, that is, the complexity associated with diffusion-driven action generation and the complexity associated with network updates. Following the complexity analysis methodology in [36], we emphasize that under the CTDE paradigm, the observation collection and policy updates for all vehicles are performed at a centralized server with sufficient computational resources. Consequently, our complexity analysis focuses primarily on the vehicle during the execution process. Given the parallel deployment of agents in the SVEC, the system complexity is bounded by the computation load of a single vehicle. Specifically, we assume that each agent  $n \in \mathcal{V}$  runs DMADRL based on the neural network depth  $L$ , hidden layer dimension  $d$ , and denoising step  $K$  of the diffusion model. Thus, the overall computational complexity of the algorithm is given by  $O((K \cdot L \cdot d^2 + \mathcal{B} \cdot P))$ , where  $P$  is the policy network parameter.

## VI. SIMULATION RESULTS

### A. Experimental Setup

1) *Simulation Parameters*: To evaluate the performance of the proposed DMADRL framework, we simulate a SVEC system. All the simulated experiments are conducted using a Python 3.8 and Pytorch 2.3 platform, with one Intel i7-13700K CPU and two NVIDIA RTX4090 GPUs. The simulation consists of 20 connected vehicles, each equipped with an

---

### Algorithm 1 DMADRL for Adaptive Decision-making

---

```

1 Initialize critic and actor network parameters;
2 for each episode do
3   Receive the initial local state  $\mathcal{O}$ ;
4   for each time slot  $t$  do
5     for  $n = 1$  to  $N$  do
6       Initialize noise distribution  $x_n^K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
7       for denoising step  $k = K$  to 1 do
8         Estimate  $\varepsilon_\theta$  via diffusion model;
9         Compute the mean and distribution;
10        Utilize the reparameterization technique
            to compute the distribution  $x_n^{k-1}$ ;
11      end
12      Determine the local action  $a_n(t)$  based on
            the probability distribution  $\pi_{\theta_n}$ ;
13      Execute the action and calculate the
            corresponding reward  $r_n(t)$ ;
14      Store the transition  $(\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{O}')$  in  $\mathcal{D}$ ;
15      Transfer to the next state  $o_n(t+1)$ ;
16      for each vehicle  $n = 1$  to  $N$  do
17        Sample mini-batch  $\mathcal{B} \sim \mathcal{D}$ ;
18        Update each vehicle’s actor network
            according to Eq. (36);
19        Update each vehicle’s critic network
            following Eq. (37);
20      end
21    end
22    Update the target actor and critic network
            according to Eq. (39);
23  end
24 end
```

---

independent DRL agent, where each agent is made up of an actor network and a critic network. The optimizer used to update the network parameters is Adam, where the learning rate of the Actor network is 0.0001 and the learning rate of the Critic network is 0.001, ensuring that both networks can be efficiently optimized during training. To further assess the framework’s performance in a multimodal SVEC, we select three distinct datasets: the European Parliament Minutes as the text dataset, the German Traffic Sign Recognition Benchmark as the image dataset, and the Edinburgh International Speech Corpus as the speech dataset. Each vehicle in the simulation is initialized with manually configured data distributions to simulate real-world variations in data availability and task types. The simulation parameters are listed in Tab. II.

2) *Benchmark Solutions*: To evaluate the performance of the proposed scheme, we compare it against the following five benchmarks. **Greedy [37]**: Each connected vehicle selects the optimal semantic extraction, task offloading and resource allocation strategy at the current moment based on the historical system gain. **SAC [38]**: SAC uses random strategies to explore the complex relationship between semantic feature extraction factors and channel states to achieve semantic communication collaborative optimization. **DDPG [39]**: This scheme calculates the quality of actions through the critic network, which

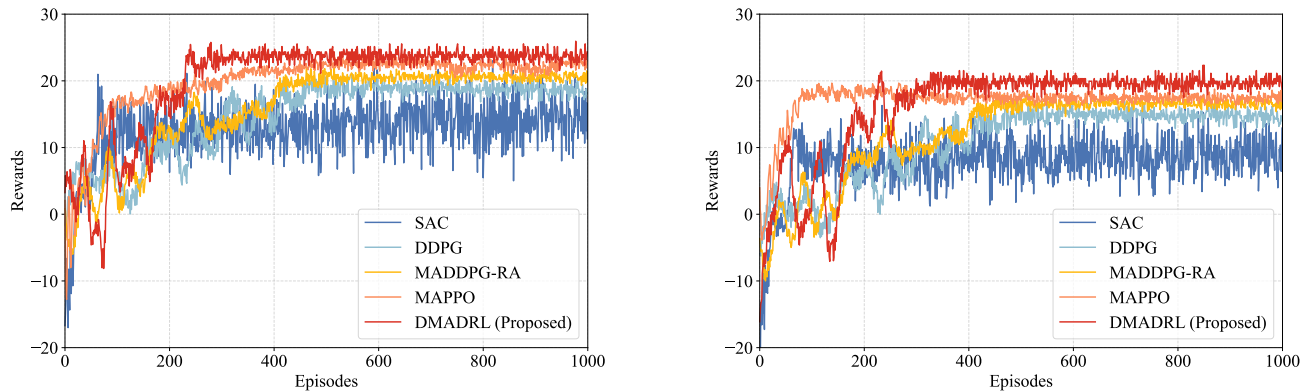


Fig. 4. Comparison of reward curves of DMADRL and benchmarks under different communication environments.

TABLE II  
SIMULATION PARAMETERS

| Parameter                              | Value                  |
|--|------------------------|
| Total channel bandwidth $B^{max}$      | 50MHz                  |
| Vehicle computation resource $F_n$     | [3, 5]GHz              |
| Vehicle transmit power $p_n$           | [0.6, 0.8]W            |
| Edge server computation resource $F_e$ | 100GHz                 |
| Noise power $\sigma^2$                 | $8 \times 10^{-12}$ mW |
| Data size of task $d_n^m$              | [1, 3]Mbits            |
| Semantic extraction factor $\lambda_n$ | [0.4, 1.0]             |
| Reward discount factor $\gamma$        | 0.99                   |
| Size of replay buff $\mathcal{D}$      | 50000                  |
| Target network update rate             | 0.001                  |
| Task request latency $T_{n,m}^{max}$   | [5, 25]ms              |
| Number of task types                   | [1, 3]                 |

evaluates the performance of the chosen actions. The actor network then adjusts its parameters based on the feedback from the critic network. **MADDPG-RA** [40]: Each vehicle drives the intelligent agent to compete for edge computing resources through a CTDE mechanism, where the Critic network models the impact of multi-vehicle joint actions and the Actor network generates policies that meet the constraints. **MAPPO** [41]: MAPPO takes into account the stability and sample efficiency issues in multi-agent environments, and limits the scope of policy updates to ensure that the policy does not change excessively, thereby avoiding an unstable training process.

### B. Simulation Results

1) *Convergence Performance*: Fig. 4 depicts the convergence behavior of the proposed scheme and the benchmarks under different communication scenarios. As shown in Fig. 4(a), in general, in the scenario of semantic communication, our proposed DMADRL is more stable, converges at 250 episodes, and obtains higher situational rewards in the training phase, showing the substantial benefits of the diffusion process. The diffusion process significantly improves the efficiency of action samples and alleviates the problem of policy collapse in multi-agent environments. The reward curve of SAC fluctuates greatly, and the reward value of

DDPG is always less than 20, indicating that their exploration efficiency in SVEC is insufficient. MADDPG-RA converges quickly in the early episodes but falls into a local optimum in the later episodes. MAPPO performs smoothly, and the reward value gradually rises to 22, showing the stability advantage of its conservative update strategy in multi-agent tasks, but its exploration ability is limited. As shown in Fig. 4(b), in the scenario of non-semantic communication, as the number of episodes increases, the DMADRL algorithm and the benchmarks finally converge to a smaller reward value than the semantic communication scenario. Compared with Fig. 4(b) the introduction of semantic communication improves the performance of all algorithms.

2) *Effect of the Number of Vehicles*: Fig. 5 shows the impact of the number of vehicles on the average reward value in different communication environments. In the semantic communication scenario, DMADRL always outperforms other benchmarks, demonstrating the efficiency of DMADRL in achieving task offloading and resource allocation in SVEC. As shown in Fig. 5(a), as the number of vehicles increases, the gap between DMADRL and other benchmarks further widens, demonstrating its scalability and excellent ability to handle larger networks. SAC and DDPG perform relatively poorly due to their limited ability to adapt to complex multi-agent environments. MADDPG-RA and MAPPO perform better, but are still inferior to DMADRL in terms of adaptability and dynamic decision-making capabilities. As shown in Fig. 5(b), in the non-semantic communication scenario, as the number of vehicles increases, the DMADRL algorithm and the benchmark eventually converge to a smaller reward value than the semantic communication scenario. Compared with Fig. 4(b), the average reward values of all algorithms are improved after the introduction of semantic communication. Fig. 4 and Fig. 5 show that the integration of semantic communication enhances the collaboration between agents and achieves more effective learning and reward maximization.

Fig. 6 depicts the impact of increasing the number of vehicles on the average latency. The DMADRL algorithm effectively reduces the average latency through semantic communication enhancement and always outperforms other

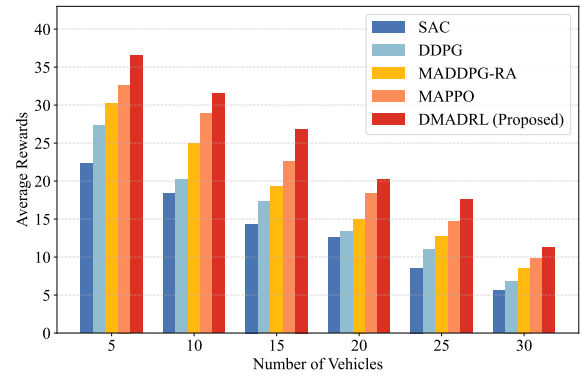
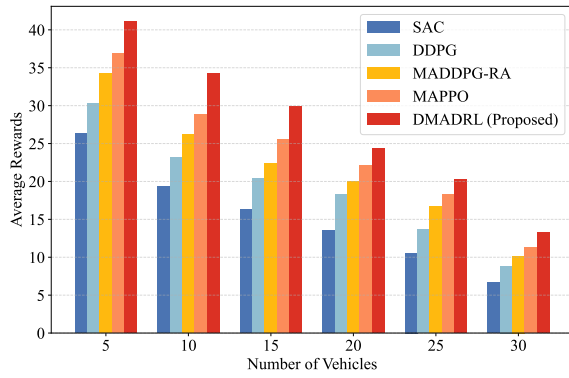


Fig. 5. Comparison of average rewards for different numbers of vehicles between DMADRL and the benchmarks under different communication scenarios.

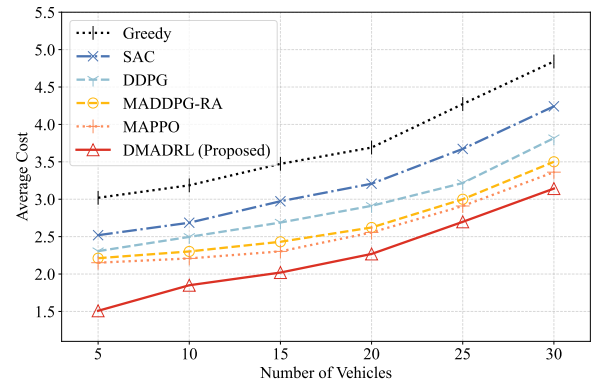
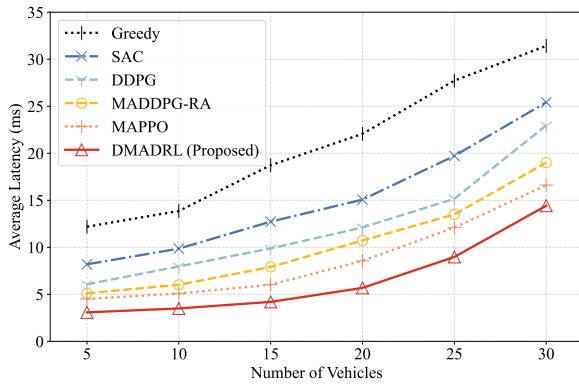


Fig. 6. Comparison of average system latency under different numbers of vehicles.

Fig. 7. Comparison of average cost under different numbers of vehicles.

algorithms. This performance advantage is primarily due to DMADRL’s ability to adapt to the dynamic conditions of the SVEC environment, where communication and task offloading are influenced by network congestion and resource availability. As shown in Fig. 6, the SAC and DDPG algorithms exhibit the highest latency and poor performance, especially when the number of vehicles increases. This is because the SAC and DDPG algorithms are inefficient in adapting to multi-agent environments and cannot effectively handle dynamic network conditions. Although MADDPG-RA and MAPPO have lower latency than SAC and DDPG, they still have higher latency and lower task offloading and resource allocation efficiency compared to DMADRL.

Fig. 7 compares the proposed scheme with the average cost under different numbers of vehicles, where the cost is calculated as the weighted sum of latency and 10 times the energy consumption, and lower values indicate better performance. As shown in Fig. 7, DMADRL maintains the lowest cost as the number of vehicles increases, demonstrating its effectiveness in maximizing task offloading performance while reducing the overall system cost. In contrast, the SAC and DDPG algorithms exhibit relatively high average costs, especially when the number of vehicles increases. These algorithms have difficulty adapting to dynamic network conditions,

resulting in inefficient task offloading and resource allocation. Although MADDPG-RA and MAPPO outperform SAC and DDPG, their costs are still higher than those of DMADRL, underscoring the superior trade-off between performance and cost achieved by DMADRL.

3) *Average Queue Length*: Fig. 8 compares the average queue lengths of the proposed scheme and the benchmark. As shown in Fig. 8, DMADRL maintains a stable minimum queue length as the number of events increases, highlighting the effectiveness of the algorithm in reducing congestion and optimizing task offloading. This is particularly important for ensuring that vehicles are able to efficiently offload tasks without experiencing excessive delays or congestion. In contrast, SAC and DDPG exhibit significantly longer queue lengths, especially as the number of events increases, reflecting their inability to efficiently manage network resources and task offloading. MADDPG-RA and MAPPO, while showing improvements over SAC and DDPG, still result in longer queues compared to DMADRL, reinforcing the superiority of DMADRL’s dynamic decision-making ability in minimizing queue lengths and optimizing resource allocation in complex environments like SVEC.

4) *Effect of the SNR and bandwidth*: Fig. 9 compares the semantic fidelity of the proposed scheme and the benchmarks for different modal data as the signal-to-noise ratio (SNR)

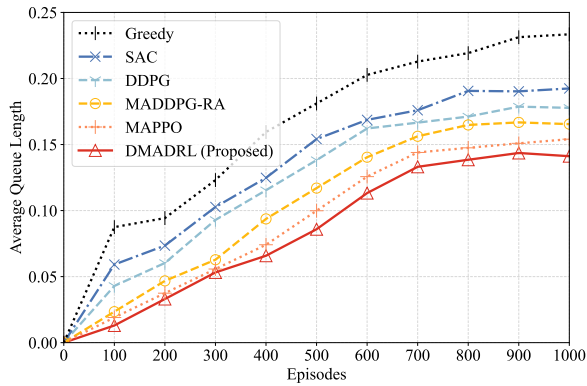


Fig. 8. Comparison of average queue length under different numbers of episodes.

changes. The conventional scheme is a semantic communication scheme without the introduction of the DMADRL algorithm, and the error-free transmission scheme is considered for comparison. As shown in Fig. 9, the semantic fidelity between the image, speech, and text of the proposed scheme and the traditional semantic communication all improve significantly with the increase of SNR. It is worth noting that the performance of the proposed scheme is always better than the traditional semantic communication scheme and obtains higher semantic fidelity values. This shows that the dynamic decision process based on DMADRL optimizes the semantic extraction factor, task offloading and resource allocation strategy, thereby achieving more efficient task offloading and resource allocation.

Fig. 10 compares the semantic fidelity of the proposed scheme and the benchmarks as the maximum bandwidth changes under different modal data. As shown in Fig. 10, the proposed scheme is consistently better than the traditional semantic communication scheme under different modalities. The performance improvement of DRMADRL is particularly significant with the increase of the maximum bandwidth, highlighting the importance of dynamically adjusting the semantic extraction factor. In contrast, the traditional semantic communication scheme shows relatively stable but lower performance. The DMADRL algorithm can achieve task offloading and resource allocation more efficiently in the SVEC by dynamically adjusting the semantic extraction factors, highlighting the adaptive advantage of DMADRL in dealing with different network conditions.

5) *Effect of the Number of Denoising Steps*: Fig. 11 depicts the impact of denoising steps on the performance of the proposed algorithm. As shown in Fig. 11, fewer denoising steps initially achieve higher rewards, likely due to reduced computational overhead and faster convergence in early training phases. Increasing the denoising step to Step=7 improves long-term performance, with the reward stabilizing at around 24 after 1,000 episodes, as the diffusion process can better model complex action distributions and increase the ability to explore the environment. However, increasing the number of denoising steps beyond a certain point may lead to performance degradation. Specifically, too many denoising steps

can result in the diffusion model removing excessive noise, leading to a loss of valuable data details that are crucial for accurate decision-making. In addition, errors introduced during each denoising step accumulate over time, which can cause the final output to diverge more significantly from the true distribution. To strike an optimal balance between efficiency and robustness, we found that setting the denoising step to 7 in DMADRL provides the best trade-off. This configuration offers a good balance between computational efficiency, exploration capability, and maintaining sufficient detail in the data, thereby maximizing performance without introducing unnecessary complexity.

## VII. CONCLUSION

In this paper, we proposed a novel SVEC architecture with semantic communication and task oriented task offloading strategies to address the highly constrained communication and computing resources issues of VEC. This architecture could extract key semantic features from different modal tasks using semantic communication for efficient data compression and adaptively offload the computing tasks to edge servers. To further improve the performance of SVEC, we proposed a diffusion-based multi-agent deep reinforcement learning scheme, which enhances the ability of agents to explore the system environment through a diffusion process and adaptively determines the optimal semantic extraction, task offloading and resource allocation strategies. Simulation results show that the proposed scheme improves the overall system performance of VEC while reducing offload latency and average system cost over the compared benchmark schemes. While SVEC has shown strong performance, challenges remain in scenarios with high vehicular density or poor communication links, where limited bandwidth and transmission noise hinder low-latency offloading and accurate semantic extraction. These limitations can significantly degrade system performance, leading to delays in task execution and loss of critical data integrity. Additionally, heterogeneous vehicle capabilities, such as varying processing power and sensor quality, along with dynamic workloads that fluctuate in real-time, may reduce the system's adaptability and robustness.

Future work could explore enhancing the scalability and resilience of SVEC through strategies such as adaptive model compression, meta-learning, and federated reinforcement learning to better support dynamic environments. These approaches aim to mitigate communication delays and processing inefficiencies by reducing model complexity, enabling efficient learning with limited data, and supporting decentralized decision-making. Additionally, investigating robust semantic extraction techniques and adaptive communication protocols to improve task offloading efficiency under bandwidth constraints and high latency is also a valuable research area.

## ACKNOWLEDGMENTS

This work was supported in part by RC Grant No. EP/Y027787/1, UKRI under grant No. EP/Y028317/1, and Horizon European program under grant No. 101086228, in

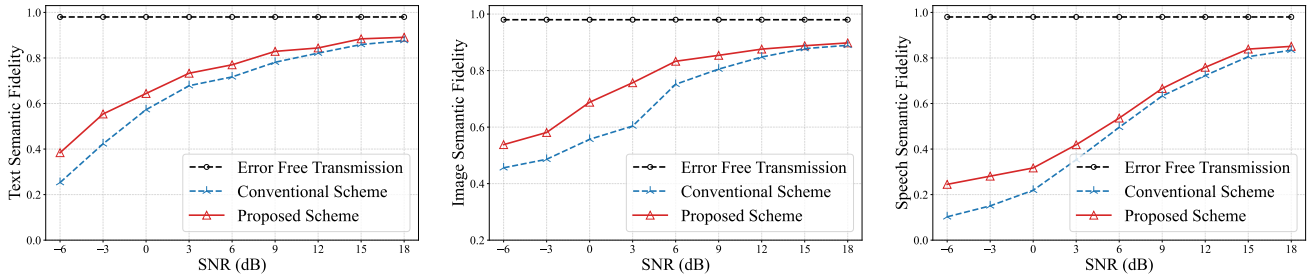


Fig. 9. Comparison of semantic fidelity for different modal data under varying SNR with the maximum bandwidth of 50 MHz.

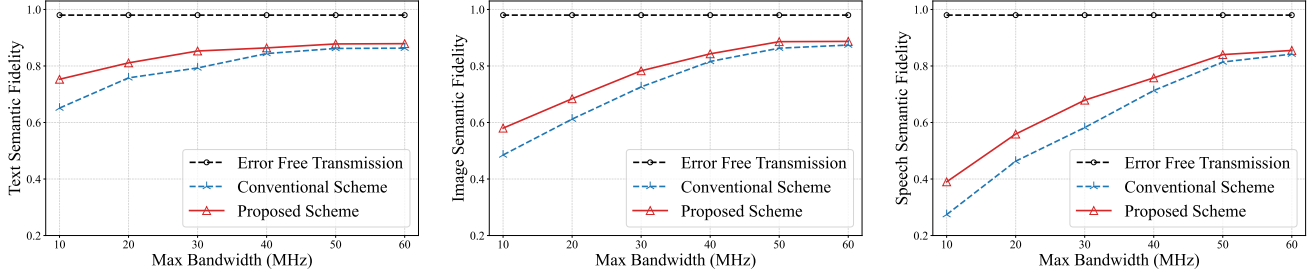


Fig. 10. Comparison of semantic fidelity for different modal data under varying bandwidth with the SNR of 15 dB.

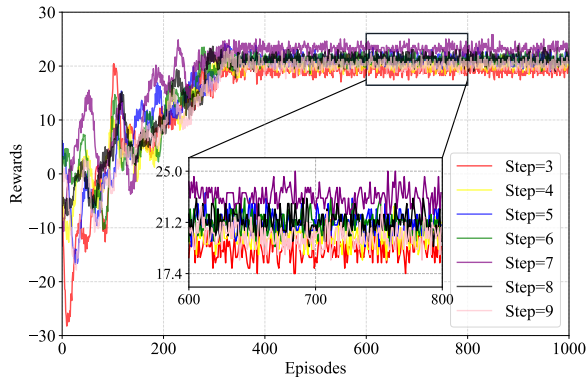


Fig. 11. Comparison of rewards for different denoising steps.

part by the National Natural Science Foundation of China under Grant 62272391 and Grant 62402391, in part by the Team-based Research Fund under Reference No. TBRF/2024/1.10, and in part by the Fundamental Research Funds for the Central Universities under Grant No. D5000250275.

## REFERENCES

- [1] M. Ahmed, S. Raza, A. A. Soofi, F. Khan, W. U. Khan, F. Xu, S. Chatzinotas, O. A. Dobre, and Z. Han, "A survey on reconfigurable intelligent surfaces assisted multi-access edge computing networks: State of the art and future challenges," *Computer Science Review*, vol. 54, p. 100668, 2024.
- [2] Z. Gao, L. Yang, and Y. Dai, "Vrccs-ac: Reinforcement learning for service migration in vehicular edge computing systems," *IEEE Transactions on Services Computing*, 2024.
- [3] F. Xiao, W. Fan, L. Han, T. Qiu, and X. Cheng, "Joint service deployment and task offloading for datacenters with edge heterogeneous servers," *IEEE Transactions on Services Computing*, 2025.
- [4] Q. Luo, C. Li, T. H. Luan, and W. Shi, "Minimizing the delay and cost of computation offloading for vehicular edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2897–2909, 2021.
- [5] D. Wu, Z. Wang, H. Pan, H. Yao, T. Mai, and S. Guo, "In-network computing empowered mobile edge offloading architecture for internet of things," *IEEE Transactions on Services Computing*, 2024.
- [6] L. Wang, W. Wu, F. Zhou, Z. Yang, Z. Qin, and Q. Wu, "Adaptive resource allocation for semantic communication networks," *IEEE Transactions on Communications*, vol. 72, no. 11, pp. 6900–6916, 2024.
- [7] J. Peng, H. Xing, X. Chen, Y. Li, Y. Cui, D. Zheng, L. Ale, and L. Feng, "Security enhanced computation offloading for collaborative inference at semantic-communication-empowered edge," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2025.
- [8] G. Zheng, Q. Ni, K. Navaie, and H. Pervaiz, "Semantic communication in satellite-borne edge cloud network for computation offloading," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1145–1158, 2024.
- [9] B. Guo, Z. Xiong, B. Wang, T. Q. S. Quek, and Z. Han, "Semantic communication-aware end-to-end routing in large-scale leo satellite networks," in *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, 2024, pp. 137–142.
- [10] H. Peng, Z. Zhang, Y. Liu, Z. Su, T. H. Luan, and N. Cheng, "Semantic communication in non-terrestrial networks: A future-ready paradigm," *IEEE Network*, vol. 38, no. 4, pp. 119–127, 2024.
- [11] V.-P. Bui, T. Q. Dinh, I. Leyva-Mayorga, S. R. Pandey, E. Lagunas, and P. Popovski, "Semantic image encoding and communication for earth observation with leo satellites," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 2, pp. 1210–1224, 2025.
- [12] J. Huang, J. Jiao, Y. Wang, R. Lu, and Q. Zhang, "Semantic-empowered utility loss of information transmission policy in satellite-integrated internet," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHOPS)*, 2024, pp. 1–6.
- [13] Q. Chen, X. Song, T. Song, and Y. Yang, "Vehicular edge computing networks optimization via drl-based communication resource allocation and load balancing," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2025.
- [14] L. Yan, Z. Qin, C. Li, R. Zhang, Y. Li, and X. Tao, "Qoe-based semantic-aware resource allocation for multi-task networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 11958–11971, 2024.
- [15] R. Zhang, K. Xiong, H. Du, D. Niyato, J. Kang, X. Shen, and H. V. Poor, "Generative ai-enabled vehicular networks: Fundamentals, framework, and case study," *IEEE Network*, vol. 38, no. 4, pp. 259–267, 2024.
- [16] J. Shi, J. Du, Y. Shen, J. Wang, J. Yuan, and Z. Han, "Drl-based

- v2v computation offloading for blockchain-enabled vehicular networks,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3882–3897, 2023.
- [17] W. Zhao, K. Shi, Z. Liu, X. Wu, X. Zheng, L. Wei, and N. Kato, “Drl connects lyapunov in delay and stability optimization for offloading proactive sensing tasks of rsus,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7969–7982, 2024.
- [18] Z. Shao, Q. Wu, P. Fan, K. Wang, Q. Fan, W. Chen, and K. B. Letaief, “Semantic-aware resource management for c-v2x platooning via multi-agent reinforcement learning,” *arXiv preprint arXiv:2411.04672*, 2024.
- [19] H. Liang, L. Zhu, F. R. Yu, and C. Yuen, “Cloud-edge-end collaboration for intelligent train regulation optimization in tacs,” *IEEE Transactions on Vehicular Technology*, 2024.
- [20] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, “Deep learning enabled semantic communication systems,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.
- [21] W. Yang, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Cao, and K. B. Letaief, “Semantic communication meets edge intelligence,” *IEEE wireless communications*, vol. 29, no. 5, pp. 28–35, 2022.
- [22] Z. Qin, J. Ying, D. Yang, H. Wang, and X. Tao, “Computing networks enabled semantic communications,” *IEEE Network*, vol. 38, no. 2, pp. 122–131, 2024.
- [23] Q. Qi, X. Chen, and C. Yuen, “Joint offloading selection and resource allocation for integrated localization and computing in edge-intelligent networks,” *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 11 427–11 440, 2024.
- [24] L. Wang, W. Wu, F. Zhou, Z. Yang, Z. Qin, and Q. Wu, “Adaptive resource allocation for semantic communication networks,” *IEEE Transactions on Communications*, 2024.
- [25] Y. Zheng, T. Zhang, and J. Loo, “Dynamic multi-time scale user admission and resource allocation for semantic extraction in mec systems,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 16 441–16 453, 2023.
- [26] Y. Cang, M. Chen, Z. Yang, Y. Hu, Y. Wang, C. Huang, and Z. Zhang, “Online resource allocation for semantic-aware edge computing systems,” *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 094–28 110, 2024.
- [27] J. Guo, H. Chen, B. Song, Y. Chi, C. Yuen, F. R. Yu, G. Y. Li, and D. Niyato, “Distributed task-oriented communication networks with multimodal semantic relay and edge intelligence,” *IEEE Communications Magazine*, vol. 62, no. 6, pp. 82–89, 2024.
- [28] Y. Zhang, Z. Yu, J. Zhang, L. Wang, T. H. Luan, B. Guo, and C. Yuen, “Learning decentralized traffic signal controllers with multi-agent graph reinforcement learning,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 7180–7195, 2023.
- [29] N. T. Hoa, C. T. T. Hai, H. L. Hung, N. Cong Luong, and D. Niyato, “Joint edge computing and semantic communication in uav-enabled networks,” *IEEE Communications Letters*, vol. 29, no. 1, pp. 80–84, 2025.
- [30] Z. Shao, Q. Wu, P. Fan, N. Cheng, W. Chen, J. Wang, and K. Ben Letaief, “Semantic-aware spectrum sharing in internet of vehicles based on deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 11, no. 23, pp. 38 521–38 536, 2024.
- [31] S. Wang, C. Yuen, W. Ni, Y. L. Guan, and T. Lv, “Multiagent deep reinforcement learning for cost-and delay-sensitive virtual network function placement and routing,” *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5208–5224, 2022.
- [32] Z. Ji, Z. Qin, X. Tao, and Z. Han, “Resource optimization for semantic-aware networks with task offloading,” *IEEE Transactions on Wireless Communications*, 2024.
- [33] Y. Liang, H. Tang, H. Wu, Y. Wang, and P. Jiao, “Lyapunov-guided offloading optimization based on soft actor-critic for isac-aided internet of vehicles,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14 708–14 721, 2024.
- [34] Z. Liu, H. Du, J. Lin, Z. Gao, L. Huang, S. Hosseinalipour, and D. Niyato, “Dnn partitioning, task offloading, and resource allocation in dynamic vehicular networks: A lyapunov-guided diffusion-based reinforcement learning approach,” *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 1945–1962, 2025.
- [35] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” *arXiv preprint arXiv:1611.01144*, 2016.
- [36] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, “Cooperative task offloading and block mining in blockchain-based edge computing with multi-agent deep reinforcement learning,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, 2021.
- [37] Y. Jia, R. Mao, Y. Sun, S. Zhou, and Z. Niu, “Mass: Mobility-aware sensor scheduling of cooperative perception for connected automated

driving,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14 962–14 977, 2023.

- [38] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.
- [39] J. Liu, Y. Lu, H. Wu, and Y. Dai, “Efficient resource allocation and semantic extraction for federated learning empowered vehicular semantic communication,” in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*. IEEE, 2023, pp. 1–5.
- [40] L. Wu, J. Qu, S. Li, C. Zhang, J. Du, X. Sun, and J. Zhou, “Attention-augmented maddpg in noma-based vehicular mobile edge computational offloading,” *IEEE Internet of Things Journal*, 2024.
- [41] F. Zhao, G. Bagwe, E. Mohammed, L. Feng, L. Zhang, and Y. Sun, “Joint computing resource and bandwidth allocation for semantic communication networks,” in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*. IEEE, 2023, pp. 1–5.



**Yi Yang** (Student Member, IEEE) received the B.Eng. degree software engineering from Northwestern Polytechnical University, Xi’an, China in 2017. He is currently working toward the Ph.D. degree in Cybersecurity with Northwestern Polytechnical University, Xi’an, China. His research interests include incentive mechanism, wireless mobile communications, digital twins, Internet of Things, and federated learning.



**Wenqiang Ma** (Student Member, IEEE) received the B.E. degree from the School of Computer, Northwestern Polytechnical University in 2021. He is currently pursuing a Ph.D. degree in the School of Cybersecurity, Northwestern Polytechnical University, Xi’an, China. His research interests include federated learning, mobile edge computing, semantic communication and reinforcement learning.



**Wen Sun** (Senior Member, IEEE) received the B.E. degree from the Harbin Institute of Technology, Harbin, China, in 2009, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2014. She is currently a Full Professor with the School of Cybersecurity, Northwestern Polytechnical University, Xi’an, China. She has authored or coauthored more than 70 peer-reviewed papers in various prestigious IEEE journals and conferences, including *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Wireless Communications*, *IEEE Network*, and *IEEE Wireless Communications*. Her research interests include wireless mobile communications, IoT, 5G, and blockchain. She was the recipient of the Best Paper Award of *GlobeCom2019*. She is the publicity Chair of *WiMob2019* and *CNS2020*, and a TPC Member of *ICC* and *GlobeCom* in 2018 and 2019.



**Jianhua He** (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2002. He was with the University of Bristol, Swansea University, and Aston University. He is currently a Reader with the University of Essex, U.K. He has published more than 150 research papers in refereed international journals and conferences. He is the Coordinator of EU Horizon 2020 projects COSAFE and VESAFE on cooperative connected autonomous vehicles. His main research interests include 5G/6G wireless communi-

cations and networks, connected vehicles, autonomous driving, the Internet of Things, mobile edge computing, intelligent transport systems, data analytics, AI, and machine learning. He was the Workshop Chair of MobiArch'20 and ICAV'21 and a Steering Committee Member of MobiArch'21 and MobiArch'22. He is a member of the editorial board of IEEE Wireless Communications Letters and The Computer Journal.



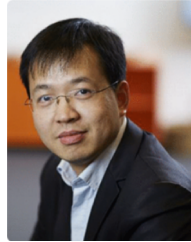
**Yaru Fu** (Member, IEEE) currently serves as an Associate Professor and Head of the Centre for Research in Advanced Network Technologies (CRANT) at Hong Kong Metropolitan University, Hong Kong, China. She earned her Ph.D. from the Department of Electronic Engineering at the City University of Hong Kong, Hong Kong, China, in 2018. Her primary research interests include 5G/6G technologies, digital twins, and machine learning. She has authored over 100 papers in prestigious IEEE journals and conferences.

Dr. Fu is on the editorial boards of IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, IEEE INTERNET OF THINGS JOURNAL, IEEE WIRELESS COMMUNICATIONS LETTERS, and IEEE NETWORKING LETTERS. In recognition of her contributions, Dr. Fu received the IEEE WCL Best Editor Award in 2021, the IEEE Transactions on Communications Exemplary Reviewer Award in 2022, the President's Award of Hong Kong Metropolitan University (the sole recipient) in 2023, and the IEEE Global Communications Conference (GLOBECOM) Best Paper Award in 2024. She was also honored with the 2023, 2024, and 2025 World's Top 2% Scientists ranking by Stanford University. Furthermore, Dr. Fu has held positions such as Track Chair, Workshop Chair, Tutorial Leading Speaker, and Technical Program Committee Member for various IEEE leading conferences, including IEEE ICC, WCNC, VTC, WF-IoT, and GLOBECOM.



**Chau Yuen** (Fellow, IEEE) received the B.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore, in 2000 and 2004, respectively. He was a Post-Doctoral Fellow with the Lucent Technologies Bell Laboratories, Murray Hill, in 2005. From 2006 to 2010, he was with the Institute for Infocomm Research, Singapore. From 2010 to 2023, he was with the Engineering Product Development Pillar, Singapore University of Technology and Design. Since 2023 he has been with the School of Electrical and Electronic Engineering, Nanyang

Technological University. Currently he is the Provost's Chair of Wireless Communications, the Assistant Dean of the Graduate College, and the Cluster Director for Sustainable Built Environment at ER@IN. He received the IEEE Communications Society Fred W. Ellersick Prize in 2023, the IEEE Marconi Prize Paper Award in Wireless Communications in 2021, and the EURASIP Best Paper Award for Journal on Wireless Communications and Networking in 2021. He received the IEEE Asia-Pacific Outstanding Young Researcher Award in 2012 and the IEEE VTS Singapore Chapter Outstanding Service Award in 2019. He is a Distinguished Lecturer of the IEEE Vehicular Technology Society, the Top 2% Scientists by Stanford University, and a Highly Cited Researcher by Clarivate Web of Science.



**Yan Zhang** (Fellow, IEEE) received the PhD degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. He is currently a full professor with the Department of Informatics, University of Oslo, Oslo, Norway. His research interests include next-generation wireless networks leading to 6G and green and secure cyber-physical systems (e.g., smart grid and transport). He was a recipient of the Global Highly Cited Researcher Award (Web of Science top 1% most cited worldwide), since 2018. He is

a Symposium/Track chair in a number of conferences, including IEEE ICC 2021, IEEE Globecom 2017, IEEE PIMRC 2016, and IEEE SmartGridComm 2015. He is the chair of IEEE Communications Society Technical Committee on Green Communications and Computing. He is an editor (or an area editor, a senior editor, and an associate editor) for several IEEE Transactions/magazines, including IEEE Communications Magazine, IEEE Network Magazine, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Vehicular Technology, IEEE Transactions on Industrial Informatics, IEEE Transactions on Green Communications and Networking, IEEE Communications Survey and Tutorials, IEEE Internet of Things Journal, IEEE Systems Journal, IEEE Vehicular Technology Magazine, and IEEE Blockchain Technical Briefs. He is a CCF senior member, an elected member of CCF Technical Committee of Blockchain, and a CCF distinguished speaker, in 2019. He is a fellow of IET and an elected member of Academia Europaea and the Norwegian Academy of Technological Sciences.